An Engineering Approach to Secure System Analysis, Design, and Integration

Edward Amoroso
W. E. Kleppinger
David Majette

A system security engineering (SSE) methodology is used within the Secure Systems Engineering Department of AT&T Bell Laboratories during the analysis, design, and integration of computer and network systems. This evolving methodology focuses on how threats, vulnerabilities, and attacks on these systems are identified and mitigated, and how safeguards based on engineering estimates of risk are identified and integrated.

Introduction: A Problem of Trust

In his book, *The Cuckoo's Egg,*[1] Cliff Stoll relates a whimsical tale about a group of villagers who casually leave their windows and doors unlocked in the evening when they go to bed. No one in the village worries about having anything in their home stolen or damaged, because a basic trust exists that everyone will respect the rights of others. One day, a stranger moves into the village and decides that it is his responsibility to break into everyone's homes and remove their belongings. This stranger eventually replaces these belongings and justifies his action by saying that it was intended to raise the awareness of the villagers to the potential catastrophes that might arise if they continued to leave their homes unlocked.

Based on this tale, computer scientists and network engineers might pose a fundamental question: As computing environments become more integrated and networked, can the computing analogy of the isolated, trusting village in Cliff Stoll's tale ever be realized again? Or have all computing environments become irreversibly vulnerable to the types of attacks by strangers that are so often justified on the grounds that they raise user awareness? And this, perhaps, is an even more disturbing question: Did closed, protected computing communities ever really exist?

The question of trust in computing and networking environments has prompted Bell Laboratories to explore approaches to protecting computing assets and resources from malicious attacks. During the past decade, these approaches have evolved into a comprehensive methodology based on experience with a variety of customers. These customers range from the Ballistic Missile Defense Organization (BMDO), for whom AT&T developed various types of security protection and approaches on the Global Protection Against Limited Strike (GPALS) system (formerly referred to in the press as the Star Wars system), to various investment banking and brokerage firms, which need to protect their transaction-based financial systems. (See Panel 1 for definitions of abbreviations, acronyms, and terms.)

This paper presents a brief technical summary and introduction to system security engineering (SSE). Descriptions of the SSE methodology and related technologies have appeared in various forums,[2] as well as in a recent broad survey of comparable methodologies.[3]

Threats to Computer Systems

It is well known that computer systems exist to provide a useful mechanism for processing, handling, storing, and transferring computing assets. These assets include the information that manages and controls computations, and the tangible hardware components that help accomplish the system mission. A *threat* to a computer system is defined as any unauthorized action that could adversely affect computer system assets. The primary threats include any unauthorized type of disclosure, change, blocking, or theft of assets.

Identifying threats to computer and network systems is a requirements analysis activity. Unlike most engineering-oriented approaches, however, such as structured

Panel 1. Abbreviations, Acronyms, and Terms

ARPA—Advanced Research Projects Agency, a government organization that initiated the Computer Emergency Response Team approach.

attack—a heuristic procedure that uses a set of vulnerabilities to enact a set of threats.

BMDO—Ballistic Missile Defense Organization, a government organization that promotes the use of system security engineering processes, such as AT&T's SSE process.

CERT—Computer Emergency Response Team, an approach to security risk management and disaster recovery.

countermeasure—a procedure or mechanism used to protect against or mitigate an existing security problem.

DISA—Defense Information Systems Agency, a government agency that promotes and funds security engineering activities.

GIS—AT&T Global Information Solutions

GPALS—Global Protection Against Limited Strike, a current reference to a project popularly known as "Star Wars," in which system security engineering processes based on AT&T's SSE process are being followed.

LAE—level of adversary effort, the relative degree of difficulty for a malicious intruder to cause security problems.

NTCB—network trusted computing base

Risks to the General Public—a public forum moderated by Peter Neumann of SRI International, in which accounts of security problems are maintained and published.

safeguard—a procedure or mechanism that lowers the degree of security risk for a potential set of threats, vulnerabilities, or attacks.

security risk—a measure of system-weighted penalty, squared, divided by the level of adversity effect.

security tiger team—a team of individuals experienced in security analysis and solutions, used to find system security weaknesses and holes.

SSE—system security engineering, a new engineering discipline being promoted by AT&T and other corporations as a means for systematically dealing with security problems.

SWP—system-weighted penalty, the relative criticality associated with a given security problem.

threat—a possible action on a computer or network system that could introduce damage, disclosure, denial, or theft to a system asset.

vulnerability—a system characteristic that may allow for a threat to be enacted on a system using a selected attack method.

analysis, prototyping, and formal methods, typical threat identification approaches follow an ad hoc, unstructured process. These processes can be improved by organizing brainstorming sessions, in which trained personnel can assemble a creative list of possible threats. Such brainstorming, however, is inherently subjective and may produce different results when performed in different environments with different personnel.

To offer a rigorous and repeatable method of threat identification for software and system engineering efforts, AT&T has adopted the use of a hierarchical technique known as a threat tree. This technique, which is detailed in Panel 2, offers the following advantages:

- Organizing threats hierarchically helps to determine the severity of each type of threat and its effect on other threats.
- Ensuring that each level of the threat tree includes a demonstrably complete set of threats can establish confidence that major areas of potential threats have not been ignored or missed.
- Once the leaf nodes of the threat tree have been established, the system security engineering team can focus its attention on completing the set of threats, using each branch of the tree as a helpful, heuristic guide.
- Including all levels of the threat tree in the threat analysis documentation can help determine the rationale and justification for the final list (i.e., those associated with the leaf nodes in the threat tree).

Constructing a threat tree is an important first step in the SSE process because it pinpoints the types of damage that system assets can incur. Too often, security safeguards are instituted before determining exactly what must be protected. Proposing security safeguards before threats are determined wastes money, causes unnecessary annoyance to system users, and adds little protection for critical assets.

Panel 2. The Threat Tree Technique

The threat tree technique in the AT&T SSE process creates a hierarchical structure to describe and justify the relationships between the various security problems identified for a given environment or application. To illustrate the technique, we introduce a typical electronic mail application showing the steps and principles involved in the creation and use of a threat tree.

As illustrated below at left, a typical electronic mail application includes the *originator* of the message (and the originator's local mail application software or user agent), the *recipient* of the message (and the recipient's user agent), and the *Message Handling System* (which, in addition to including the message transport system, may also include other services, such as a Mail Store). Also shown are *other subscribers* to the system and *external* factors (e.g., non-subscribers).

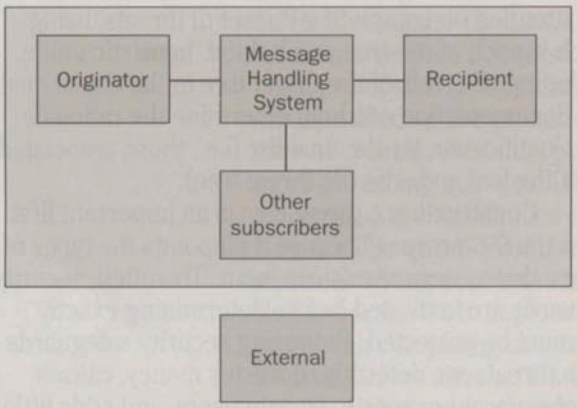
The threat tree for this model electronic mail system is also shown. At the highest level of the tree is the root node, labeled "Threats to Electronic Mail," which is defined to include all threats (i.e., it is complete). At the second level of the threat logic tree, the threats are decomposed into disclosure, integrity and denial/theft of service threats. Any potential threat must fall into one of these three classes, which implies that the second level is also complete. At the third level of the threat logic tree, each second-level node is decomposed into potential threat agents based on the architecture shown. Again, any potential threat agent must fall into one of these five classes, which preserves completeness.

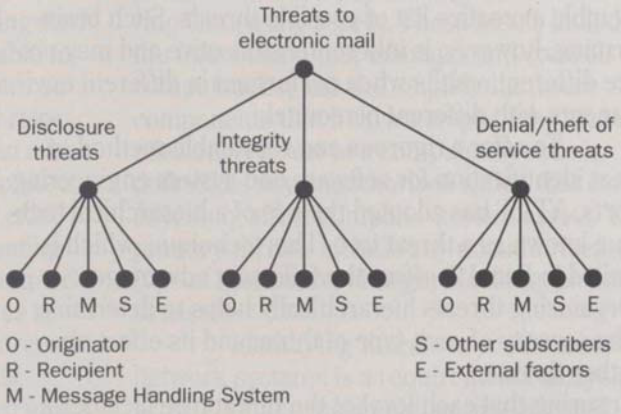
The electronic mail system can also be analyzed using the vulnerability views discussed in this paper. For example, under the personnel view, an originator can either accidentally or intentionally send messages to unauthorized recipients. Under the network view, a subscriber can forge a message from another user of the mail system.

Based on the identification of threats, vulnerabilities and attacks described, the security risks associated with each can be calculated using the empirical security risk formula given in this paper. To assess risks, a set of rules is defined for assigning values to estimates for LAE and SWP. Defining these rules makes the risk assessments reproducible and verifiable. For example, for the electronic mail system, we can define the rules:

SWP = {None: 0, Moderate: 5, Severe: 10},
LAE = {Easy: 1, Moderate: 5, Difficult: 10}.

(An LAE of zero is not allowed.) For example, it may be easy for a recipient to deny receiving a message (an integrity threat) with moderate damage to the system (the message can always be resent). The resultant risk is then Risk = 25. On the other hand, it may be difficult for other subscribers to forge messages (also an integrity threat) with severe damage to the system (based on actions taken on the message). The resultant risk is then calculated to be Risk = 10. From this assessment, the integrity/recipient threat would have higher priority than the integrity/other subscriber threat.





To illustrate, a customer of AT&T recently initiated a dialogue with its Global Information Solutions (GIS) account executives to determine if AT&T could provide a specific security solution that the customer had determined was necessary. This solution was potentially expensive to develop and had several negative implications on system usability, maintainability, and future migration of software applications.

Using the SSE process, members of the Secure Systems Engineering Department were able to direct this customer toward less expensive solutions (i.e., enhanced physical and operational procedures) that focused specifically on the assets to be protected, rather than on a fancy technical solution that did not address the right problem. Perhaps more importantly, AT&T earned the customer's good will, because their user community was spared the expense and aggravation of having to deal with a complex software solution that probably would have had no significant effect on the security of their application environment.

Vulnerabilities in Computer Systems

A *vulnerability* is defined as a system characteristic (usually a feature or a flaw) that may be exploited by a malicious intruder to cause a security threat. Most previous security analysis approaches in the security community focus primarily on identifying system vulnerabilities. Frequently, a security *tiger team*, a team of individuals experienced in security analysis and solutions, is used to find system security problems and holes. As with the threat analysis process, this approach has advantages and disadvantages.

As its primary advantage, a security tiger team often uncovers vulnerabilities that will not be found by other means. These individuals typically use ad hoc procedures and penetration tests to determine whether suspected vulnerabilities are present. Their efforts usually uncover subtle weaknesses and loopholes, perhaps in obscure locations, that could be exploited by attackers. Finding such subtle problems is especially important in open environments, for which a great deal of system-related information may be available to potential intruders.

Perhaps the most familiar example of such an open environment is the UNIX operating system, whose security-related vulnerabilities have been identified and are therefore available to attackers. (UNIX is a registered trademark of X/OPEN.) Competent security tiger teams are required to ensure that these well-known problems are not present in specific UNIX-based implementations. This may seem like a minor task, but the plethora of different UNIX versions and the frequent kernel and application customizations that have arisen complicate the task considerably. In addition to identifying known vulnerabilities, security tiger teams also typically attempt to determine if the possibly unique UNIX system configuration introduces problems that have not been previously noticed in other environments.

The primary disadvantage of using a security tiger team approach for identifying vulnerabilities is that it may not produce a complete or consistent set of results. For instance, once a security tiger team has identified a set of potential vulnerabilities for a customer, it is common for this customer to inquire about the likelihood that other vulnerabilities may surface in the future. Security tiger teams can rarely offer much assurance in response to such customer concerns.

As a result, AT&T has determined that an engineering approach can be used not only to identify threats, but also to pinpoint potential vulnerabilities and complement the activities of any security tiger teams. In particular, examining vulnerabilities from several points of view offers a structured means for organizing the vulnerability analysis. Specific vulnerability views that are examined in the context of an AT&T SSE process include the:

- Personnel view,
- Physical view,
- Operational view,
- Communications view,
- Network view,
- Computing view, and
- Information view.

These are described in more detail in the paragraphs that follow.

Personnel View. This area of vulnerability analysis is often neglected, especially by security tiger teams or experts who tend to focus on technical vulnerabilities. Disgruntled employees, for example, are a primary vulnerability if they are in a position to damage a given system. As a result, personnel-related issues identified during the SSE process are particularly important.

Physical View. Another often-neglected area, the physical view includes all facility and site-oriented con-

cerns, such as protecting computing resources from physical and environmental damage. All too often, forums such as Peter Neumann's Risks to the General Public Forum[4] report that—after organizations integrate expensive functional security mechanisms—damage often results from physical problems, such as unlocked doors to computer labs, and malfunctioning air conditioners or electrical power units.

Operational View. This area involves the disclosure of information that results from the procedures or operations used by certain individuals within an organization. It ranges from disclosing a password by means of neglect (e.g., writing it down or typing it in a slow, deliberate manner when an intruder is present) to more subtle operational issues, such as divulging information by allowing key personnel (e.g., the heads of different companies) to gather in an observable or public location. For example, if a company is rumored to be for sale and key AT&T personnel are seen entering that company's headquarters, observers can deduce sensitive information.

Communications View. This view is more specific than the general network view described in the next paragraph. In the communications view, vulnerabilities are examined in the context of the specific transmission media and associated devices (often hardware-based) that pass information between remote locations. A typical vulnerability is the open transmission of unencrypted information using media that is easily tapped by intruders.

Network View. Many vulnerabilities are introduced when computing systems are arranged in a local- or wide-area network configuration. Most vulnerabilities associated with networks can be traced to shortcomings in network components, such as routers, hubs, and bridges that do not support desirable access control and audit collection features. Other types of network vulnerabilities focus on a weakest link investigative approach, in which the SSE process tries to locate unprotected entry points in the network. Internet gateways, in particular, generally represent fruitful entry points for malicious intruders. Above all others, the network view may be the most difficult to substantiate completely. In many cases, defining the actual bounds and configuration of a network represents an intractable problem. Typically, a perimeter is defined instead, and vulnerabilities are identified within its confines.

Computing View. Perhaps the best known, this view focuses on the vulnerabilities that may be present in the operating system and associated application infrastructure. The security tiger team approach mentioned earlier, combined with threat trees and other investigative approaches, provides a good start at addressing the computing view of security vulnerabilities in a reasonable manner.

Information View. This final view is of more recent interest to many government organizations, such as the Defense Information Systems Agency (DISA) of the U.S. Government. The term INFOSEC, used in agencies such as DISA, comprises a system view that examines vulnerabilities based on the manner in which information is created, disseminated, stored, and transferred. It is less oriented toward specific architectural components and, as such, tends to uncover more end-to-end scenarios of possible vulnerabilities. For example, certain information vulnerabilities may only become evident in the combined context of information about computing, networking, and communications issues.

Attacks to Computer Systems

Security *attacks* to computer systems are defined as heuristic procedures that exploit vulnerabilities in a manner that causes a threat. If a malicious intruder became aware of a system vulnerability, he could design an attack that might harm the system. This situation implies, as one might already suspect, that the most successful attackers are generally also the most knowledgeable about the system being targeted.

Creating properly engineered, general taxonomies of attacks is not easy, because the details of a particular attack will always be system-specific. If, for example, one version of an operating system has a design flaw that allows users to misuse some feature, this represents an attack that must be identified and mitigated. Certainly, any taxonomy of attacks created for that target system must include this attack. As a result, including the attack in a general taxonomy is only reasonable if the associated flaw can be viewed as potentially repeatable on different systems.

Attacks are identified in the AT&T SSE process based on the observation that obtaining the most complete set of attacks for a given system will require at least the following two activities:

- A detailed comparison of the system environment, with taxonomies of attacks that have occurred previously in similar environments. This helps ensure that known attacks are avoided.
- A detailed investigation and penetration analysis by individuals who have intimate knowledge of the target system. This provides added confidence that the system does not allow obvious attacks. Depending on the qualifications of the individual performing the testing, this process may also include attention to certain subtle attacks as well.

Creating a detailed analysis with respect to existing attacks requires a suitable taxonomy. AT&T has had success in previous SSE efforts using an attack taxonomy introduced by Peter Neumann and Donn Parker of SRI International,[5] which is based on reported attacks in the Risks to the General Public Forum, moderated by Neumann (cited earlier). The taxonomy classifies attacks into nine general categories, to enable system security engineers to examine target systems and determine if any of these attack categories might apply.

During the past decade, AT&T has also successfully performed SSE analyses on UNIX-based operating systems using extensive taxonomies of reported internal attacks that have been created to test the UNIX operating system. Many of these attacks have been reported in forums such as the Computer Emergency Response Team (CERT) advisory, which has been funded and administered by the U.S. Advanced Research Projects Agency (ARPA) since 1988.

Unfortunately, this approach does not adequately ensure that all possible attacks have been identified. The study of previously reported attacks certainly provides confidence that those types of attacks are being addressed, but it does not guarantee that new types of attacks can be avoided. In addition, penetration testing relies too much on the qualifications of the individuals performing the testing. If the tests are inadequate (and provision of penetration test adequacy guidelines is difficult), then the reported attack analysis results will also be inadequate.

Nevertheless, the reported approach provides the best available technology for identifying potential attacks. The risk introduced by the drawbacks mentioned here may be alleviated somewhat by including as many relevant taxonomies as possible, as well as by using highly knowledgeable penetration testers.

Estimating Security Risks

Once the AT&T SSE process has identified the set of threats, vulnerabilities, and attacks for a given system, the system security engineer must consider the relationship between the various threats and their relative priorities. In the absence of a well-defined process, the system security engineer would be forced to determine, without guidance, how to alleviate an unprioritized list of security problems. This issue would be less critical if every system manager had sufficient resources and funds to mitigate every security problem completely. Even in such cases, however, inevitable tradeoffs would have to be considered between security and other important system attributes, such as usability.

To address this issue, the AT&T SSE process incorporates an analytical step in which the security *risk* associated with all threats, vulnerabilities, and attacks is estimated using a mathematical model that incorporates the following factors:

- Level of adversary effort (LAE)
- System-weighted penalty (SWP), and
- Calculated security risk.

Level of Adversary Effort. How easily might a malicious intruder enact a given threat, access a given vulnerability, or complete a given attack? To estimate the degree of ease associated with a given threat, vulnerability, or attack, the AT&T SSE process examines the number of possible attacks that can be used to enact that threat or exploit that vulnerability. This estimate is referred to as the *level of adversary effort* (LAE). Roughly, the LAE increases in proportion to how difficult it is for an intruder to attack a system.

System-Weighted Penalty. The relative criticality associated with the assets affected by the threat, vulnerability, or attack must also be estimated. Generally, this estimate is based on how much impact the removal of that asset would have on the successful performance of the system mission, and is further adjusted according to the relative criticality of that mission. Assets associated with life-critical applications, for example, would generally be viewed as more critical than comparable assets associated with computer games. The resultant criticality estimate is referred to in the AT&T SSE process as the *system-weighted penalty* (SWP).

Security Risk. The *security risk* of a particular threat or vulnerability is calculated from the LAE and SWP estimates using the empirical formula:

$$Risk = SWP^2 \div LAE$$

As SWP increases (i.e., the associated asset is more critical to the system), its security risk increases. Similarly, as LAE increases (i.e., it becomes more difficult for a malicious intruder to cause security problems), the security risk decreases. In the formula used, SWP is squared. AT&T network systems analysts, using an empirical process based on analysis of attacks and faults in the AT&T long distance network, have recommended this relation as having the desired effect. Using this formula on a variety of SSE efforts has confirmed the validity of the relation.

Associating risks with all threats, vulnerabilities, and attacks offers a means of prioritizing the security problems for a given system. As one might expect, it is better to start with the most serious problem in any effort to improve the security of a computer or network system.

Some useful heuristics that should help the system security engineer to prioritize threats, vulnerabilities, and attacks include:

- Only prioritizing threats may relegate all vulnerabilities and attacks to the same priority. Therefore, high-priority threats should be viewed in relation to the vulnerabilities and attacks that caused them.
- Prioritizing vulnerabilities results in a system-architecture-oriented approach to security mitigation. This has been found useful for proposed system architectures that are being analyzed for security enhancement.
- Prioritizing attacks results in a hierarchy of security issues that will have varying effects on the system. This approach is often used for stable architectures that are not being considered for security retrofit (other than finding quick fixes for serious problems).
- Prioritizing everything, the most desirable approach, is also the most time-consuming and prompts the most questions. In vulnerabilities associated with multiple threats, for example, do we assign the maximum of the associated threat priorities, or some new priority that reflects its association with multiple threats? A similar question arises for attacks and associated vulnerabilities. The answers to these questions are typically generated based on engineering consensus and the unique characteristics of the system being analyzed.

Identifying Security Solutions

Once a prioritized collection of security problems—either in the form of threats, vulnerabilities, attacks, or some combination of these—has been identified for a given system environment or application, security solutions must be identified to mitigate the associated risks. Such mitigation ranges from complete avoidance of a particular problem to a slight alleviation of the effects of that problem. One difficulty in identifying security solutions, discussed later, is how difficult it is to accurately determine the exact mitigation effect of a particular solution.

Certainly, any number of specific security solutions can be identified for all types of security problems that might arise on computer and network systems. Security solutions can be partitioned into two primary types: security *safeguards* and *countermeasures*. Security safeguards are roughly those solutions designed into a system in anticipation of a potential problem, whereas security countermeasures are responses to a detected or suspected security problem. This distinction is often blurred by many security engineers, but it is useful for explaining the SSE process to customers.

Security solutions can also be partitioned into those realized using automated security *mechanisms,* and those realized using non-automated operational security *procedures*. For example, a database could be protected by automated access control mechanisms, or by non-automated procedural guidelines (e.g., company policies).

Although it is not possible to describe all types of security safeguards and countermeasures available, some of the more common and useful approaches include:

- Security policy,
- Authentication,
- Access control,
- Auditing and intrusion detection,
- Trusted computing base (TCB), and
- Network security.

The discussions that follow focus on the manner in which security risks are mitigated using these safeguards. Reference 6 describes additional details of safeguard technology.

Security Policy. The collection of security requirements incorporated into a computer or network system is generally referred to as a security policy. To support security policy, the computer security community has promoted the notion of a *reference monitor,*[7] an abstract model of the mediation provided by a secure system against potential attacks. A security policy thus provides a detailed formal or informal specification of the reference monitor requirements for a system. AT&T system security engineers have found that customers rarely have well-defined security policies for their computing and networking environments, and would benefit from any effort to resolve this problem.

Authentication. Typically, before computing or networking services are granted to a requesting user, most systems require specific information from the user to verify his or her identity. Because this information should not be trusted (a malicious intruder might try to lie about his or her identity), systems generally require proof that this identity is accurate. The authentication step is used to ensure that reported identities are correct.

The most frequently used approach involves simple passwords, but more sophisticated schemes exist, including the use of personal identification number (PIN) protected smart card or hand-held authenticators. In such schemes, it is common for the system to provide a challenge (usually a string of characters) to the user, who then must enter the challenge as input to the hand-held device. The device calculates the value of some predetermined computation using the challenge and returns the results to the system, which also performs the same computation. If the results match, then the system concludes that the user is authentic (or that an attacker has stolen the hand-held device).

AT&T system security engineers commonly work with customers to ensure that the optimal authentication approach is being used, especially in environments with many network entry points.

Access Control. Earlier, we mentioned the reference monitor model to help describe the mediation that secure systems often provide. This model is implemented by two types of access control: *discretionary access control* (DAC), in which users have the ability to control who can and cannot access their information and resources; and *mandatory access control* (MAC), in which users do not have this ability.

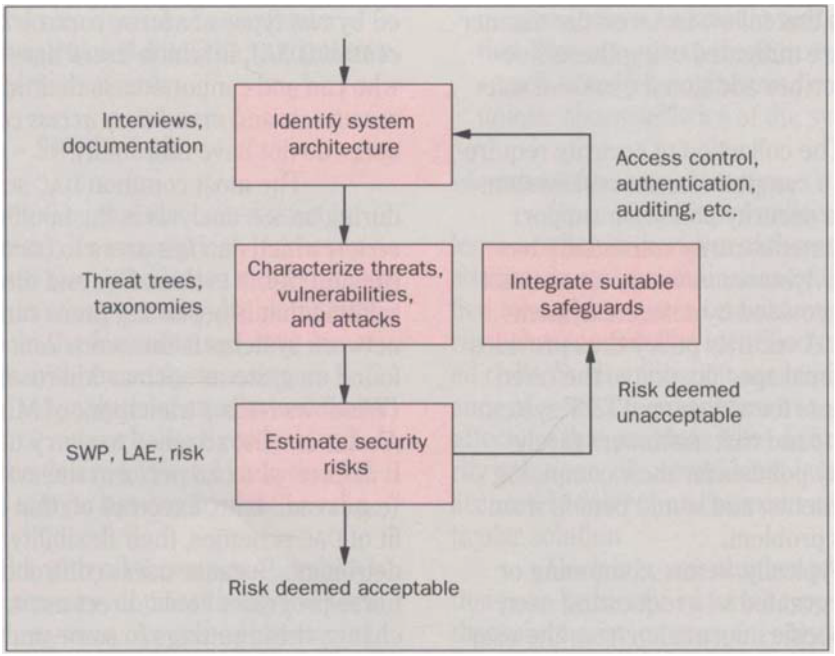
The most common DAC scheme encountered during an SSE analysis is the familiar UNIX *permissions vector*, which enables users to control read, write, and execute rights to their files and directories. A similar DAC scheme that is becoming more common in computer and network systems is the *access control list* (ACL) feature, found on systems such as Microsoft's Windows NT. (Windows NT is a trademark of Microsoft Corporation.) An ACL is a list attached to every object to be protected. It defines who can perform the available types of access (e.g., read, write, execute) on that object. The main benefit of DAC schemes, their flexibility, can also be their main detriment. Because users control the settings, Trojan horse programs could direct users to inadvertently change these settings to some undesirable configuration.

In the most common MAC scheme, internal security label representations are included on a system for all information repositories (e.g., files and directories) and active invoking agents (e.g., processes). A security policy then defines the mediation requirements to be enforced based on these security labels. For example, processes with low-security labels might not be allowed to read files with high-security labels. This type of mediation often requires operating system enhancements or explicit kernel functionality.

Because MAC is controlled by system or security administrators, its normal users cannot be fooled by Trojan horse attacks into changing MAC settings. It is therefore a desirable choice in many SSE efforts for protecting highly sensitive information. It is also, however, less flexible than DAC, and users have complained about the distinct reduction in system usability that often accompanies migration to a MAC scheme.

Auditing and Intrusion Detection. The security safeguards described earlier are all considered intrusive—they affect user operations directly. A less intrusive technology available to the system security engineer involves on-line auditing of all security-relevant activity into a protected log for later perusal. This deters intruders who do not want to get caught (as opposed to so-called kamikaze,

Figure 1. The system security engineering process.



or truck bomb, attacks), but its non-intrusive aspect makes it a poor choice for avoiding potentially catastrophic security problems.

In addition, auditing schemes may produce such voluminous quantities of information (e.g., often several megabytes of information on a daily basis) that system and security administrators would be hard-pressed to go through this data manually in a reasonable time frame. As a result, automated data reduction, data summary, and intrusion-detection approaches and tools have been developed to scan, process, and interpret audit data.

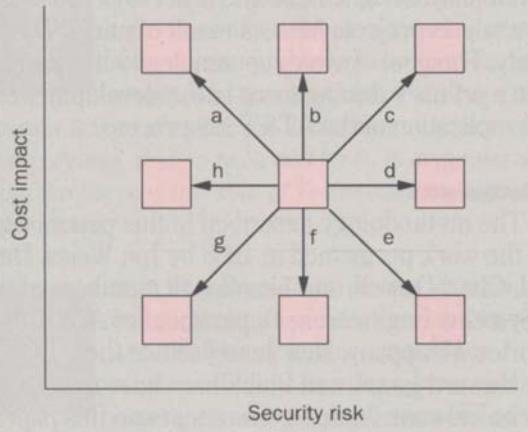
The most common intrusion-detection algorithm involves capturing expected behavior in an internal representation that is often referred to as a profile. Profiles are created based on the best available predictions of system or user behavior. Then, audit data, which represents what was observed, can be filtered into a similar internal representation so it can be compared with the associated profile. If expected behavior differs from observed behavior, then a problem might exist, precipitating alarms or other warnings. Additional details on such schemes are available in many works.[8]

Trusted Computing Base. The software (and hardware) mechanisms that implement the security safeguards described earlier must be analyzed and validated to operate properly; if they are not, the security aspects of a system would have to be viewed as suspect. System security engineers have therefore developed what they call a trusted computing base (TCB), which localizes all security safeguard implementations for a given system. A TCB is designed to minimize complexity, ensure tamper-proof operation, and avoid bypass approaches. In networked environments, identifying a network TCB (or NTCB) is complicated by the complexities typical of a distributed architecture.

Network Security. All the safeguard solutions discussed earlier refer to any type of application, including networking, database, embedded, and real-time. Nevertheless, because networking has become such an important application, especially in light of the explosive growth of the Internet, certain security safeguards have arisen that are most commonly found in network systems. In particular, encryption is usually included in network applications that require secrecy in communications. By encrypting transmissions, users reduce the likelihood that attackers can listen to their data or voice conversations.

Ingenious algorithms have also been developed to maximize the usefulness of encryption. By applying a key (or parameterization) to the encryption function, users can set up single-key transmissions in which both the sender and receiver of information share a key that is hidden to all others. This procedure ensures secrecy and

Panel 3: Candidate Security



Each of the labeled paths in the graph above describes a decision path about security for a given baseline architecture (the box at the center of the figure).

Decision Path a — A change to the baseline architecture that requires an expenditure, but reduces the security risk. This is generally viewed as an acceptable decision path.

Decision Path b — An expenditure for a change that has no impact on security risk.

Decision Path c — An increase in security risk and an associated positive expenditure.

Decision Path d — An increase in security risk with no associated expenditure.

Decision Path e — An increase in security risk and an associated reduction in expenditure for the change. This is sometimes viewed as an acceptable decision path.

Decision Path f — A reduction in cost with no effect on security. This is a desirable path.

Decision Path g — A reduction in security risk and a reduction in expenditure. This is the most desirable decision path of all.

Decision Path h — A reduction in security risk with no associated expenditure. This is a desirable path.

authentication (i.e., the sender must be authentic if the proper key was used to encrypt the transmission).

In single-key solutions, however, complex networking environments, such as the Internet, require too many keys. As a result, two-key transmissions exist in which a public key, known to all, and a private key, known only to its owner, are required to encrypt and decrypt a message. Thus, senders can encrypt with the public key of the recipient to ensure secrecy, because only the receiver can decrypt that message. Similarly, senders can encrypt with their private key to ensure authentication, because no other entity owns that key.

This scheme, attributed to Diffie and Hellman,[9] has been implemented in a variety of ways, the most famous of which is the Rivest–Shamir–Adleman (RSA) algorithm.[10] In the RSA scheme, straightforward number-theoretic notions are used to create public and private keys in an efficient, arguably secure manner.

AT&T system security engineers have actively assisted customers with the problems associated with Internet connectivity. Firewalls—typically routers that filter Internet protocol (IP) packets, among other things—are useful safeguards in this area.

Integrating Security Solutions

Although identifying candidate security solutions is a critical step in the AT&T SSE process, it is usually not until the security solution integration step has been initiated that progress is visible in a given system, environment, or application. This vital step in the process determines an optimal selection and integration of the best available security technology into the existing target system. As a system is being conceptualized, the integration is targeted at the system requirements. If the system already exists, however, and its security properties are to be analyzed, the integration must be aimed at the existing characteristics and attributes of that system.

Because this integration mainly involves system-specific considerations, it is difficult to generalize about the exact placement of security solutions in new or existing systems. In making these integration decisions, however, the system security engineer should be guided by five particularly significant factors—criticality, cost, threat, impact, and migration.

Criticality. The most critical components of an architecture should generally be viewed as higher-

priority candidates for safeguard integration than lower-priority components. This concept is difficult to quantify. From a qualitative perspective, however, engineering judgment and consensus based on previous experiences are generally considered a feasible approach.

Cost. Safeguard integration is an iterative process that terminates when all security threats have been suitably mitigated, or when all available funds and resources have been spent. In most previous applications of the AT&T SSE process, customers have typically depleted their funds before all security threats have been mitigated. Figure 1 shows a natural life cycle for integrating safeguards.

Threat. Safeguards must be selected to optimally counter available threats. Different safeguards may have different effects on a threat. For example, mandatory access control and auditing are both preventive mechanisms, but in slightly different ways. Mandatory access control mediates requests using explicit functional mechanisms, whereas auditing mediates requests indirectly by increasing the likelihood that an intruder will get caught. For threats that are potentially catastrophic, however, a system or security manager might prefer the mandatory nature of the access control mechanism, rather than the punitive nature of auditing. Panel 3 shows some of the decision paths involved in countering the effects of threats that may occur while integrating security safeguards.

Impact. Security safeguards should be expected to have some impact on certain aspects of a system or environment. The most obvious area of such an impact is the usability attribute. Security features should be selected to minimize unnecessary and unwelcome effects on any critical attribute. In fact, to be accepted in most computer and network environments, security integration must minimize its impact on user operations, system performance, system resources, and application integrity.

Migration. The last principle of security safeguard integration is that provision for future migration of both security and nonsecurity features must be included in the integration strategy.

Conclusion

This paper has presented the salient aspects of the AT&T SSE process, its component steps, and technical concerns. One aspect of this process that should be evident to the reader, but has not been stressed in this discussion, is the relative simplicity of the process with respect to the more baroque techniques sometimes encountered in the security community. This simplicity was intentionally designed into the process because most security analysis projects fail as a result of unmanaged complexity. The goal of avoiding complexity has therefore been a primary driving force in the development and practical application of the AT&T SSE process.

Acknowledgments

The methodology described in this paper originated in the work performed in 1988 by Jon Weiss, Dan Goddard, Cheri Dowell, and Lina So, all members of the Secure Systems Engineering Department of AT&T Bell Laboratories, Whippany, New Jersey. Since then, Don Gazzale, Howard Israel, and Phil Sikora have made additional technical contributions. Comments on this paper by Jack Lahti and Bennett Karp were appreciated.

References

1. C. Stoll, *The Cuckoo's Egg*, Doubleday, New York, 1989.
2. J. Weiss, "A System Security Engineering Process," *Proceedings of the 14th National Computer Security Conference*, Washington, D.C., October 1–4, 1991, pp. 572–581.
3. R. Baskerville, "Information Systems Security Design Methods: Implications for Information Systems Development," *ACM Computing Surveys*, Vol. 25, No. 4, December 1993, pp. 375–414.
4. P. Neumann, "RISKS: Cumulative Index of Software Engineering Notes — Illustrative Risks to the Public in the Use of Computer Systems and Related Technology," *ACM Software Engineering Notes*, Vol. 14, No. 1, 1989.
5. P. Neumann and D. Parker, "A Summary of Computer Misuse Techniques," *Proceedings of the 12th National Computer Security Conference*, Baltimore, Maryland, October 10–13, 1989, pp. 396–407.
6. E. Amoroso, *Fundamentals of Computer Security Technology*, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.
7. J. Anderson, *Computer Security Technology Planning Study, ESD-TR-73-51, Vol. I and II*, Air Force Electronic Systems Division (available from NTIS: AD758206), 1972.
8. D. Denning, "An Intrusion Detection Model," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, April 7–9, 1986, pp. 118–131.
9. W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, November 1976, pp. 644–654.
10. R. Rivest, A. Shamir, and L. Adelman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120–126.

(Manuscript approved June 1994)

Edward Amoroso *is a distinguished member of technical staff in the Secure Systems Engineering Department of AT&T Bell Laboratories in Whippany, New Jersey. He is currently involved in a variety of different security-related research and development efforts. Mr. Amoroso joined AT&T in 1985, and holds a B.S. in physics from Dickinson College, in Carlisle, Pennsylvania, and an M.S. and Ph.D. in computer science from the Stevens Institute of Technology, in Hoboken, New Jersey.*

W.E. Kleppinger *is a member of technical staff in the Secure Systems Engineering Department of AT&T Bell Laboratories in Whippany, New Jersey. He is currently performing system security engineering on a variety of government and international projects. Mr. Kleppinger holds an A.B. from Princeton University in New Jersey, and M.S. and Ph.D. degrees from Stanford University in Palo Alto, California, all in physics. He joined AT&T in 1985.*

David Majette *is a supervisor in the Secure Systems Engineering Department of AT&T Bell Laboratories in Whippany, New Jersey. He is currently managing several domestic and international system security engineering efforts and software security tasks. He has also been a developer on the UNIX System V/MLS operating system and several of AT&T's secure computing and networking applications. Mr. Majette joined AT&T in 1968, after receiving a B.S. in physics from William and Mary College, in Williamsburg, Virginia.*