Security Assessment, Analysis, and Assurance

7

7.1 Introduction

The rapid development in both computer and telecommunication technologies has resulted in massive interconnectivity and interoperability of systems. The world is getting more and more interconnected every day. Most major organization systems are interconnected to other systems through networks. The bigger the networks, the bigger the security problems involving the system resources on these networks. Many companies, businesses, and institutions whose systems work in coordination and collaboration with other systems as they share each other's resources and communicate with each other face a constant security threat to these systems, yet the collaboration must go on.

The risks and potential of someone intruding into these systems for sabotage, vandalism, and resource theft are high. For security assurance of networked systems, such risks must be assessed to determine the adequacy of existing security measures and safeguards and also to determine if improvement in the existing measures is needed. Such an assessment process consists of a comprehensive and continuous analysis of the security threat risk to the system that involves an auditing of the system, assessing the vulnerabilities of the system, and maintaining a creditable security policy and a vigorous regime for the installation of patches and security updates. In addition, there must also be a standard process to minimize the risks associated with nonstandard security implementations across shared infrastructures and end systems.

The process to achieve all these and more consists of several tasks including a system security policy, security requirements specification, identification of threat and threat analysis, vulnerability assessment, security certification, and the monitoring of vulnerabilities and auditing. The completion of these tasks marks a completion of a security milestone on the road to a system's security assurance. These tasks are shown in Table 7.1 below.

Security is a process. Security assurance is a continuous security state of the security process. The process, illustrated in Table 7.1 and depicted in Fig. 7.1, starts

J.M. Kizza, *Guide to Computer Network Security*, Computer Communications and Networks, DOI 10.1007/978-3-319-55606-2_7

Table 7.1 System security process

System security policy
Security requirements specification
Threat identification
Threat analysis
Vulnerability identification and assessment
Security certification
Security monitoring and auditing

with a thorough system security policy, whose components are used for system requirement specifications. The security requirement specifications are then used to identify threats to the system resources. An analysis of these identified threats per resource is then done. The vulnerabilities identified by the threats are then assessed, and if the security measures taken are good enough, they are then certified, along with the security staff.

After certification, the final component of the security process is the auditing and monitoring phase. This phase may reveal more security problems which require revisiting the security policy that makes the process start to repeat itself. That security cycle process is security assurance. The process of security assurance is shown in Fig. 7.1.

7.2 System Security Policy

To a system administrator, the security of the organization's system is very important. For any organization system, there must be somebody to say *no* when the *no* needs to be said. The *no* must be said because the administrator wants to limit the number of network computers, resources, and capabilities people have been using to ensure the security of the system. One way of doing this in fairness to all is through the implementation of a set of policies, procedures, and guidelines that tell all employees and business partners what constitutes acceptable and unacceptable use of the organization's computer system. The security policy also spells out what resources need to be protected and how organization can protect such resources. A security policy is a living set of policies and procedures that impact and potentially limit the freedoms and of course levels of individual security responsibilities of all users. Such a structure is essential to an organization's security. Having said that, however, let us qualify our last statement. There are as many opinions on the usefulness of security policies in the overall system security picture as there are security experts. However, security policies are still important in the security plan of a system. It is important for several reasons including:

- Firewall installations: If a functioning firewall is to be configured, its rule base must be based on a sound security policy.
- User discipline: All users in the organization who connect to a network such as the Internet, through a firewall, say, must conform to the security policy.

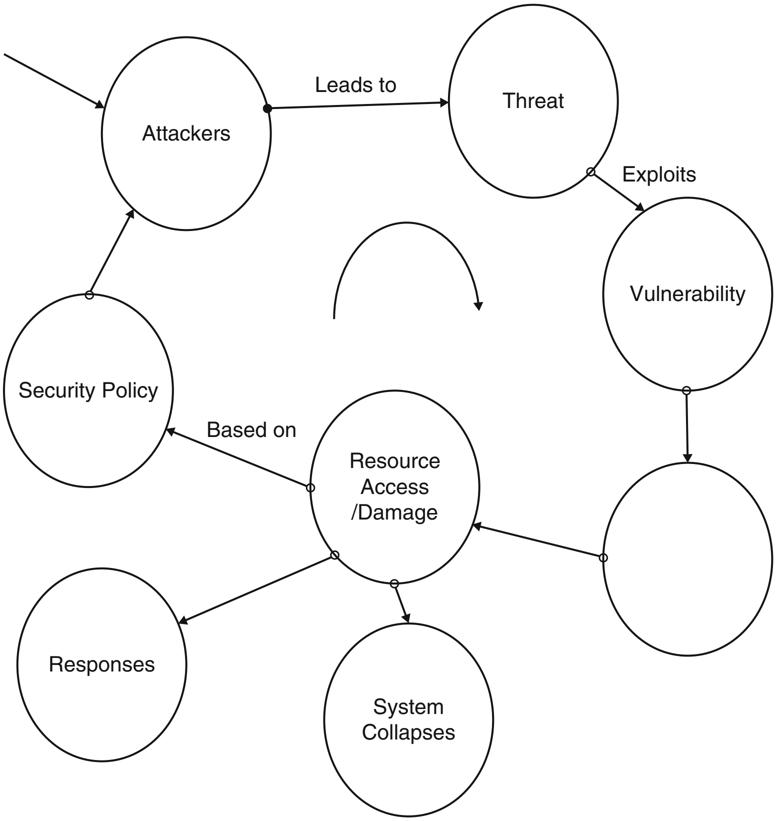


Fig. 7.1 System security assurance cycle

Without a strong security policy that every employee must conform to, the organization may suffer from data loss, employee time loss, and productivity loss all because employees may spend time fixing holes, repairing vulnerabilities, and recovering lost or compromised data among other things.

A security policy covers a wide variety of topics and serves several important purposes in the system security cycle. Constructing a security policy is like building a house; it needs a lot of different components that must fit together. The security policy is built in stages, and each stage adds value to the overall product, making it unique for the organization. To be successful, a security policy must:

- Have the backing of the organization top management.
- Involve everyone in the organization by explicitly stating the role everyone will play and the responsibilities of everyone in the security of the organization.
- Precisely describe a clear vision of a secure environment stating what needs to be protected and the reasons for it.
- Set priorities and costs of what needs to be protected.

- Be a good teaching tool for everyone in the organization about security and what needs to be protected, why, and how it is to be protected.
- Set boundaries on what constitutes appropriate and inappropriate behavior as far as security and privacy of the organization resources are concerned.
- Create a security clearing house and authority.
- Be flexible enough to adapt to new changes.
- Be consistently implemented throughout the organization.

To achieve these subgoals, a carefully chosen set of basic steps must be followed to construct a viable implementable and useful security policy.

According to Jasma, the core five steps are the following [1, 2]:

- Determine the resources that must be protected, and for each resource, draw a profile of its characteristics. Such resources should include physical, logical, network, and system assets. A table of these items ordered in importance should be developed.
- For each identified resource, determine from whom you must protect.
- For each identifiable resource, determine the type of threat and the likelihood of such a threat. For each threat, identify the security risk and construct an ordered table for these based on importance. Such risks may include:
  - Denial of service
  - Disclosure or modification of information
  - Unauthorized access
- For each identifiable resource, determine what measures will protect it the best and from whom.
- Develop a policy team consisting of at least one member from senior administration, legal staff, employees, member of IT department, and an editor or writer to help with drafting the policy.
- Determine what needs to be audited. Programs such as Tripwire perform audits on both Unix and Windows systems. Audit security events on servers and firewalls and also on selected network hosts. For example, the following logs can be audited:
  - Log files for all selected network hosts, including servers and firewalls
  - Object accesses
- Define acceptable use of system resources such as:
  - E-mail
  - News
  - Web
- Consider how to deal with each of the following:
  - Encryption
  - Password
  - Key creation and distributions
  - Wireless devices that connect on the organization’s network

- Provide for remote access to accommodate workers on the road and those working from home and also business partners who may need to connect through a virtual private network (VPN).

From all this information, develop two structures, one describing the access rights of users to the resources identified and the other structure describing user responsibilities in ensuring security for a given resource. Finally, schedule a time to review these structures regularly.

7.3 Building a Security Policy

Several issues, including the security policy access matrix, need to be constructed first before others can fit in place. So let us start with that.

7.3.1 Security Policy Access Rights Matrix

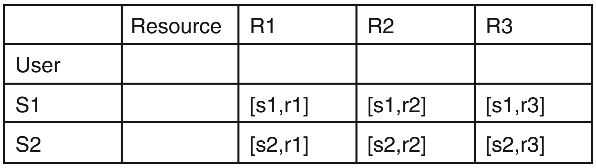
The first consideration in building a security policy is to construct a security policy access rights matrix M = {S, R} where S = {set of all user groups, some groups may have one element} and R = {set of system resources}. For example R = {network hosts, switches, routers, firewalls, access servers, databases, files, e-mail, Web site, remote access point, etc.}. And S = [{administrator}, {support technicians}, {Human Resource users}, {Marketing users}, etc.}].

For each element *rj* of R, develop a set of policies Pj. For example, create policies for the following members of R:

- E-mail and Web access (SNMP, DNS, NTP, WWW, NNTP, SMTP)
- Hardware access (logon passwords/usernames)
- Databases (file access/data backup)
- Wireless devices (access point logon/authentication/access control)
- Laptops' use and connection to organization's network
- Remote access (telnet, FTP)

For each element *si* of S, develop a set of responsibilities Ni. For example, create responsibilities for the following members of S:

- Who distributes system resources access rights/remote access/wireless access?
- Who creates accounts/remote access accounts?
- Who talks to the press?
- Who calls law enforcement?
- Who informs management of incidents and at what level?
- Who releases and what data?
- Who follows on a detected security incident?



	Resource	R1	R2	R3
User				
S1		[s1,r1]	[s1,r2]	[s1,r3]
S2		[s2,r1]	[s2,r2]	[s2,r3]

Fig. 7.2 Security policy access rights matrix M

Once all access rights and responsibilities have been assigned, the matrix M is fully filled, and the policy is now slowly taking shape. Up to this point, an entry in $\mathrm{M} = \{[si, rj]\}$ means that user from group *si* can use any of the rights in group *rj* for the resource j. See Fig. 7.2.

A structure $\mathrm{L} = \{\mathrm{S}, \mathrm{R}\}$, similar to M, for responsibilities can also be constructed. After constructing these two structures, the security policy is now taking shape, but it is far from done. Several other security issues need to be taken care of, including those described in the following sections [3]:

7.3.1.1 Logical Access Restriction to the System Resources

Logical access restriction to system resources involves the following:

- Restricting access to equipment and network segments using:
  - Preventive controls that uniquely identify every authorized user (via established access control mechanisms such as passwords) and deny others.
  - Detective controls that log and report activities of users, both authorized and intruders. This may employ the use of intrusion detection systems, system scans, and other activity loggers. The logs are reported to a responsible party for action.
- Creating boundaries between network segments:
  - To control the flow of traffic between different cabled segments such as subnets by using IP address filters to deny access of specific subnets by IP addresses from nontrusted hosts.
  - Permit or deny access based on subnet addresses, if possible.
- Selecting a suitable routing policy to determine how traffic is controlled between subnets.

7.3.1.2 Physical Security of Resources and Site Environment

Establish physical security of all system resources by:

- Safeguarding physical infrastructure including media and path of physical cabling. Make sure that intruders cannot eavesdrop between lines by using detectors such as time domain reflectometer for coaxial cable and optical splitter using an optical time domain reflectometer for fiber optics.
- Safeguarding site environment. Make sure it is as safe as you can make it from security threats due to:
  - Fire (prevention/protection/detection)
  - Water

  - Electric power surges
  - Temperature/humidity
  - Natural disasters
  - Magnetic fields

7.3.1.3 Cryptographic Restrictions

We have defined a computer and also a network system as consisting of hardware, software, and users. The security of an organization's network, therefore, does not stop only at securing software such as the application software like browsers on the network hosts. It also includes data in storage in the network, that is, at network servers, and also data in motion within the network.

Ensuring this kind of software and data requires strong cryptographic techniques. So an organization's security policy must include a consideration of cryptographic algorithms to ensure data integrity. The best way to ensure as best as possible that traffic on the network is valid is through the following:

- Supported services, such as firewalls, relying on the TCP, UDP, ICMP, and IP headers, and TCP and UDP source and destination port numbers of individual packets to allow or deny the packet.
- Authentication of all data traversing the network, including traffic specific to the operations of a secure network infrastructure such as updating of routing tables.
- Checksum to protect against the injection of spurious packets from an intruder, and in combination with sequence number techniques, protects against replay attacks.
- Software not related to work will not be used on any computer that is part of the network.
- All software images and operating systems should use a checksum verification scheme before installation to confirm their integrity between sending and receiving devices.
- Encryption of routing tables and all data that pose the greatest risk based on the outcome of the risk assessment procedure in which data is classified according to its security sensitivity. For example, in an enterprise, consider the following:
  - All data dealing with employee salary and benefits
  - All data on product development
  - All data on sales
- Also pay attention to the local network address translation (NAT)—a system used to help network administrators with large pools of hosts from renumbering them when they all come on the Internet.
- Encrypt the backups making sure that they will be decrypted when needed.

7.3.2 Policy and Procedures

No security policy is complete without a section on policy and procedures. In fact, several issues are covered under policy and procedures. Among the items in this section is a list of common attacks for technicians to be aware of education of users, equipment use, equipment acquisition, software standards and acquisition, and incident handling and reporting.

7.3.2.1 Common Attacks and Possible Deterrents

Some of the most common deterrents to common attacks include the following:

- Developing a policy to insulate internal hosts (hosts behind a firewall) from a list of common attacks.
- Developing a policy to safeguard Web servers, FTP servers, and e-mail servers, which of these are at most risk because even though they are behind a firewall, any host, even those inside the network, can misuse them. You are generally better of putting those exposed service providers on a *demilitarized zone* (DMZ) network.
- Installing a honey port.

The following list provides an example of some items in an infrastructure and data integrity security policy:

7.3.2.2 Staff

- Recruit employees for positions in the implementation and operation of the network infrastructure who are capable and whose background has been checked.
- Have all personnel involved in the implementation and supporting the network infrastructure must attend a security seminar for awareness.
- Instruct all employees concerned to store all backups in a dedicated locked area.

7.3.2.3 Equipment Certification

To be sure that quality equipments are used, make every effort to ensure that:

- All new equipment to be added to the infrastructure should adhere to specified security requirements.
- Each site of the infrastructure should decide which security features and functionalities are necessary to support the security policy.
- The following are good guidelines:
  - All infrastructure equipment must pass the acquisition certification process before purchase.
  - All new images and configurations must be modeled in a test facility before deployment.
  - All major scheduled network outages and interruptions of services must be announced to those who will be affected well ahead of time.
- Use of Portable Tools
  - Since use of portable tools such as laptops always pose some security risks, develop guidelines for the kinds of data allowed to reside on hard drives of portable tools and how that data should be protected.

7.3.2.4 Audit Trails and Legal Evidence

Prepare for possible legal action by:

- Keeping logs of traffic patterns and noting any deviations from normal behavior found. Such deviations are the first clues to security problems.
- Keeping the collected data locally to the resource until an event is finished, after which it may be taken, according to established means involving encryption, to a secure location.
- Securing audit data on location and in backups.

7.3.2.5 Privacy Concerns

There are two areas of concern with audit trail logs:

- Privacy issue of the data collected on users
- Knowledge of an intrusive behavior of others including employees of the organization

7.3.2.6 Security Awareness Training

The strength of a security policy lies in its emphasis on both employee and user training. The policy must stress that:

- Users of computers and computer networks must be made aware of the security ramifications caused by certain actions. The training should be provided to all personnel.
- Training should be focused and involve all types of security that are needed in the organization, the internal control techniques that will meet the security requirements of the organization, and how to maintain the security attained.
- Employees with network security responsibilities must be taught security techniques probably beyond those of the general public, methodologies for evaluating threats and vulnerabilities to be able to use them to defend the organization’s security, the ability to select and implement security controls, and a thorough understanding of the importance of what is at risk if security is not maintained.
- Before connecting to a LAN to the organization’s backbone, provide those responsible for the organization’s security with documentation on network infrastructure layout, rules, and guidelines on controlled software downloads. Pay attention to the training given to those who will be in charge of issuing passwords.
- Social engineering.
- Train employees not to believe anyone who calls/e-mails them to do something that might compromise security.
- Before giving any information, employees must positively identify who they are dealing with.

7.3.2.7 Incident Handling

The security of an organization's network depends on what the security plan says should be done to handle a security incident. If the response is fast and effective, the losses may be none to minimum. However, if the response is bungled and slow, the losses may be heavy. Make sure that the security plan is clear and effective:

- Build an incident response team as a centralized core group, whose members are drawn from across the organization, who must be knowledgeable, and well rounded with a correct mix of technical, communication, and political skills. The team should be the main contact point in case of a security incident and responsible for keeping up-to-date with the latest threats and incidents, notifying others of the incident, assessing the damage and impact of the incident, finding out how to minimize the loss, avoid further exploitation of the same vulnerability, and making plans and efforts to recover from the incident.
- Detect incidents by looking for signs of a security breach in the usual suspects and beyond. Look for abnormal signs from accounting reports, focus on signs of data modification and deletion, check out complaints of poor system performance, pay attention to strange traffic patterns and unusual times of system use, and pick interest in large numbers of failed log-in attempts.
- Assess the damage by checking and analyzing all traffic logs for abnormal behavior, especially on network perimeter access points such as Internet access or dial-in access. Pay particular attention when verifying infrastructure device checksum or operating system checksum on critical servers to see whether operating system software has been compromised or if configuration changes in infrastructure devices such as servers have occurred to ensure that no one has tampered with them. Make sure to check the sensitive data to see whether it has been assessed or changed and traffic logs for unusually large traffic streams from a single source or streams going to a single destination, passwords on critical systems to ensure that they have not been modified, and any new or unknown devices on the network for abnormal activities.
- Report and alert.
  - Establish a systematic approach for reporting incidents and subsequently notifying affected areas.
  - Essential communication mechanisms include a monitored central phone, e-mail, pager, or other quick communication devices.
  - Establish clearly whom to alert first and who should be on the list of people to alert next.
  - Decide on how much information to give each member on the list.
  - Find ways to minimize negative exposure, especially where it requires working with agents to protect evidence.
- Respond to the incident to try to restore the system back to its pre-incident status. Sometimes it may require shutting down the system; if this is necessary, then do so but keep accurate documentation and a log book of all activities during the incident so that that data can be used later to analyze any causes and effects.
- Recover from an incident

  - Make a postmortem analysis of what happened, how it happened, and what steps need to be taken to prevent similar incidents in the future.
  - Develop a formal report with proper chronological sequence of events to be presented to management.
  - Make sure not to overreact by turning your system into a fortress.

7.4 Security Requirements Specification

Security requirements specification derives directly from the security policy document. The specifications are details of the security characteristics of every individual and system resource involved. For details on individual users and system resources, see the security access matrix. These security requirements are established after a process of careful study of the proposed system that starts with a brainstorming session to establish and maintain a skeleton basis of a basket of core security requirements by all users. The brainstorming session is then followed by establishing a common understanding and agreement on the core requirements for all involved. For each requirement in the core, we then determine what we need and how far to go to acquire and maintain it, and finally for each core requirement, we estimate the time and cost for its implementation.

From the security policy access right matrix, two main entries in the matrix, the user and the resources, determine the security requirements specifications as follows [4]:

- For the user: Include username, location, and phone number of the responsible system owner and data/application owner. Also determine the range of security clearance levels, the set of formal access approvals, and the need-to-know of users of the system.
  - Personnel security levels: Set the range of security clearance levels, the set of formal access approvals, and the need-to-know of users of the system
- For the resources: Include the resource type, document any special physical protection requirements that are unique to the system, and brief description of a secure operating system environment in use. If the resource is data, then include the following also:
  - Classification level, top secret, secret, and confidential, and categories of data, restricted and formally restricted
  - Any special access programs for the data
  - Any special formal access approval necessary for access to the data
  - Any special handling instructions
  - Any need-to-know restrictions on users
  - Any sensitive classification or lack of

After the generation of the security requirements for each user and system resource in the security policy access matrix, a new security requirements matrix, Table 7.2, is drawn.

Table 7.2 Listing of system security requirements

System components (resources and content)	Security requirements
Network client	Sign-on and authentication of user
	Secure directory for user ID and passwords
	Secure client software
	Secure session manager to manage the session
Network server	Secure software to access the server
	Secure client software to access the server
Content/data	Data authentication
	Secure data on server
	Secure data on client

7.5 Threat Identification

To understand system threats and deal with them, we first need to be able to identify them. Threat identification is a process that defines and points out the source of the threat and categorizes it as either a person or an event. For each system component whose security requirements have been identified, as shown in Fig. 4.4, also identify the security threats to it. The security threats to any system component can be deliberate or nondeliberate. A threat is deliberate if the act is done with the intention to breach the security of an object. There are many types of threats under this category, as we saw in Chap. 3. Nondeliberate threats, however, are acts and situations that, although they have the potential to cause harm to an object, were not intended. As we saw in Chap. 3, the sources of threats are many and varied including human factors, natural disasters, and infrastructure failures.

7.5.1 Human Factors

Human factors are those acts that result from human perception and physical capabilities and may contribute increased risks to the system. Among such factors are the following [5]:

- Communication—Communication between system users and personnel may present risk challenges based on understanding of policies and user guidelines, terminology used by the system, and interpersonal communication skills, and languages.
- Human-machine interface—Many users may find a challenge in some of the system interfaces. How the individual using such interfaces handles and uses them may cause a security threat to the system. The problem is more so when there is a degree of automation in the interface.

- Data design, analysis, and interpretation—Whenever there is more than one person, there is always a chance of misinterpretation of data and designs. So if there is any system data that needs to be analyzed and interpreted, there is always a likelihood of someone misinterpreting it or using a wrong design.
- New tools and technologies—Whenever a system has tools and technologies that are new to users, there is always a risk in the use of those tools and technologies. Also long-term exposure to such tools may cause significant neuromusculoskeletal adaptation with significant consequences on their use.
- Workload and user capacity—Users in many systems become victims of the workload and job capacity; this may, if not adjusted, cause risk to systems. Attributes of the task such as event rate, noise, uncertainty, criticality, and complexity that affect human mental and physical behavior may have an effect on the effort required for users to complete their assigned tasks.
- Work environment—As many workers know, the work environment greatly affects the human mental and physical capacity in areas of perception, judgment, and endurance. The factors that affect the working environment include things such as lighting, noise, workstations, and spatial configuration.
- Training—Training of system personnel and also users creates a safer user environment than that of systems with untrained users and personnel. Trained users will know when and how certain equipment and technologies can be used safely.
- Performance—A safe system is a system where the users and personnel get maximum performance from the system and from the personnel. Efficient and successful completion of all critical tasks on the system hinges on the system personnel and users maintaining required physical, perceptual, and social capabilities.

7.5.2 Natural Disasters

There is a long list of natural acts that are sources of security threats. These include earthquakes, fires, floods, hurricanes, tornados, lightning, and many others. Although natural disasters cannot be anticipated, we can plan for them. There are several ways to plan for the natural disaster threats. These include creating up-to-date backups stored at different locations that can be quickly retrieved and set up and having a comprehensive recovery plan. Recovery plans should be implemented rapidly.

7.5.3 Infrastructure Failures

System infrastructures are composed of hardware, software, and humanware. Any of these may fail the system anytime without warning.

7.5.3.1 Hardware Failures

Because computers have been in use for a long time now, this gives us some digree of relief that computer hardware are getting more relaible than ever before. But still, hardware failures are common due to wear and tear and age. The operating environment also contributes greatly to hardware failures. For example, a hostile environment due to high temperatures and moisture and dust always results in hardware failures. There are several approaches to overcome hardware threats, including redundancy, where there is always a standby similar system to kick in whenever there is an unplanned stoppage of the functioning system. Another way of overcoming hardware failure threats is to have a monitoring system where two or more hardware units constantly monitor each other and report to others whenever one of them fails. In addition, advances in hardware technology have led to the development of self-healing hardware units whenever a system detects its component performance, and if one component shows signs of failure, the unit quickly disables the component and reroutes or reassigns the functions of the failing component and also reports the failing component to all others in the unit.

7.5.3.2 Software Failures

Probably the greatest security threat, when everything is considered, is from software. The history of computing is littered with examples of costly catastrophes of failed software projects and potential software failures and errors such as the millennium scare. Failure or poor performance of a software product can be attributed to a variety of causes, most notably human error, the nature of software itself, and the environment in which software is produced and used.

Both software professionals and nonprofessionals who use software know the differences between software programming and hardware engineering. It is in these differences that lie many of the causes of software failure and poor performance. Consider the following [6]:

- *Complexity*: Unlike hardwired programming in which it is easy to exhaust the possible outcomes on a given set of input sequences, in software programming a similar program may present billions of possible outcomes on the same input sequence. Therefore, in software programming, one can never be sure of all the possibilities on any given input sequence.
- *Difficult testing*: There will never be a complete set of test programs to check software exhaustively for all bugs for a given input sequence.
- *Ease of programming*: The fact that software programming is easy to learn encourages many people with little formal training and education in the field to start developing programs, but many are not knowledgeable about good programming practices or able to check for errors.
- *Misunderstanding of basic design specifications*: This affects the subsequent design phases including coding, documenting, and testing. It also results in improper and ambiguous specifications of major components of the software and in ill-chosen and poorly defined internal program structures.

- *Software evolution*: It is almost an accepted practice in software development that software products that grow out from one version or release to another are made by just additions of new features without an overhaul of the original version for errors and bugs. This is always a problem because there are many incompatibilities that can cause problems, including different programmers with different design styles from those of the original programmers; different software modules, usually newer, that may have differing interfaces; and different expectations and applications that may far exceed the capabilities of the original version. All these have led to major flaws in software that can be exploited and have been exploited by hackers.
- *Changing management styles*: Quite often organizations change management, and the new management comes in with a different focus and different agenda that may require changes that may affect the software used by the organization in order to accommodate the new changes. Because of time and cost considerations, many times the changes are made in-house. Introducing such changes into existing software may introduce new flaws and bugs or may reactivate existing but dormant errors.

7.5.3.3 Humanware Failures

The human component in the computer systems is considerable and plays a vital role in the security of the system. While inputs to and sometimes outputs from hardware components can be predicted, and also in many cases software bugs once found can be fixed and the problem forgiven, the human component in a computer system is so unpredictable and so unreliable that the inputs to the system from the human component may never be trusted, a major source of system threat. The human link in the computing system has been known to be a source of many malicious acts that directly affect the security of the system. Such malicious acts include hacking into systems and creating software that threaten the security of systems. In later chapters, we will talk more about these activities.

7.6 Threat Analysis

A working computer system with numerous resources is always a target of many security threats. A *threat* is the combination of an asset such as a system resource, a vulnerability, or an exploit that can be used by a hacker to gain access to the system. Although every system resource has value, there are those with more intrinsic value than others. Such resources, given a system vulnerability that can let in an intruder, attract system intruders more than their counterparts with limited intrinsic value. Security threat analysis is a technique used to identify these resources and to focus on them. In general, *system security threat analysis* is a process that involves ongoing testing and evaluation of the security of a system's resources to continuously and critically evaluate their security from the perspective of a malicious intruder and then use the information from these evaluations to increase the overall system's security.

The process of security threat analysis involves the following:

- Determining those resources with higher intrinsic value, prioritizing them, and focusing on that list as defense mechanisms are being considered.
- Documenting why the chosen resources need to be protected in the hierarchy they are put in.
- Determining who causes what threat to whose resources.
- Identifying known and plausible vulnerabilities for each identified resource in the system. Known vulnerabilities, of course, are much easier to deal with than vulnerabilities that are purely speculative.
- Identifying necessary security services/mechanisms to counter the vulnerability.
- Increasing the overall system security by focusing on identified resources.

7.6.1 Approaches to Security Threat Analysis

There are several approaches to security threat analysis, but we will consider two of them here: the simple threat analysis by calculating *annualized loss expectancies* (ALEs) and attack trees.

7.6.1.1 Threat Analysis by Annualized Loss Expectancies

Before we define annualized loss expectancies, let us define the terms from which ALE is derived. For a resource identified as having a high threat risk, the cost of replacing or restoring that resource if it is attacked is its *single-loss expectancy* cost. The security threat is a resource's vulnerability. So if the vulnerability is likely to occur a certain number of times (based on past occurrences), then the vulnerability's *expected annual rate of occurrence* (EAO) can be computed.

Then multiplying these two terms gives us the vulnerability's annualized loss expectancy as [7]

Annualized loss expectancy (ALE for a resource) = *single-loss expectancy* (cost) × (expected) *annual rate of occurrences.*

The reader is referred to a good example in Mich Bauer's paper: "Paranoid Penguin: Practical Threat Analysis and Risk Management." *Linux Journal,* Issue 93, March 2003.

7.6.1.2 Schneier's Attack Tree Method

Schneier approaches the calculation of risk analysis using a tree model he called an *attack tree*. An attack tree is a visual representation of possible attacks against a given target. The root of the attack forms the goal of the attack. The internal node from the leaves form the necessary subgoals an attacker must take in order to reach the goal, in this case the root.

The attack tree then grows as subgoals necessary to reach the root node are added depending on the attack goal. This step is repeated as necessary to achieve the level of detail and complexity with which you wish to examine the attack. If the attacker must pass through several subgoals in order to reach the goal, then the path in the tree from the leaves to the root is long and probably more complex.

Each leaf and corresponding subgoals are quantified with a cost estimate that may represent the cost of achieving that leaf's goal via the subgoals. The cheapest path in the tree from a leaf to the root determines the most likely attack path and probably the riskiest.

7.7 Vulnerability Identification and Assessment

A security vulnerability is a weakness in the system that may result in creating a security condition that may lead to a threat. The condition may be an absence of or inadequate security procedures and physical and security controls in the system. Although vulnerabilities are difficult to predict, no system is secure unless its vulnerabilities are known. Therefore, in order to protect a computer system, we need to be able to identify the vulnerabilities in the system and assess the dangers faced as a result of these vulnerabilities. No system can face any security threat unless it has a vulnerability from which a security incident may originate. However, it is extremely difficult to identify all system vulnerabilities before a security incident occurs. In fact, many system vulnerabilities are known only after a security incident has occurred. However, once one vulnerability has been identified, it is common to find it in many other components of the system. The search for system vulnerabilities should focus on system hardware, software, and also humanware as we have seen so far. In addition, system vulnerabilities also exist in system security policies and procedures.

7.7.1 Hardware

Although hardware may not be the main culprit in sourcing system vulnerabilities, it boasts a number of them originating mainly from design flows, imbedded programs, and assembling of systems. Modern computer and telecommunication systems carry an impressive amount of microprograms imbedded in the system. These programs control many functions in the hardware component.

However, hardware vulnerabilities are very difficult to identify, and even after they are identified, they are very difficult to fix for a number of reasons. One reason is cost; it may be very expensive to fix imbedded microprograms in a hardware component. Second, even if a vulnerability is inexpensive and easy to fix, the expertise to fix it may not be there. Third, it may be easy to fix, but the component required to fix it may not be compatible and interoperable with the bigger hardware. Fourth, even if it is cheap, easy to fix, and compatible enough, it may not be of priority because of the effort it takes to fix.

7.7.2 Software

Vulnerabilities in software can be found in a variety of areas in the system. In particular, vulnerabilities can be found in system software, application software, and control software.

7.7.2.1 System Software

System software includes most of the software used by the system to function. Among such software is the operating system that is at the core of the running of the computer system. In fact the vulnerabilities found in operating systems are the most serious vulnerabilities in computer systems. Most of the major operating systems have suffered from vulnerabilities, and intruders always target operating systems as they search for vulnerabilities. This is due to the complexity of the software used to develop operating systems and also the growing multitude of functions the operating system must perform. As we will discuss later, since the operating system controls all the major functions of the system, access to the system through the operating system gives the intruders unlimited access to the system. The more popular an operating system gets, the greater the number of attacks directed to it. All the recent operating systems such as Unix, Linux, Mac OS, Windows, and especially Windows NT have been major targets for intruders to exploit an ever-growing list of vulnerabilities that are found daily.

7.7.2.2 Application Software

Probably, the largest number of vulnerabilities is thought to be sourced from application software. There are several reasons for this. First, application software can be and indeed has been written by anybody with a minimum understanding of programming etiquettes. In fact, most of the application software on the market is written by people without formal training in software development. Second, most of the application software is never fully tested before it is uploaded on the market, making it a potential security threat. Finally, because software produced by independent producers is usually small and targeted, many system managers and security chiefs do not pay enough attention to the dangers produced by this type of software in terms of interface compatibility and interoperability. By ignoring such potential sources of system vulnerabilities, the system managers are exposing their systems to dangers of this software. Also security technologies are developing a lot faster than the rate at which independent software producers can include them in their software. In addition, since software is usually used for several years during that period, new developments in API and security tools tend to make the software more of a security threat. And as more reusable software becomes commonly used, more flaws in the libraries of such code are propagated into more user code. Unfortunately more and more software producers are outsourcing modules from independent sources, which adds to the flaws in software because the testing of these outsourced modules is not uniform.

7.7.2.3 Control Software

Among the control software are system and communication protocols and device drivers. Communication control protocols are at the core of digital and analog devices. Any weaknesses in these protocols expose the data in the communication channels of the network infrastructure. In fact, the open architecture policies of the major communication protocol models have been a major source of vulnerabilities in computer communication. Most of the recent attacks on the Internet and other communication networks have been a result of the vulnerabilities in these communication protocols. Once identified, these vulnerabilities have proven difficult to fix for a number of reasons. First, it has been expensive in some quarters to fix these vulnerabilities because of lack of expertise. Second, although patches have on many occasions been issued immediately after a vulnerability has been identified, in most cases, the patching of the vulnerability has not been at the rate the vulnerabilities have been discovered, leading to a situation where most of the current network attacks are using the same vulnerabilities that have been discovered, sometimes years back and patches issued. Third, because of the open nature of the communication protocols, and as new functional modules are added onto the existing infrastructure, the interoperability has been far from desirable.

7.7.3 Humanware

In Sect. 4.5.1, we discussed the human role in the security of computer systems. We want to add to that list the role social engineering plays in system security. Social engineering, as we saw in Chap. 3, is the ability of one to achieve one's stated goal, legally or otherwise, through the use of persuasion or misrepresentation. Because there are many ways of doing this, it is extremely difficult to prepare people not to fall for sweet talkers and masqueraders. Among the many approaches to social engineering are techniques that play on people's vulnerability to sympathy, empathy, admiration, and intimidation. Hackers and intruders using social engineering exploit people's politeness and willingness to help.

7.7.4 Policies, Procedures, and Practices

The starting point for organization security is a sound security policy and a set of security procedures. Policies are written descriptions of the security precautions that everyone using the system must follow. They have been called the building blocks of an organization's security. Procedures on the other hand are definitions spelling out how to implement the policies for a specific system or technology. Finally, practices are day-to-day operations to implement the procedures. Practices are implemented based on the environment, resources, and capabilities available at the site.

Many organizations do not have written policies or procedures or anything that is directly related to information security. In addition to security policies and procedures, security concerns can also be found in personnel policies and physical security procedures, for example, the protocols for accessing buildings and intellectual property statements.

The effectiveness of an organization's security policies and procedures must be measured against those in the same industry. Security policies and procedures are useless if they are applied to an industry where they are ineffective. When compared to a similar industry, weaknesses should be noted in quality, conformity, and comprehensiveness.

7.7.4.1 Quality

A policy or procedure has quality if it addresses all industry issues it is supposed to address. In addition to addressing all issues, policies and procedures are also tested on their applicability, that is, they are being specific enough in order to be effective. They are judged effective if they address all issues and protect system information.

7.7.4.2 Conformity

Conformity is a measure of the level of compliance based on the security policies and procedures. The measure includes how the policies or procedures are being interpreted, implemented, and followed. If the level is not good, then a security threat exists in the system. Besides measuring the level of compliancy, conformity also measures the effectiveness of the policies and procedures in all areas of the organization. A policy presents a security threat if it is not fully implemented or not implemented at all or not observed in certain parts of the organization.

7.7.4.3 Comprehensiveness

If the organization's security is required in a variety of forms such as physical and electronic, then the organization's security policy and procedures must effectively address all of them. In addition, all phases of security must be addressed including inspection, protection, detection, reaction, and reflection. If one phase is not effectively addressed or not addressed at all, then a security threat may exist in the system. Comprehensiveness also means that the policies and procedures must be widely and equitably applied to all parts of the system. And the policies and procedures must address all known sources of threats which may include physical, natural, or human.

7.8 Security Certification

Certification is the technical evaluation of the effectiveness of a system or an individual for security features. The defenses of a system are not dependent solely on secure technology in use, but they also depend on the effectiveness of staffing and training. A well-trained and proficient human component makes a good complement to the security of the system, and the system as a whole can withstand and react to intrusion and malicious code. Certification of a system or an individual attempts to achieve the following objectives that the system [5]:

- Employs a set of structured verification techniques and verification procedures during the system life cycle
- Demonstrates that the security controls of the system are implemented correctly and effectively
- Identifies risks to confidentiality, integrity, and availability of information and resources

7.8.1 Phases of a Certification Process

For the certification process to run smoothly, the following phases must be undertaken [5]:

- Developing a security plan to provide an overview of the system security requirements. The plan, as we have seen above, describes existing or planned security requirements and ways to meet them. In addition, the plan delineates responsibilities and expected behavior of individuals who access the system. The plan should be updated as frequently as possible.
- Testing and evaluation must be done, and it includes the verification and verification procedures to demonstrate that the implementation of the network meets the security requirements specified in the security plan.
- Risk assessment to determine threats and vulnerabilities in the system, propose and evaluate the effectiveness of various security controls, calculate trade-offs associated with the security controls, and determine the residual risk associated with a candidate set of security controls.
- Certification to evaluate and verify that the system has been implemented as described in the security policy and that the specified security controls are in place and operating properly. This provides an overview of the security status of the system and brings together all of the information necessary for the organization to make an informed and risk-conscious decision.

7.8.2 Benefits of Security Certification

In security, certification is important and has several benefits including:

- Consistency and comparability
- Availability of complete and reliable technical information leading to better understanding of complex systems and associated security risks and vulnerabilities

7.9 Security Monitoring and Auditing

Security monitoring is an essential step in security assurance for a system. To set up continuous security monitoring, controls are put in place to monitor whether a secure system environment is maintained. The security personnel and sometimes management then use these controls to determine whether any more steps need to be taken to secure the systems. The focus of the monitoring controls may depend on the system manager and what is deemed important for the system security, but in general control focuses on violation and exception reports that help the security personnel to determine quickly the status of security in the system and what needs to be done if the system is being or has been compromised.

Although monitoring decisions are made by the security administrator, what should be monitored and the amount of information logged are usually determined by either management or the security administrator. Also what should be included in the report and the details to be included to convey the best overall understanding of the security status of the system must be decided by the security administrator. It is not good, and in fact it is resource wasting to log too much information without being able to analyze it properly. Let us now focus on tools used to monitor, type of data gathered, and information analyzed from the data.

7.9.1 Monitoring Tools

There are several tools that can be used to monitor the performance of a system. The monitoring tool, once selected and installed, should be able to gather vital information on system statistics, analyze it, and display it graphically or otherwise. In more modern systems, especially in intrusion detection tools, the monitor can also be configured to alert system administrators when certain events occur. Most modern operating systems such as Microsoft Windows, Unix, Linux, Mac OS, and others have built-in performance monitors. In addition, there is a long list of independent security and system performance monitors that monitor, among other things, real-time performance monitoring and warehousing of event logs, real-time or delayed alerts to management, and customized performance reports that may include the history of the event and customized formats to allow quick legal proceedings and forensics analysis.

A variety of system monitoring tools are available, the majority of which fall into one of the following categories:

- System performance: This category includes most operating system performance loggers.
- Network security: This includes all IDS, firewalls, and other types of event loggers.
- Network performance and diagnosis: These are for monitoring all network performance activities.
- Networking links: To monitor the wiring in a network.

- Dynamic IP and DNS event logger.
- Remote control and file sharing applications event logger.
- File transfer tools.

7.9.2 Type of Data Gathered

Because of the large number of events that take place in a computer system, the choice of what event to monitor can be difficult. Most event loggers are preset to monitor events based on the set conditions. For example, for workstations and servers, the monitor observes system performance, including CPU performance, memory usage, disk usage, applications, system, security, DNS server, directory service, and File Replication Service. In addition, the monitor may also receive syslog messages from other computers, routers, and firewalls on a network. In a network environment, the logger may generate notifications that include e-mail, network popup, pager, syslog forwarding, or broadcast messages, to users or system administrator in real time following preset specified criteria. Further, the logger may support real-time registration of new logs, edit existing log registrations, and delete log registrations.

7.9.3 Analyzed Information

The purpose of a system monitoring tool is to capture vital system data, analyze it, and present it to the user in a timely manner and in a form in which it makes sense.

The logged data is then formatted and put into a form that the user can utilize. Several of these report formats are as follows:

- Alert is a critical security control that helps in reporting monitored system data in real time. Real time actually depends on a specified time frame. Time frames vary from, say, once a week to a few seconds. Once the alerts are selected and criteria to monitor are set, the alert tools track certain events and warn systems administrators when they occur.
- Chart is a graphic object that correlates performance to a selected object within a time frame. Most modern operating systems have Event Viewer that draws charts of the collected data.
- Log is the opposite of alerting in that it allows the system to capture data in a file and save it for later viewing and analysis. However, alerting generates a signal that it sends to the administrator based on the alert time intervals. Log information may also be used in a chart. Again most modern operating systems have log view tools.
- Report is a more detailed and inclusive form of system logs. Log reports provide statistics about the system's resources and how each of the selected system resource is being used and by whom. This information also includes how many processes are using each resource, who owns the process, and when he or she is using the resource. The timing of the generation of the report can be set, and the recipients of the report can also be listed.

7.9.4 Auditing

Auditing is another tool in the security assessment and assurance of a computer system and network. Unlike monitoring, auditing is more durable and not ongoing, and therefore, it is expensive and time-consuming. Like monitoring, auditing measures the system against a predefined set of criteria, noting any changes that occur. The criteria are chosen in such a way that changes should indicate possible security breaches.

A full and comprehensive audit should include the following steps:

- Review all aspects of the system's stated criteria.
- Review all threats identified.
- Choose a frequency of audits whether daily, weekly, or monthly.
- Review practices to ensure compliance to written guidelines.

7.10 Products and Services

A number of products and services are on the market for security assessment and audit. Hundreds of companies are competing for a market share with a multitude of products. These products fall under the following categories:

- Auditing tools
- Vulnerability assessment
- Penetration testing tools
- Forensic tools
- Log analysis tools
- Other assessment toolkits

Exercises

1. What is security assessment? Why is it important?
2. Discuss the necessary steps in analyzing the security state of an enterprise.
3. What is security assurance? How does it help in enterprise security?
4. What is security monitoring? Is it essential for enterprise security?
5. What is security auditing? Why is it necessary for system security?
6. What are the differences between security monitoring and auditing? Which is better?
7. What is risk? What is the purpose of calculating risk when analyzing security?
8. Give two ways in which risk can be calculated. Which is better?
9. What is social engineering? Why do security experts worry about social engineering? What is the best way to deal with social engineering?
10. Humanware is a cause of security threat. Discuss why this is so.

Advanced Exercises

1. Discuss any security surveillance system.
2. Discuss a good security auditing system.
3. Compare or discuss the differences between any two security systems.
4. Discuss human error or human factors as a major security threat.
5. What is the best way to deal with the security threat due to human factors?

References

1. Jamsa K (2002) Hacker proof: the ultimate guide to network security, 2nd edn. Onword Press, Albany
2. Holden G (2004) Guide to firewalls and network security: intrusion detection and VPNs. Delmar Thomson Learning, Boston
3. Kaeo M (1999) Designing network security: a practical guide to creating secure network infrastructure. Macmillan Technical Publishing, Indianapolis
4. Guidelines for the development of security plans for classified computer systems. http://cio.doe.gov/ITReform/sqse/download/secplngd.doc
5. Ross R. The development of standardized certification and accreditation guidelines and provider organizations. http://csrc.nist.gov/sec-cert/CA-workshop-fiac2002-bw.pdf
6. Kizza JM (2002) *Ethical and social issues in the information age*, 2nd edn. Springer, New York
7. Bauer M (2003) Paranoid penguin: practical threat analysis and risk management. Linux J 93:9

Additional References

1. Security architecture and patterns, KPMG. http://www.issa-oc.org/html/1
2. Threat analysis and vulnerability assessments. http://www.primatech.com/consulting/services/threat_analysis_and_vulnerability_assessments.htm