Chapter 5

Perform Vulnerability and Threat Analyses

This chapter describes the second component of an effective information security/IA program — performing vulnerability and threat analyses. Outputs from the previous component, defining the system boundaries, are used during the vulnerability and threat analyses. To conduct vulnerability and threat analyses, the following activities are performed:

- IA analysis techniques are selected and used.
- Vulnerabilities, their type, source, and severity are identified.
- Threats, their type, source, and likelihood are identified.
- Transaction paths, critical threat zones, and risk exposure are evaluated.

These activities are conducted in a sequential and iterative manner, as explained in the following discussion. Again, all stakeholders should be involved in these activities.

5.1 Definitions

On occasion, the terms "vulnerability" and "threat," "hazard," and "risk" are used interchangeably. These terms are related; however, they have distinct meanings. Vulnerability is defined as[362]:

> *a weakness in a system that can be exploited to violate the system's intended behavior relative to safety, security, reliability, availability, integrity, and so forth.*

Vulnerabilities are inherent in the design, operation, or operational environment of a system. They accrue as a result of errors of omission, errors of commission, and operational errors that occur during the life of a system.

Threat is defined as:[362]

> *the potential danger that a vulnerability may be exploited intentionally, triggered accidentally, or otherwise exercised.*

In other words, a vulnerability is a weakness that can be taken advantage of to violate system safety, reliability, and/or security, while a threat represents the potential to exploit that weakness.

Hazard is defined as:[56]

> *a source of potential harm or a situation with potential to harm.*

A hazard represents potential injury or death to humans, or damage or destruction to property or the environment.

Risk is defined as:[143]

> *(1) A combination of the likelihood of a hazard occurring and the severity of the consequences should it occur; (2) an expression of the possibility and impact of an unplanned event or series of events resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment in terms of potential severity and probability of occurrence.*

A hazard is an undesirable event with negative consequences, while risk represents the likelihood that the hazard will occur and the severity of the consequences thereof.

These four concepts are closely related, as shown in Exhibit 1. A vulnerability, or inherent weakness in a system, leads to one or more potential hazards. Hazards represent potential sources of harm or injury to individuals, property, or the environment. It is important to note that the harm or injury caused by a hazard may or may not be physical. For example, a system weakness that allows credit card information to be stolen results in financial harm. A hazard occurs when a threat is instantiated accidentally or intentionally. It is possible for more than one threat to trigger the same hazard. For example, the same hazard could be triggered accidentally or intentionally through different mechanisms. Risk is the composite of the likelihood of a threat being instantiated and the worst-case severity of the hazard consequences.

The use of the terms "severity" and "likelihood" also needs clarification. Severity characterizes the consequences of a potential hazard, the extent of harm or injury that could be inflicted. Following standard risk management practices, the worst-case scenario is evaluated. Most international standards[24,57,63,129–130,143] recognize four levels of severity:

- **Catastrophic:** fatalities or multiple severe injuries; loss of one or more major systems
- **Critical:** single fatality or severe injury; loss of a major system

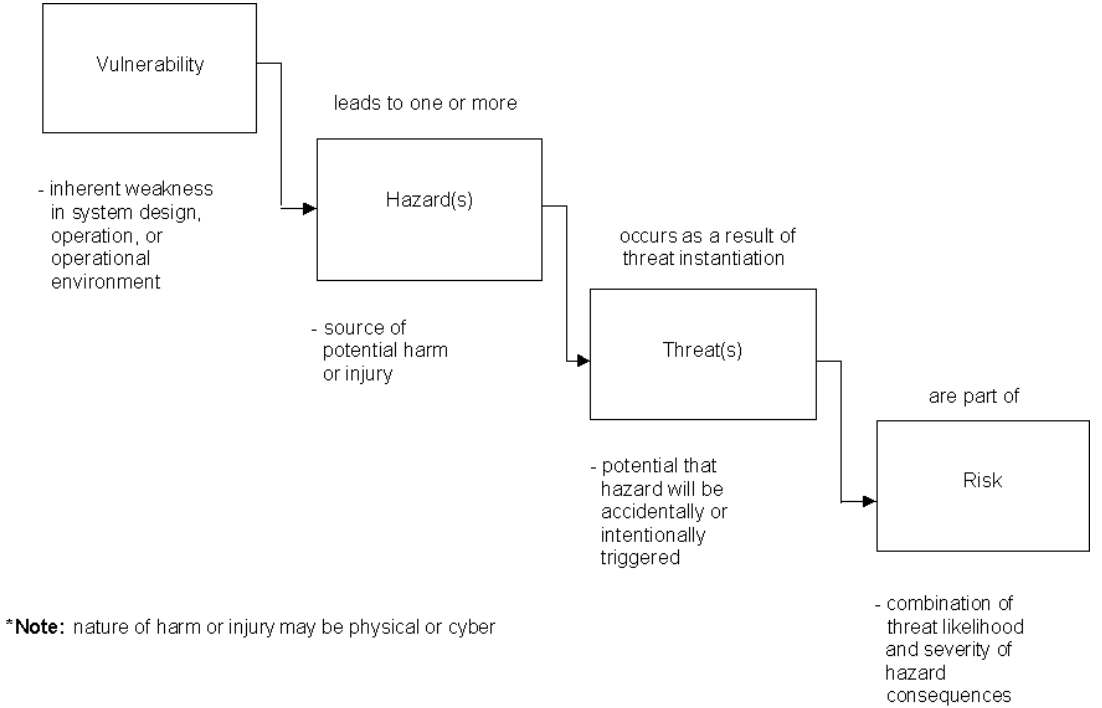

Exhibit 1 Interaction Between Vulnerabilities, Hazards, Threats, and Risk

- **Marginal:** minor injury; severe system(s) damage
- **Insignificant:** possible single minor injury; system damage

Remembering that injury refers to potential harm that may or may not be physical in nature, severity levels can be applied to the full range of IA safety, reliability, and security concerns.

Likelihood characterizes the probability of threat instantiation, that is, a hazard being effected. The most likely scenario is evaluated. Many international standards[24,57,63,129–130,143] recognize six levels of likelihood:

1. **Frequent:** likely to occur frequently; the hazard will be experienced continually (10^{-2})
2. **Probable:** will occur several times; the hazard can be expected to occur often (10^{-3})
3. **Occasional:** likely to occur several times over the life of the system (10^{-4})
4. **Remote**: likely to occur at some time during the life of the system (10^{-5})
5. **Improbable:** unlikely but possible to occur during the life of the system (10^{-6})
6. **Incredible:** extremely unlikely to occur during the life of the system (10^{-7})

Likelihood can be assessed qualitatively or quantitatively, depending on the nature of a system. Most international standards support both types of assessments. The quantitative assessment of random hardware failures is straightforward. Systematic software failures, operational errors, and malicious intentional acts lend themselves to qualitative assessments.

As noted earlier, risk is the composite of threat likelihood and the severity of the consequences of a potential hazard should a vulnerability be exploited. Risk is evaluated for every potential vulnerability/threat pair. Vulnerabilities are often exposed as a result of an unusual unplanned combination of events occurring simultaneously or sequentially. For example, three isolated events could each be considered low risk; however, if they occurred simultaneously or immediately after one another, a high-risk scenario could result. Consequently, a system risk assessment must evaluate the likelihood and severity of individual events and combinations of events.

5.2 Select/Use IA Analysis Techniques

A variety of analytical techniques are employed to discover vulnerabilities in the specification, design, implementation, operation, and operational environment of a system, the potential hazards associated with these vulnerabilities, and the threat that these hazards will be triggered accidentally or with malicious intent. Some vulnerabilities can be identified through informal brainstorming sessions. However, a comprehensive exploration of vulnerabilities, hazards, and threats requires the use of more formal techniques.

Exhibit 2 lists 19 current proven IA analysis techniques. A description of each technique is provided in Annex B, which discusses the purpose, benefits,

Exhibit 2 Information Assurance Analysis Techniques

I. IA Analysis Techniques	C/R	Type	Life-Cycle Phase in which Technique is Used: Concept	Development	Operations
Bayesian Belief networks (BBNs)[b]	C1	All	x	x	x
Cause consequence analysis[a,b]	R1/C1	SA, SE	x	x	x
Change impact analysis	C1	All		x	x
Common cause failure analysis[a]	C1	All	x	x	x
Develop operational profiles, formal scenario analysis	C1	All	x	x	x
Develop IA integrity case	C1	All	x	x	x
Event tree analysis[a,b]	R1/C1	All	x	x	x
Functional analysis	C1	SA, SE	x	x	x
Hazard analysis	C1	SA, SE	x	x	x
HAZOP studies[a,b]	C1	SA, SE	x	x	x
Highlighting requirements likely to change	C1	All	x		
Petri nets[a,b]	C1	SA, SE		x	x
Reliability block diagrams	C1	RE	x	x	x
Reliability prediction modeling	C1	RE	x	x	
Response time, memory, constraint analysis	C1	All		x	x
Software, system FMECA[a,b]	C1	All	x	x	x
Software, system FTA[a,b]	R1/C1	SA, SE	x	x	x
Sneak circuit analysis[a,b]	C1	SA, SE		x	x
Usability analysis	C1	SA, SE	x	x	x

[a] These techniques can also be used during verification of the effectiveness of threat control measures.

[b] These techniques can also be used during accident/incident investigations.

Source: Adapted from Hermann, D., *Software Safety and Reliability: Techniques, Approaches and Standards of Key Industrial Sectors*, IEEE Computer Society Press, 1999.

Legend for Exhibit 2

Column	Code	Meaning
Type	SA	Technique primarily supports safety engineering
	SE	Technique primarily supports security engineering
	RE	Technique primarily supports reliability engineering
	All	Technique supports a combination of safety, security, and reliability engineering
C/R	Cx	Groups of complementary techniques
	Rx	Groups of redundant techniques; only one of the redundant techniques should be used

and limitations of each technique, and provides pointers to references for further information.

IA analysis techniques uncover safety, reliability, and security vulnerabilities. Some IA analysis techniques focus on one aspect, such as safety; others evaluate a combination of safety, reliability, and security concerns, as noted in Chapter 3. The type column in Exhibit 2 indicates the primary focus of each technique.

IA analysis techniques are employed iteratively throughout the life of a system: when a new system concept is being defined, during design and development, when a system is deployed and becomes operational, and as part of system upgrades or other maintenance activities. IA analysis techniques are also undertaken following a system failure/compromise and as part of a periodic reassessment of IA strategies. Vulnerabilities, hazards, and threats are monitored throughout the life of a system because of the potential for new vulnerabilities, hazards, and threats to be introduced or the status of known vulnerabilities, hazards, or threats to change at any point. The last three columns in Exhibit 2 indicate the generic life-cycle phase(s) in which a technique can be used most effectively.

Not all techniques are used at one time or for one system. Rather, a complementary set of techniques are employed that: (1) evaluate all aspects of a system's design, operation, and operational environment; and (2) comprehensively assess safety, reliability, and security issues. The set of techniques employed will vary, depending on the life-cycle phase. The C/R column in Exhibit 2 identifies groups of complementary and redundant techniques.

After the IA analysis techniques have been selected and the phase(s) when they will be used identified, the next step is to train the team that will use the techniques. It is essential that all team members thoroughly understand how to use the techniques and correctly interpret the results obtained from using them. Again, it is imperative to have all stakeholders participate; not all stakeholders will perform the analysis, but they should participate and review the results. Exhibit 3 cites the IA analysis role of each technique.

A high-level example is developed next that illustrates how to identify vulnerabilities using the input from the system boundary definition (Chapter 4). Exhibit 4 summarizes this process. An online banking system is evaluated in this example.

Input from System Boundary Definition

1. IA goals:
   Protect the privacy and integrity of customer records from accidental or malicious intentional unauthorized disclosure, manipulation, alteration, abuse, corruption, and theft. Protect personal identifying information: name, address, phone number, e-mail address, account number, and fax number. Protect customer account balance and transaction information.

2. System entity definition:
   Internal: Web servers
   Local LAN, workstations, printers
   Bank employees
   External: Links to other financial systems
   Links to other financial institutions
   Customers, their workstations, and ISPs
   Telecommunications backbone
   Power, environmental controls
   Maintenance and vendor staff
3. System operation characterization:
   normal modes and states:

Online	Offline
Start-up	Shutdown
Application-specific functions:	Reconfiguration
Customer logon/logoff	Restart/reset
Check account balance	Backup
Move funds between accounts	Standby
Pay bills	Maintenance
Order stocks or insurance	Decommission
Check if transaction has cleared	
Open CD	
Process and post customer transactions	
Charge fees for online banking services	

   Operational profiles (developed for):
   Customers
   Bank employees
   System administrator
   Vendor and maintenance staff
   Potential intruders
4. System entity control analysis:
   Total control: Bank employees
   Partial control: Web servers
   Local LAN, workstations, printers
   Links to other financial systems
   Links to other financial institutions
   Maintenance and vendor staff
   None: Customers, their workstations, ISPs
   Potential intruders
   Power, environmental controls
   Telecommunications backbone, ISP

The set of IA analysis techniques corresponds to IA goals and system specifics. In this example, the IA goals concern protecting the privacy and integrity of customer records. The emphasis is on information security and reliability; no safety goals are stated. From these goals it can be inferred that

Exhibit 3 Analysis Role of IA Techniques

Analysis Technique	*IA Analysis Role*
Bayesian belief networks (BBNs)	Provide a methodology for reasoning about uncertainty as part of risk analysis and assessment.
Cause consequence analysis	Enhance IA integrity by identifying possible sequences of events that can lead to a system compromise or failure.
Change impact analysis	Analyze *a priori* the potential local and global effects of changing requirements, design, implementation, data structures, or interfaces on system performance, safety, reliability, and security; prevent errors from being introduced during enhancements or maintenance.
Common cause failure (CCF) analysis	Enhance IA integrity by identifying scenarios in which two or more failures or compromises occur as the result of a common design defect.
Develop operational profiles, formal scenario analysis	Identify operational profiles, capture domain knowledge about MWFs and MNWFs; understand human factors safety, reliability, and security concerns.
Develop IA integrity case	Collect, organize, analyze, and report information to prove that IA integrity requirements have been (or will be) achieved and maintained.
Event tree analysis	Enhance IA integrity by preventing defects through analysis of sequences of system events and operator actions that could lead to failures, compromises, or unstable states.
Functional analysis	Identify safety and security hazards associated with normal operations, degraded mode operations, incorrect usage, inadvertent operation, absence of function(s), and accidental and intentional human error.
Hazard analysis	Enhance IA integrity by identifying potential hazards associated with using a system so that appropriate mitigation features can be incorporated into the design and operational procedures.
HAZOP studies	Prevent potential hazards (accidental and intentional, physical and cyber) by capturing domain knowledge about operational environment, parameters, modes/states, etc. so that this information can be incorporated in the requirements, design, and operational procedures.
Highlighting requirements likely to change	Enhance the maintainability of threat control measures and IA integrity.
Petri nets	Identify potential deadlock, race, and nondeterministic conditions that could lead to a system compromise or failure.
Reliability block diagrams	Enhance IA integrity by identifying diagrammatically the set of events that must take place and the conditions that must be fulfilled for a system or task to execute correctly[69,131]; support initial reliability allocation, reliability estimates, and design optimization.

Exhibit 3 Analysis Role of IA Techniques (continued)

Analysis Technique	*IA Analysis Role*
Reliability prediction modeling	Predict future reliability of a software system.
Response time, memory, constraint analysis	Ensure that the operational system will meet all stated response time, memory, and other specified constraints under low, normal, and peak loading conditions.[333]
Software, system FMECA	Examine the effect of accidental and intentional, random and systematic failures on system behavior in general and IA integrity in particular.
Software, system FTA	Identify potential root causes of undesired system events (accidental and intentional) so that mitigating features can be incorporated into the design and operational procedures.
Sneak circuit analysis	Identify hidden unintended or unexpected hardware or software logic paths or control sequences that could inhibit desired system functions, initiate undesired system events, or cause incorrect timing and sequencing, leading to a system compromise or failure.
Usability analysis	Enhance operational IA integrity by ensuring that software is easy to use so that effort by human users to obtain the required service is minimal[18]; prevent induced or invited errors that could lead to a system failure/compromise.

the primary transactions involve the acquisition, manipulation, storage, retrieval, and display of information and that the system is data (text) intensive with active user involvement.

Next, the system entity definition is used to identify high-level potential failure points. Both internal and external entities are considered. Failure points represent potential attack points. In this example, that would include:

- Web server failures
- Local LAN, workstation, or printer failures
- Failure of links to other financial systems
- Failure of links to other financial institutions
- Telecommunications backbone or ISP failures
- Faulty power source or environmental controls
- User actions (customers, bank employees, maintenance or vendor staff, potential intruders)

Failure scenarios are postulated for each potential failure point, using the system operational characterization and entity control analysis as input. The intent is to premise all the ways in which an entity could fail or be induced to fail, that is, perform contrary to specified intended functionality. In other words, what could go wrong with a system? How could a system "break" or be broken? Failure scenarios include safety and security compromises, and

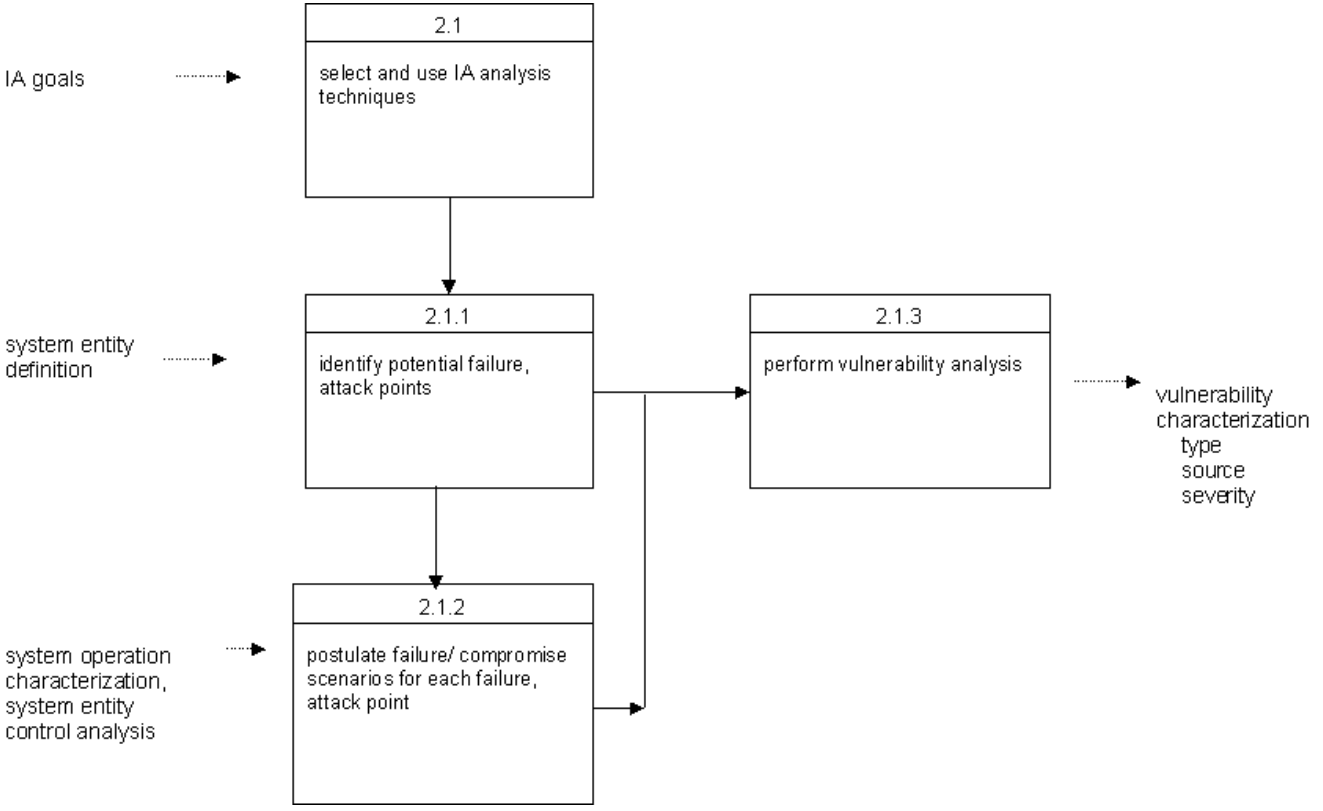

Exhibit 4 Vulnerability Identification Process

reliability degradation, in addition to the more well-known system unavailability, slowness, and crashes. Induced failures are the outcome of malicious intentional action. Other failures are the consequence of accidental errors or inaction during the specification, design, development, operation, or maintenance of a system.

In our example, failure scenarios are postulated for each of the seven potential failure points. For the sake of brevity, only the Web server failure scenarios are expanded. The Web server could fail or be induced to fail through a failure in the: server hardware, communications equipment and software, operating system, applications software, or DBMS. The server hardware could fail or be induced to fail through a fault/failure in: memory, the CPU, hard drive, power supply, back plane, bus, etc., as well as incorrect environmental controls or a faulty power source. The communications equipment and software could fail or be induced to fail through a fault/failure in the: gateway hardware and interfaces, communications protocols, routing table, etc. The Web server operating system could fail or be induced to fail through a fault/failure that: causes the system to become saturated, interferes with interrupt handling, corrupts the registry, causes buffer overflows, corrupts file and directory structures, overwrites user accounts and privileges, initiates conflicts between different software applications, etc. The web server application software could fail or be induced to fail through a fault/failure resulting from: functional errors, the inability to access the resources needed, erroneous command sequences, illegal data input, incompatibility with a new version of the server operating system or DBMS, etc. The server DBMS could fail or be induced to fail through a fault/failure which causes: data files to become corrupted, errors in query/retrieval functions, errors in display functions, errors when data is entered, modified, or deleted, etc. The crashing, slow operation, or unavailability of any of these five components also constitutes a failure scenario.

IA analysis techniques are used to further decompose failure scenarios to identify and characterize vulnerabilities. Exhibit 5 demonstrates the link between failure points, failure scenarios, and nine of the many potential vulnerabilities for the hypothetical high-level online banking example.

5.3 Identify Vulnerabilities, Their Type, Source, and Severity

IA vulnerabilities are classified three ways, as shown in Exhibit 6:

- The type of action that caused the vulnerability to manifest itself: accidental action (or inaction) or intentional malicious action or inaction
- The method by which the vulnerability is exploited: direct or indirect involvement on the part of the perpetrator
- The nature of the vulnerability or weakness: safety, reliability, security, or some combination thereof

A vulnerability may be the result of accidental or intentional human action. An accidental vulnerability is the result of an error of commission or an error

Exhibit 5 Correlation of Failure Points, Failure Scenarios, and Vulnerabilities

System: online banking

Failure Point	*Failure Scenario*	*Vulnerability*
Web server	Application software not protected	Little or no error detection/ correction or fault tolerance
Web server	Operating system saturated	If number of simultaneous users exceeds x, system becomes unstable and exhibits unpredictable behavior
Web server	DBMS data files corrupted	Data files can be accessed directly without going through DBMS applications software
Web server	Sporadic system shut down or unpredictable behavior	Server hardware subjected to extreme environmental conditions
User action	User authorizations not checked in order to speed up system response times; security compromised	End users and system administrator lack sufficient training, limited understanding of system security features and procedures
User action	Backups and archives generated sporadically or not at all; backups and archives not verified and are unreliable	Unsecure backups, archives
User action	Hardcopy printouts thrown in open trash bins; security compromised	Careless disposal of hardcopy printouts
User action	Portable equipment and storage media taken out of facility, occasionally lost or stolen; files from unknown/untrusted sources loaded onto system	No control over portable equipment or storage media
Web server	Conflicts between COTS applications cause unpredictable behavior; unauthorized user can access COTS applications	COTS components installed with default values, guest accounts, possible trap doors

of omission; that is, something was done wrong or not done during the specification, design, development, or operation of a system. Failing to specify illegal or incompatible system modes and states, so that the system is not prevented from entering them, is an example of a vulnerability that results from an error of omission. If the illegal and incompatible system modes and states were specified, but incorrectly, leaving the system unprotected, that would result in a vulnerability from an error of commission. As Lindquist and Jonsson[335] note:

> *Vulnerabilities may also be introduced by changes in the system environment or the way the system operates.*

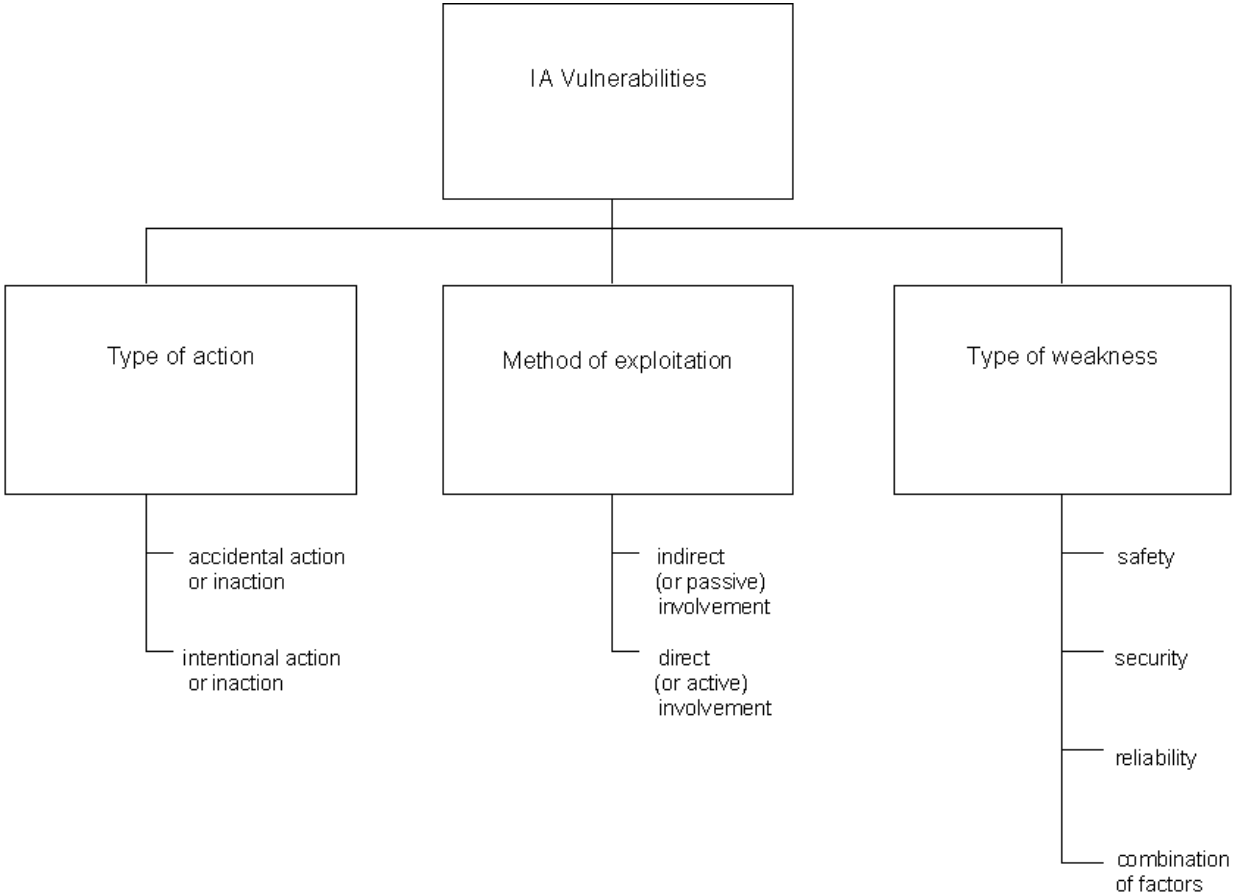

Exhibit 6 Classification of IA Vulnerabilities

As a result, the vulnerability analyses should be reviewed, updated, and revalidated frequently.

An intentional vulnerability is the result of preplanned action — something was done on purpose or deliberately not done. An intentional vulnerability allows features that control system integrity to be bypassed. Intentional vulnerabilities may be created for beneficial purposes, such as facilitating maintenance activities or remote diagnostics and help. Intentional vulnerabilities may be created for malicious purposes, such as allowing certain individuals or organizations to perform unauthorized actions that are illegal and destructive in nature. Trap doors and Trojan horses are two examples of intentional malicious vulnerabilities. A trap door is a hidden software or hardware mechanism that permits system protection mechanisms to be circumvented.[135] The inventor of the trap door is the only one who knows how to activate it. A Trojan horse is a computer program with an apparent useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process to the detriment of security.[135] Both custom and commercial products should be thoroughly analyzed to detect the presence of potential trap doors and Trojan horses. The analysis should also determine whether or not intentional beneficial vulnerabilities can be exploited for malicious purposes. The prevalence of undocumented features in commercial products is a major source of vulnerabilities.

An indirect (or passive) vulnerability is a system weakness that can be exploited independent of direct human action on the part of the perpetrator. Most often, a perpetrator takes advantage of an indirect vulnerability by relying on the actions of other people and processes, that is, waiting for an error to be made. For example, if user names and passwords are transmitted or stored in the clear, a perpetrator does not have to exert any effort to decipher them.

In contrast, a direct (or active) vulnerability is a system weakness that requires direct action by the perpetrator to exploit. Suppose a system is designed without or with inadequate protection from memory conflicts, particularly those that result when multiple COTS components are installed. A perpetrator could initiate transactions that trigger prolonged memory conflicts and thereby crash the system. Exploitation of a vulnerability such as this requires direct action on the part of a perpetrator.

The third way to classify types of vulnerabilities is by the nature of the system weakness. A system might exhibit safety vulnerabilities, security vulnerabilities, reliability vulnerabilities, or some combination thereof. Vulnerabilities may be related; a security vulnerability may give rise to a safety vulnerability, etc. To illustrate, Exhibit 7 classifies nine of the many potential vulnerabilities associated with the online banking system example. In this example, the majority of the vulnerabilities were caused by accidental inaction and can be exploited by indirect involvement on the part of the perpetrator. This is a common situation. Online banking systems do not have safety concerns; consequently, the vulnerabilities are a mixture of security and reliability weaknesses.

A vulnerability may be present in hardware, software, communications equipment, operational procedures for end users, system administrators, and

Exhibit 7 Identification of Vulnerability Types

System: online banking

Vulnerability	*Vulnerability Type*		
	Type of Action	*Method of Exploitation*	*Type of Weakness*
Little or no error detection/ correction or fault tolerance	Accidental inaction	Indirect	Security, reliability
If number of simultaneous users exceeds x, system becomes unstable and exhibits unpredictable behavior	Accidental inaction	Indirect	Security, reliability
Data files can be accessed directly without going through DBMS applications front-end	Accidental inaction	Direct	Security
Suboptimal operational environment, sporadic system shutdown or unpredictable behavior	Accidental inaction and intentional action	Indirect	Security, reliability
End users and system administrator lack sufficient training, understanding of system security features and procedures	Accidental inaction	Indirect	Security
Unsecure backups, archives	Accidental inaction	Indirect	Security, reliability
Careless disposal of hardcopy printouts	Accidental inaction	Indirect	Security
No control over portable equipment or storage media	Accidental inaction	Direct	Security, reliability
COTS components installed with default values, guest accounts, possible trap doors	Accidental inaction and intentional action	Direct	Security, reliability

maintenance staff, and the operational environment. The source of a vulnerability refers to the point in time when the vulnerability was introduced into the system. Some concrete action was taken or left undone to manifest the vulnerability. Exhibit 8 identifies the source of the online banking system example vulnerabilities.

Next, the severity of the consequences of a hazard, resulting from a vulnerability being exploited, is estimated. As previously mentioned, there are four levels of severity: catastrophic, critical, marginal, and insignificant. The primary difference between catastrophic and critical hazards is that critical hazards affect one person/system, while catastrophic hazards affect multiple people/systems. As an example, Exhibit 9 assigns a severity to the potential worst-case hazard consequences for the nine online banking system vulnerabilities. Note that in some cases a range of severities is assigned.

COTS products represent a potential source of vulnerabilities for a variety of reasons, such as the prevalence of undocumented features and functionality,

Exhibit 8 Identification of Vulnerability Sources

System: online banking

Vulnerability	*Source of Vulnerability*
Little or no error detection/correction or fault tolerance	Failure to specify and implement requirements so that system remains in known safe and secure state at all times IA goals not defined
If number of simultaneous users exceeds x, system becomes unstable and exhibits unpredictable behavior	Failure to perform response time, memory, constraint analysis
Data files can be accessed directly without going through DBMS applications front-end	Limited/weak access control and authentication mechanisms
Suboptimal operational environment, sporadic system shutdown or unpredictable behavior	Failure to perform HAZOP studies Vendor recommendations for operational environment ignored or incorrect
End users and system administrator lack sufficient training, limited understanding of system security features and procedures	Failure to develop operational profiles and scenarios Inadequate operational procedures Poor planning and training prior to system deployment
Unsecure backups, archives	Inadequate operational procedures Physical security issues not considered
Careless disposal of hardcopy printouts	Inadequate operational procedures Physical security issues not considered
No control over portable equipment or storage media	Inadequate operational procedures Physical and operational security issues not considered
COTS components installed with default values, guest accounts, possible trap doors	Inadequate analysis of COTS vulnerabilities prior to installation Failure to confine COTS products

maintenance or diagnostic trapdoors, default settings, conflicts with other COTS products, etc. As Lindquist and Jonsson[335] state:

> *Any type of COTS component might have an impact on the overall system security, depending on how it is used in a system. Therefore, every type of COTS product could be security-related.*

Zhong and Edwards[446] cite specific vulnerabilities associated with unexpected/undocumented COTS behavior:

- The component may access unauthorized resources or services.
- The component may access a resource in an unauthorized way.
- The component may abuse authorized privileges.

Concerns about vulnerabilities inherent in COTS products are not limited to software; hardware is included as well. Given the emphasis on using COTS

Exhibit 9 Identification of Vulnerability Severity

System: online banking

Vulnerability	*Hazard Consequences*	*Severity*
Little or no error detection/ correction or fault tolerance	Transactions posted to wrong accounts; interest payable, interest due calculated incorrectly; automatic deposits and payments lost; etc.	Critical - catastrophic
If number of simultaneous users exceeds x, system becomes unstable and exhibits unpredictable behavior	Screens are displayed very slowly; wrong screens are displayed; screens are displayed in wrong sequence; customer A sees customer B's transaction; etc.	Marginal - critical
Data files can be accessed directly without going through DBMS applications front-end	Critical/sensitive data can be maliciously altered, deleted, copied, and/or stolen with ease.	Critical - catastrophic
Suboptimal operational environment, sporadic system shutdown or unpredictable behavior	Customers cannot access the system; system crashes in the middle of a transaction; partial posting of transactions.	Marginal
End users and system administrator lack sufficient training, limited understanding of system security features and procedures	System security features are routinely disabled and/or bypassed.	Critical - catastrophic
Unsecure backups, archives	Critical/sensitive data can be maliciously altered, deleted, copied, and/or stolen with ease.	Critical - catastrophic
Careless disposal of hardcopy printouts	Critical/sensitive data can be stolen, copied, and/or distributed.	Critical - catastrophic
No control over portable equipment or storage media	Critical/sensitive data and applications can be stolen, copied, altered, or given to a third party.	Critical
COTS components installed with default values, guest accounts, possible trap doors	COTS components perform incorrectly, however, error is not overtly obvious; system security authorizations can be bypassed for malicious purposes.	Critical

products to (supposedly) deploy systems faster and cheaper, the concern about COTS products is likely to increase in the future. The same holds true with regard to software reuse. Exhibit 10 summarizes potential hardware and software COTS vulnerabilities that may manifest themselves at any time during the life of a system.

Vulnerability analyses evaluate internal and external entities. Like most systems, Internet-based applications rely on external entities to accomplish their mission. Each external entity is a potential source of additional vulnerabilities. To illustrate, Bradley et al.[223] have identified several potential vulnerabilities associated with routers, including:

Exhibit 10 Potential COTS Vulnerabilities

1. Component design:
   - Inadvertently flawed component design
   - Intentionally flawed component design
   - Excessive component functionality
   - Open or widely spread component design
   - Insufficient or incorrect documentation
2. Component procurement:
   - Insufficient component validation
   - Delivery through insecure channel
3. Component integration:
   - Mismatch between product security levels
   - Insufficient understanding of integration requirements
4. System Internet connection:
   - Increased external exposure
   - Intrusion information and tools easily available
   - Executable content
   - Outward channel for stolen information
5. System use:
   - Unintended use
   - Insufficient understanding of functionality
6. System maintenance:
   - Insecure updating
   - Unexpected side effects
   - Maintenance of trap doors

Source: Summarized from U. Lindquist and E. Jonsson, *Computer*, 31(6), 60–66, 1998. With permission.

- Dropping packets
- Misrouting packets
- Intelligent network sinks cooperating to conceal evidence of dropping or misrouting packets
- Altering contents of a packet, message, or destination address, copying to other addresses
- Sending false topology/routing table updates to bypass good routers and target malicious routers
- Injecting bogus packets
- Inspecting packet contents

Note that this list is a combination of accidental and intentional malicious action.

It is essential that vulnerability analyses be performed because all systems have vulnerabilities. As Neumann[362] succinctly states, "There is never an absolute sense in which a system is secure or reliable." Identifying vulnerabilities is the first step along the road to determining threats and risk exposure. Without this first step, effective threat control measures cannot be implemented.

Vulnerabilities can exist in a system's hardware, software, communications equipment, operational procedures, and operational environment. Vulnerabilities

can be related to system safety, reliability, and security. Vulnerabilities can result from accidental or malicious intentional action or inaction. Historically, the safety and reliability communities focused on accidental vulnerabilities, while the security community focused on malicious intentional vulnerabilities. Information security/IA brings these different perspectives together. As Jackson[304] notes:

> *Hazard analysis of potentially safety-critical systems has evolved on the assumptions that hardware suffers from wear and tear in normal use, that [software] logic contains accidental design errors made in good faith, and that all systems are subject to environmental occurrences. The notion of malicious interference or sabotage is not generally considered, but it puts a very different complexion on hazard analysis. … Malicious interference is normally the parallel province of security, and in some sectors, such as the nuclear sector, safety and security are inseparable.*

It is important to be thorough when performing vulnerability analyses; the effectiveness of the threat control measures depends on it. Certain kinds of vulnerabilities are often overlooked. For example, vulnerabilities related to inadvertently revealing data through ignorance, naivete, over-confidence, negligence, or other operational errors are often ignored.[357] This highlights the need to consider human factors and operational procedures when performing vulnerability analyses, not just hardware and software. Another category that is often overlooked is lateral hazards that result from vulnerabilities. Lateral hazards are unique to each system and are difficult to analyze. They generally result from an unusual unplanned combination of events, the composite of which has potential hazardous consequences. The following situation illustrates this concept. During the winter, a water pipe freezes in a high-rise condominium. The pipe bursts and floods several floors. Water seeps into the elevator shaft, causing it to malfunction. Water also leaks into the building's automatic fire alarm system, causing it to cease functioning. In this case, the vulnerability was a water pipe freezing and the hazard was water damage from a flood. However, the hazard (flood) caused other lateral hazards to occur: the elevator and automatic fire alarm malfunctions.

Vulnerabilities need to be not only identified, but also characterized according to type, source, and severity so that: (1) appropriate threat control measures can be developed, and (2) resources can be effectively applied to the most critical vulnerabilities. This is common sense because the weakest link in a system will be attacked. Exhibit 11 provides a system vulnerability characterization for the online banking example. A common misperception in the information security world is that most vulnerabilities are caused by intentional action and require direct exploitation. In contrast, in this example, 82 percent of the vulnerabilities are caused accidentally and can be exploited indirectly. Equally alarming is the fact that 56 percent of the vulnerabilities are catastrophic.

Exhibit 11 Vulnerability Characterization Summary: Online Banking System

as of date:_____

I. Vulnerability Type Summary

	Cause				Exploitation			
	Accidental		Intentional		Indirect		Direct	
Vulnerabilities	#	%	#	%	#	%	#	%
Safety	—		—		—		—	
Reliability	—		—		—		—	
Security	3	27%	—		2	18%	1	9%
Combination	6	55%	2	18%	6	55%	2	18%
Total	9	82%	2	18%	8	73%	3	27%

II. Vulnerability Source Summary

	Lack of Control				Operational				System Error			
	Internal Entity		External Entity		Environment		Procedures		Specification		Design, Integration	
Vulnerabilities	#	%	#	%	#	%	#	%	#	%	#	%
Safety	—		—		—		—		—		—	
Reliability	—		—		—		—		—		—	
Security	—		—		—		2	18%	—		1	9%
Combination	—		—		1	9%	2	18%	2	18%	3	27%
Total	—		—		1	9%	4	36.5%	2	18%	4	36.5%

III. Vulnerability Severity Summary

	Catastrophic		Critical		Marginal		Insignificant	
Vulnerabilities	#	%	#	%	#	%	#	%
Safety	—		—		—		—	
Reliability	—		—		—		—	
Security	3	33%	1	11%	—		—	
Combination	2	22%	2	22%	1	11%	—	
Total	5	56%	3	33%	1	11%	—	

Note: A vulnerability may have multiple causes and multiple exploitation methods; there is not a one-to-one correspondence. Similarly, a vulnerability may have multiple sources. Likewise, vulnerability severity can be expressed as a range (see Exhibit 9). In this case, the worst-case scenario is used to characterize the vulnerabilities.

5.4 Identify Threats, Their Type, Source, and Likelihood

Information security/IA threats are characterized in three ways, as shown in Exhibit 12:

- Type of action that can instantiate the threat
- Source of the action that can trigger the threat
- Likelihood of the threat occurring

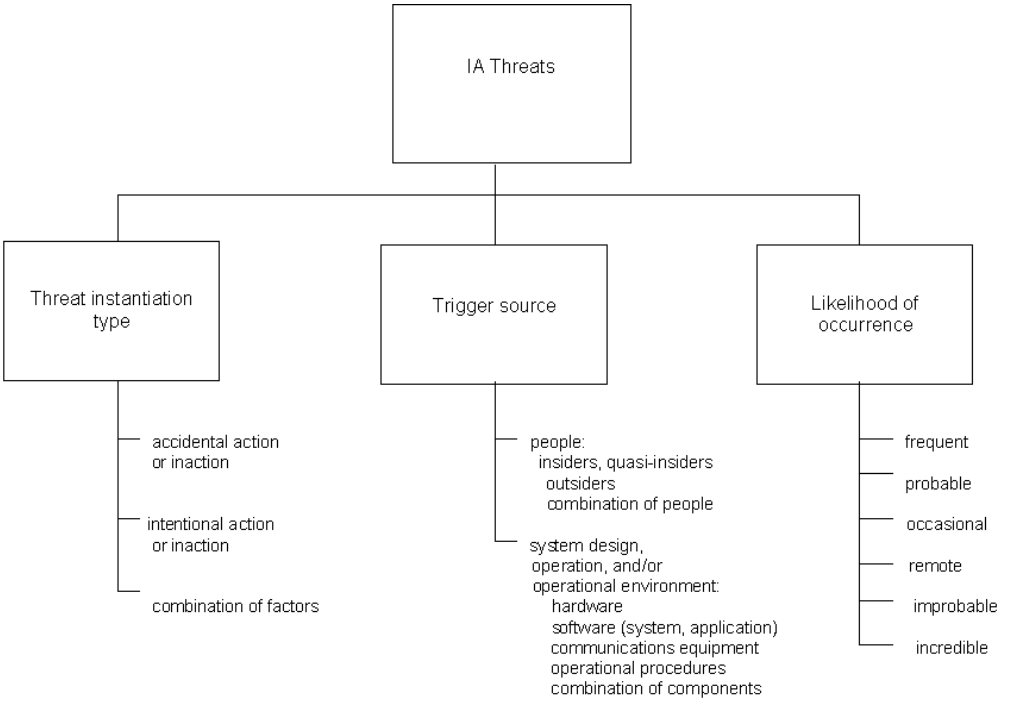

Exhibit 12 Characterization of IA Threats

It is important to maintain the distinction between vulnerabilities (weaknesses in system design, operation, or operational environment that can be exploited) and threats (the potential that a weakness will be exploited). Otherwise, the results from the vulnerability and threat analyses will be confusing and misleading.

A threat can be instantiated by accidental or intentional action or inaction, or a combination of factors. This is different than the accidental action or inaction and intentional action or inaction that are the cause of vulnerabilities. In that case, the accidental action or inaction, or intentional action or inaction, caused the vulnerability to be manifest. In this case, accidental action or inaction, or intentional action or inaction, will cause a threat to be instantiated and a vulnerability to be exploited. In addition, a combination of accidental and/or intentional action and/or inaction may trigger a threat. This is why it is important to analyze combinations of events and not just single events.

A threat can be triggered by people or system entities. People who can trigger a threat include insiders (people employed by an organization), quasi-insiders (people under contract to provide certain services to an organization), visitors (customers, vendors, meeting attendees, etc.), and outsiders (people who have no formal relationship to an organization). Most sources[248,277,362,399] consider insiders and quasi-insiders to be as likely a threat source as outsiders. Then again, insiders may collude with outsiders to trigger a threat. As Jajodia[305] points out, it may be difficult to determine if a threat is in fact originating from an insider or an outsider due to masquerading. Outsiders can include individuals, groups, or state-sponsored organizations whose goal is to inflict

physical or cyber damage. Likewise, the inherent design, operation, and operational environment of a system can trigger a threat. Factors related to the operational environment can trigger a threat and expose an unanticipated system vulnerability, such as unexpected resource contention, insufficient temperature, humidity, vibration, and dust controls, EMC, EMI, RFI, and power faults.

Once the instantiation type and trigger source are known, the likelihood of a threat occurring is of prime importance. Attempts have been made to produce precise quantitative threat likelihood estimates[375]; however, they are expensive to produce and difficult to defend. For the sake of practicality and reasonableness, qualitative estimates are sufficient. The intent is to predict whether or not a threat is likely to occur; and if so, how likely. For example, it is fair to assume that the likelihood of virus, denial-of-service, IP spoofing, password stealing, and other generic attacks is probable if not frequent. By the same token, entities shown through the entity control analysis not to be under direct control of an organization can reasonably be assumed to be as, if not more, likely to trigger a threat than an entity under an organization's control. As mentioned earlier, most international standards[57,65,129,130,143] use six qualitative likelihood categories. Once assessed, threat likelihood and vulnerability severity are correlated to prioritize resources for threat control measures. As Rathmell[391] makes clear:

> *InfoSec resources can best be applied **only** if guided by a structured threat assessment process.*

Exhibit 13 identifies the threats for the nine vulnerabilities associated with the online banking system example. This process links the vulnerabilities and threats. As Rathmell[391] notes:

> *An overall risk assessment must overlay identified or potential threats onto the vulnerabilities of the defenders' information activities in order to determine the degree of risk and so plan responses.*

The threat instantiation type is determined for each vulnerability in the first column. It can be accidental action, accidental inaction, intentional action, intentional inaction, or a combination of factors. For example, the threat instantiation type for Vulnerabilities 2 and 5 is a combination of factors. The threat trigger source for each vulnerability is fixed in the second column. The trigger source can be various groups of people, other system entities, or a combination of people and system entities, as cited for Vulnerability 2. The likelihood of the threat occurring is estimated in column three. Following standard risk management practices, likelihood estimates are developed for worst-case scenarios. Denning[248] and Rathmell[391] observe that threat likelihood is a dynamic index because it results from a combination of capability, motive, and resources available. Hence, likelihood estimates should be reviewed, updated, and revalidated frequently.

Exhibit 14 presents the system threat characterization for the online banking example. The majority of the threats are intentionally instantiated (56 percent)

Exhibit 13 Threat Identification: Online Banking System

as of date:_____

Vulnerability	*Instantiation Type*	*Trigger Source*	*Likelihood of Occurrence*
1. Little or no error detection/ correction or fault tolerance	Accidental inaction	System design: system will corrupt itself	Frequent
2. If number of simultaneous users exceeds x, system becomes unstable and exhibits unpredictable behavior	Accidental or intentional action	System design: system will corrupt itself (accidental) People: insiders or outsiders who become aware of this design flaw may purposely exploit it (intentional)	Probable Occasional
3. Data files can be accessed directly without going through DBMS applications front-end	Intentional action	People: insiders or outsiders who become aware of this design flaw may purposely exploit it	Occasional
4. Suboptimal operational environment, sporadic system shutdown or unpredictable behavior	Accidental inaction	System design: system will corrupt itself	Occasional
5. End users and system administrator lack sufficient training, limited understanding of system security features and procedures	Accidental or intentional action	People: insiders, by not understanding or following security procedures, create opportunities for outsiders to trigger more serious threats	Probable
6. Unsecure backups, archives	Intentional action	People: insiders or outsiders who become aware of this operational weakness may purposely exploit it	Occasional
7. Careless disposal of hardcopy printouts	Intentional action	People: insiders or outsiders who become aware of this operational weakness may purposely exploit it	Occasional
8. No control over portable equipment or storage media	Intentional action	People: insiders, or insiders colluding with outsiders, could purposely exploit this operational weakness	Occasional
9. COTS components installed with default values, guest accounts, possible trap doors	Intentional action	People: system components create the vulnerability, but it takes deliberate action on the part of insiders or outsiders to exploit it	Occasional

Exhibit 14 Threat Characterization Summary: Online Banking System

as of date:_____

I. Threat Instantiation Type Summary

	Accidental		*Intentional*		*Combination*	
Threats	#	%	#	%	#	%
Safety	—		—		—	
Reliability	—		—		—	
Security	—		2	22%	1	11%
Combination	2	22%	3	33%	1	11%
Total	2	22%	5	56%	2	22%

I. Threat Trigger Source Summary

	People		*Systems*		*Combination*	
Threats	#	%	#	%	#	%
Safety	—		—		—	
Reliability	—		—		—	
Security	3	33%	—		—	
Combination	3	33%	2	22%	1	11%
Total	6	67%	2	22%	1	11%

III. Threat Likelihood Summary

	Frequent		*Probable*		*Occasional*		*Remote*		*Improbable*		*Incredible*	
Threats	#	%	#	%	#	%	#	%	#	%	#	%
Safety	—		—		—		—		—		—	
Reliability	—		—		—		—		—		—	
Security	—		1	11%	2	22%	—		—		—	
Combination	1	11%	1	11%	4	45%	—		—		—	
Total	1	11%	2	22%	6	67%	—		—		—	

and triggered by people (67 percent). However, that does not mean that the other ~30 percent of the threats should be ignored, that is, those that are instantiated accidentally or triggered by inherent flaws in a system design, operation, or operational environment.

Exhibit 15 depicts the initial correlation of vulnerability severity and threat likelihood for the online banking example. Using this information, priorities can be established for threat control measures. One possible grouping would be:

- High priority: vulnerabilities 1, 3, 5, 6, 7
- Medium priority: vulnerabilities 2, 8, 9
- Low priority: vulnerability 4

Priorities are decided on a case by case basis, taking into account a variety of parameters such as: laws and regulations, liability and other legal concerns,

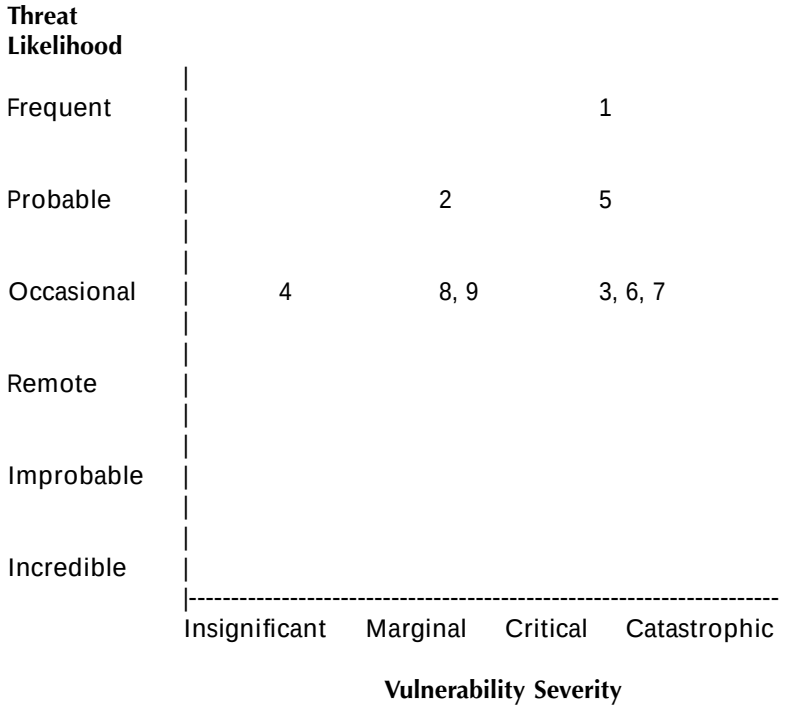

Exhibit 15 Correlation of Threat Likelihood and Vulnerability Severity to Prioritize Threat Control Measures. *Note:* numbers shown on the graph represent the vulnerability number.

organizational goals and ethics, and of course cost and schedule. A decision could be made not to do anything about vulnerabilities that are of marginal severity or lower in one instance. In another situation, a decision may be made that all (known) vulnerabilities must be eliminated or mitigated to the extent that they become insignificant.

5.5 Evaluate Transaction Paths, Threat Zones, and Risk Exposure

For the hypothetical high-level online banking example: (1) vulnerabilities, their type, source, and severity have been identified; (2) threats, their type, source, and likelihood have been determined; and (3) the vulnerability severity and threat likelihood have been correlated to prioritize threat control measures. The next step is to implement the threat control measures, right? For a very simplistic system perhaps. However, in the real world, systems are complex and as a result there are more parameters to consider. Hence, the threat and vulnerability analyses need to be refined; otherwise, a lot of the time and resources spent on threat control measures could be wasted. Consequently, the next step is to ascertain all logically possible combinations of discrete activities that could cause a system to be compromised; in other words,

potential transaction paths are developed. Then the threat zones are evaluated and initial risk exposure determined. The development of transaction paths permits the information security/IA challenge to be attacked from both ends. Transaction paths identify how a system could be (or was) compromised. Vulnerability and threat characterizations identify system weaknesses and the potential for exploitation. The two analyses reinforce and refine each other. This extra level of analysis helps to:

- Uncover new vulnerabilities and methods of exploitation
- Refine threat source definitions and likelihood estimates
- Examine different threat perspectives
- Evaluate how different operational modes and states and the time element contribute to risk exposure
- Optimize the application of threat control resources by identifying common lower-level events within transaction paths

Transaction paths capture the sequence of discrete events that could cause an event to take place — in this case, a system to be attacked/compromised. Transaction paths depict all logically possible ways in which an event might occur. Transaction paths are concerned with what is logically possible — how something could be accomplished, not whether it is feasible, economical, probable, etc. That aspect of the analysis comes later. As Sherlock Holmes repeatedly reminds Dr. Watson, "When the possible has been eliminated, consider the impossible." The rationale is that what is often considered impossible is not really impossible, but rather improbable, like an unusual combination of events. Rabbi Levi Shem Tov expressed the same idea in a different manner, "Persistence is what makes the impossible possible, the possible probable, and the probable definite."

All possible paths are shown in one diagram. Each discrete event is numbered hierarchically. An individual path represents a unique route from the top event to a bottom event. Logic symbols define the relationship between alternative events. Paths are developed to the level to which it is meaningful to carry the analysis. As explained in Chapter 8, transaction paths can also be developed *a posteriori* to reconstruct how an accident/incident occurred.

To illustrate, potential transaction paths that could lead to the compromise of a hypothetical air traffic control (ATC) system will be developed. First, the boundaries of the system are defined. As shown in Exhibit 16, the three main system entities are the aircraft/pilot, radar, and ATC system/controller. At a high level, the logical operation of the ATC system can be summarized as follows. All aircraft in airspace x continuously send a location signal that is monitored by radar x. The radar links the location information to specific aircraft and forwards it to the ATC system. The ATC system maintains real-time information about the location of airborne aircraft, aircraft on the ground, status of runways, taxiways, and gates, and projected flight plans and schedules. At the same time, voice communication takes place between the pilot and the assigned air traffic controller. Depending on the size of an airport and the volume of traffic, the responsibility for flights may be distributed

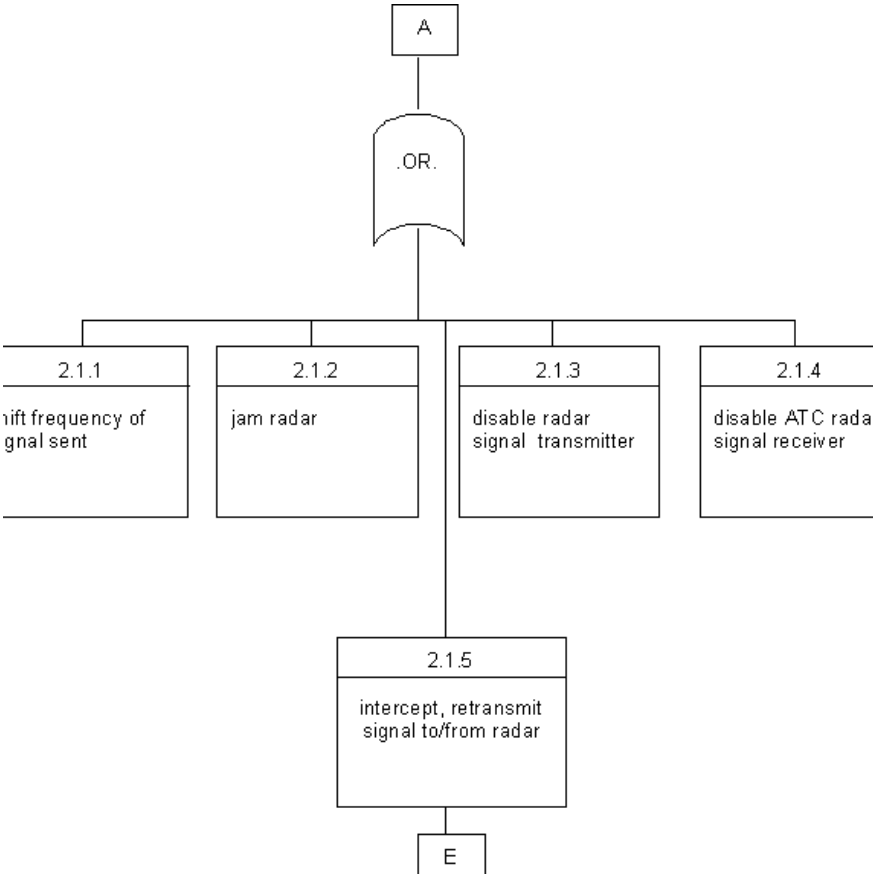

Exhibit 16 High-Level Depiction of the Logical Operation of an ATC System

among several air traffic controllers or assigned to a single person. The operation of this system depends on several assumptions, including:

- The pilot assumes that he or she is communicating with the real air traffic controller.
- The air traffic controller assumes that he or she is communicating with the real pilot.
- The radar assumes that the signal is coming from the identified aircraft.
- The ATC system assumes the signal received is from radar x.

Exhibits 17 through 25 illustrate the potential transaction paths that could lead to a compromise of the hypothetical ATC system example. At the first level, there are four possible ways in which the hypothetical ATC system could be compromised (Exhibit 17):

- Tampering with communication from the radar to the ATC system
- Tampering with voice communications between the ATC controller and pilot

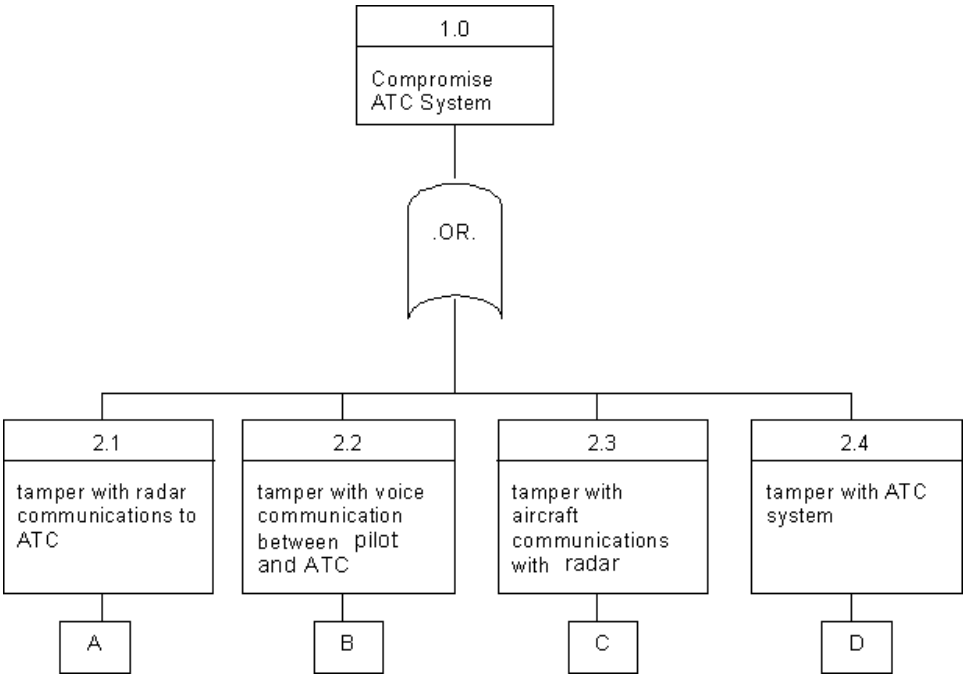

Exhibit 17 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System

- Tampering with communication from the aircraft to the radar
- Tampering with the ATC system itself

Continuing with the path, 1.0 ← A, one sees that there are five possible ways to tamper with communication from the radar to the ATC system (Exhibit 18):

- The frequency of the signal sent from the radar could be shifted, so that the ATC does not receive it — it expects a different frequency signal.
- The radar signal could be jammed so that the ATC system cannot receive it.
- The radar signal transmitter could be disabled. (This could be done quite subtly so that it appears that the radar is transmitting normally when in fact it is not.)
- The ATC radar signal receiver could be disabled.
- The signal from the radar could be intercepted and retransmitted.

Note that similar events could lead to the compromise of the ATC system if paths 1.0 ← B or 1.0 ← C were followed (see Exhibits 19 and 20).

Continue down the path 1.0 ← 2.1.5, assuming that the radar signal is intercepted and retransmitted. There are seven ways this could be accomplished (Exhibit 22):

- The signal or message could be erroneously repeated.
- All or part of the signal or message could be deleted.

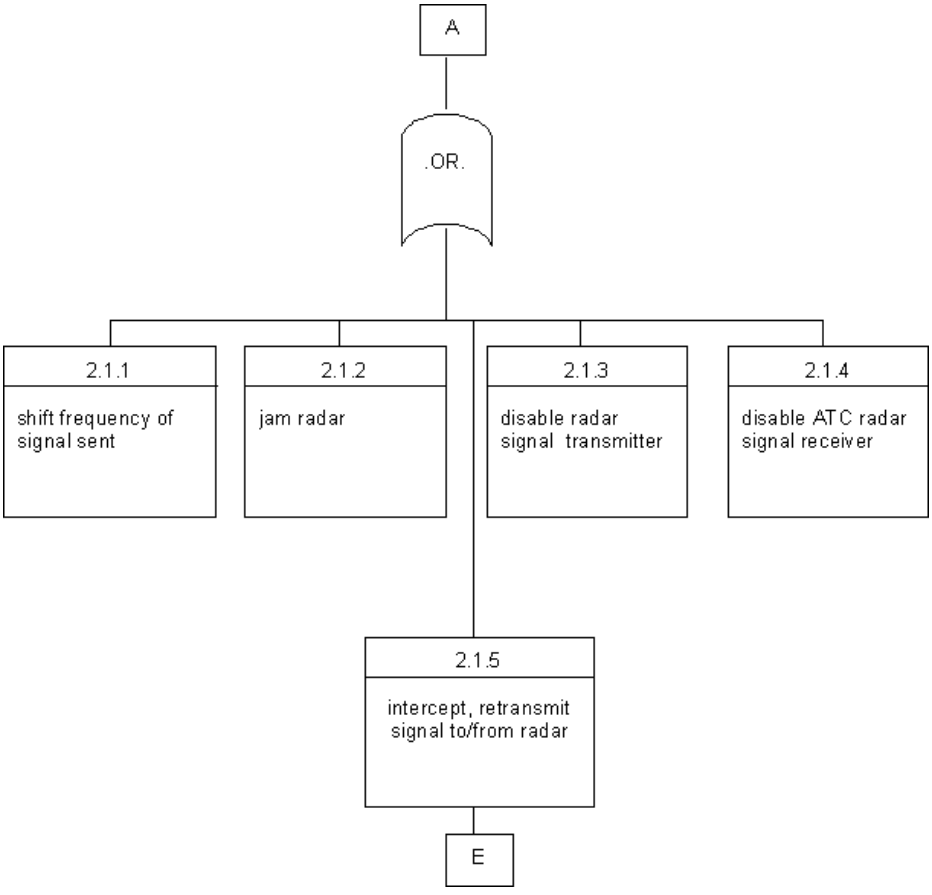

Exhibit 18 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

- A bogus signal or message could be transmitted.
- Messages or signals could be sent in the wrong order.
- Transmission of the message or signal could be delayed such that the information is no longer real-time.
- The contents or origin of a message could be falsified.
- The message or signal could be made unintelligible.

If instead, the ATC system itself were the target of an attack, there are three ways in which it could be compromised (Exhibit 21):

- The controller terminals could be corrupted.
- The ATC DBMS* could be corrupted.
- Communication between the controller terminals and the ATC DBMS could be corrupted.

* Note that, for the most part, ATC systems only process real-time data; hence, a classical DBMS is not used. The term "DBMS" is used in this example only to refer to a logical grouping of data.

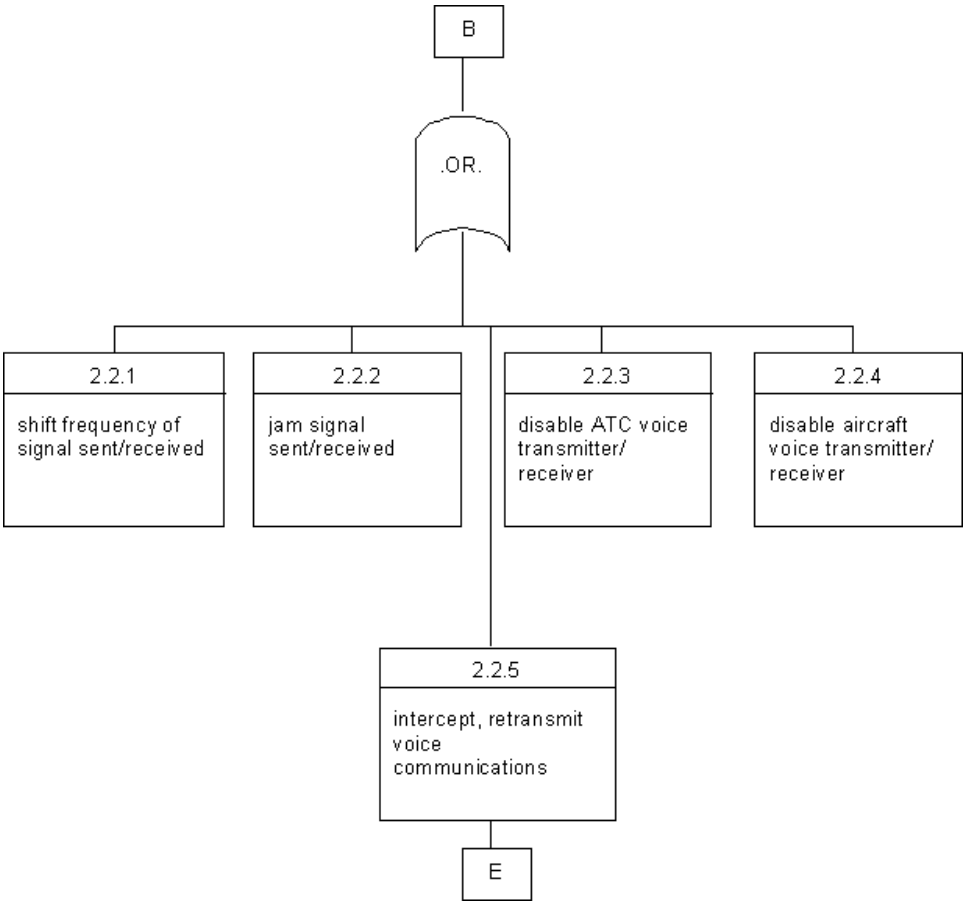

Exhibit 19 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

The controller terminal could be corrupted by (Exhibit 23):

- Causing the screen to freeze temporarily or permanently
- Displaying duplicate data
- Deleting some data points
- Causing the screen to go blank temporarily or permanently
- Inserting bogus data points
- Delaying the screen refresh showing new data points
- Displaying information for terminal x_1 on terminal x_n, in the case of multiple air traffic controllers
- Crashing the air traffic control terminal

The ATC DBMS could be corrupted by (Exhibit 24):

- Adding bogus data to the DBMS
- Deleting legitimate data from the DBMS
- Modifying legitimate data from the DBMS

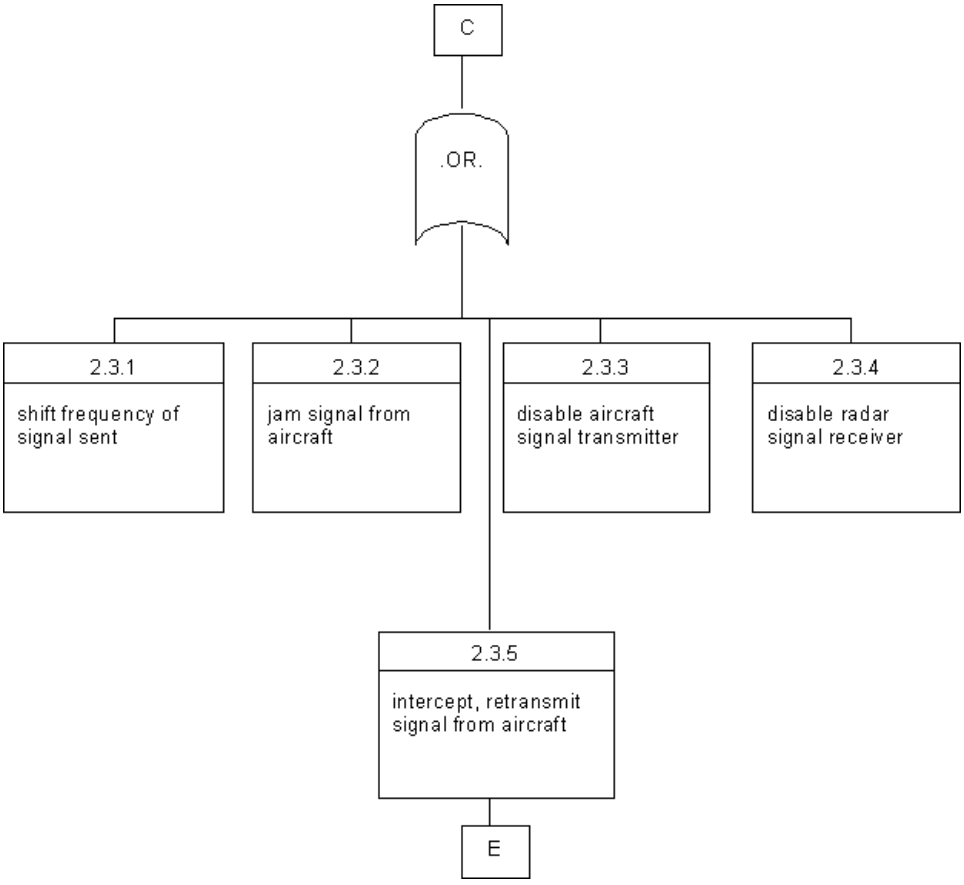

Exhibit 20 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

- Duplicating data in the DBMS
- Restoring old data so that it overwrites current information
- Scrambling pointers or indices used to access data
- Making the data unintelligible

Communication between the ATC DBMS and terminals could be corrupted by (Exhibit 25):

- Sending wrong information to the ATC terminals
- Sending information to the ATC terminals too early, too late, or in the wrong sequence
- Not sending or withholding information
- Sending information to the wrong terminal(s)
- Crashing the communications links between the DBMS and terminals

There are 61 unique paths in this example; that is, 61 different ways in which the hypothetical ATC system could be compromised. And this is not

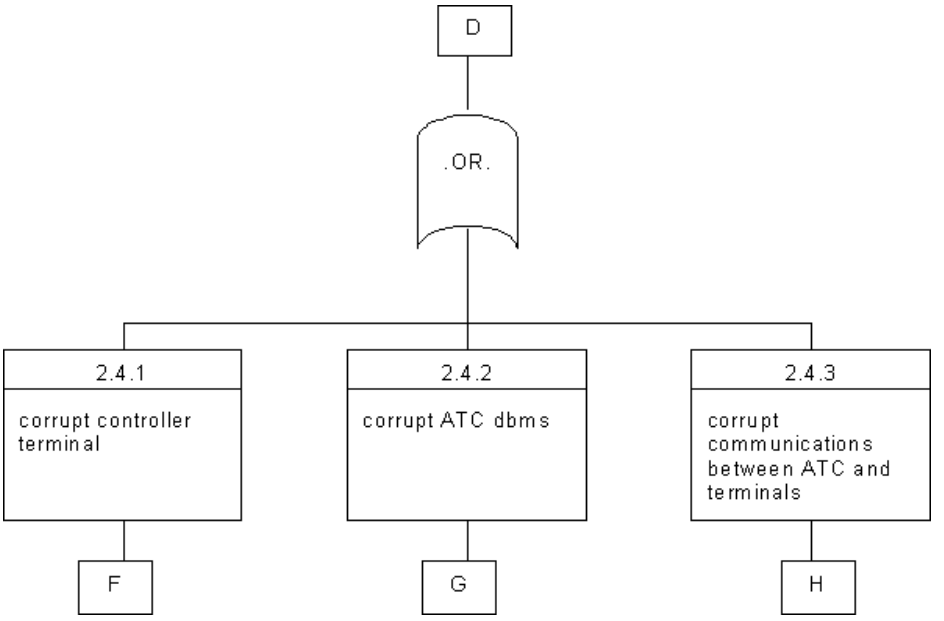

Exhibit 21 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

an exhaustive example. These 61 are single paths that could lead to a compromise. Suppose a combination of events occurred, such as tampering with the radar communications with the ATC system and tampering with the voice communications between the pilot and air traffic controller. The number of potential compromise paths increases and, most likely, the subtlety of the compromise increases as well.

As shown in this example, a transaction path and compromise may be overt or subtle. The fact that a signal is jammed is quite noticeable, while signal interception, modification, and retransmission may not be. In some cases, the compromise may be so subtle that it goes unnoticed until it is too late.[248] The transaction path chosen will vary, depending on opportunity and intent.[248,391] In this case, the intent could be to disrupt a single flight, an entire airport, or an entire geographical area.

Transaction paths can also be depicted graphically through the use of event trees. Bott of Los Alamos National Laboratories has developed a model that combines event trees and quantitative probabilistic risk assessment (PRA) to analyze potential system compromise paths from insiders and visitors. This approach consists of five steps[220]:

1. Potential compromise paths (discrete event sequences) are identified.
2. Potential compromise paths are grouped.
3. A probability model is developed for each group using historical experience and expert judgment.
4. The probability of occurrence is calculated for each potential compromise path group.

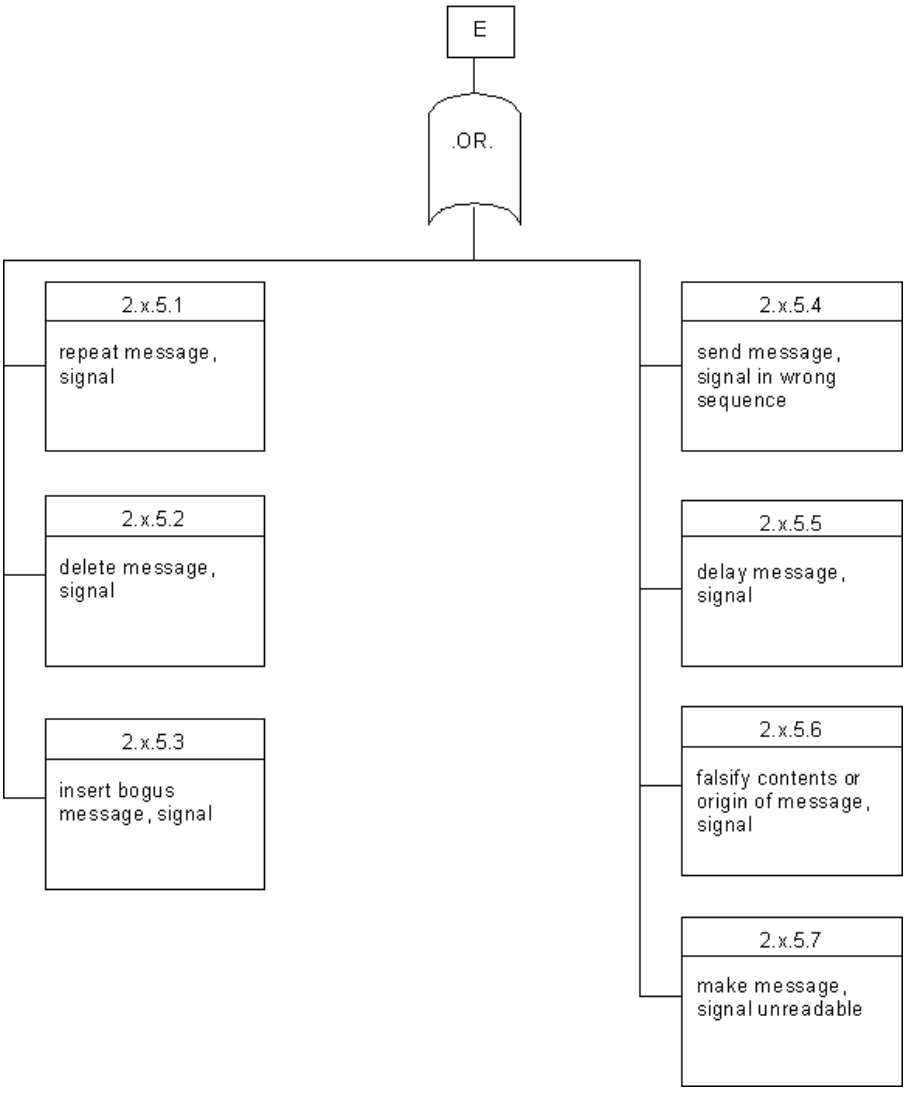

Exhibit 22 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

5. Compromise path groups are rank ordered according to security resource allocation priority.

Vulnerability and threat analyses are generally performed from the perspective of the system owner. This is a carryover from the days when defense and national security agencies were the only ones concerned about information security. To be complete, vulnerability and threat analyses should be conducted from multiple perspectives, the perspectives of all groups of people who interact with or are affected by a system. Transaction paths help to uncover these different perspectives. Once the different groups of people have been identified, potential system compromise paths are evaluated from each group's vantage

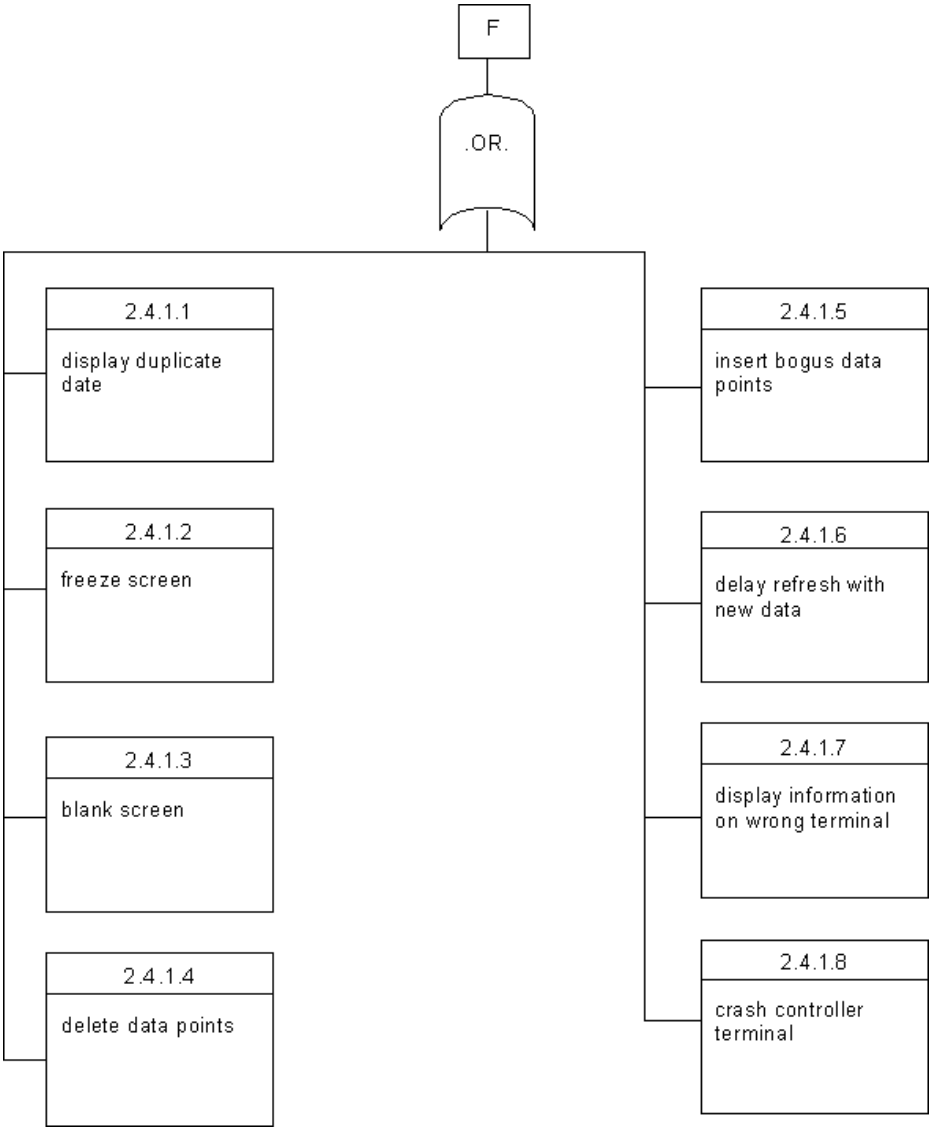

Exhibit 23 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

point. This analysis helps to uncover new vulnerabilities and new methods of exploitation, as well as refine severity estimates. To illustrate, Exhibit 26 examines the four potential high-level compromises for the hypothetical ATC system from six different threat perspectives.

Threat zones represent a segment of the transaction path that is associated with a specific operational mode/state, operational profile, and time element. Information from the system entity control analysis is factored in when isolating a threat zone. The intent is to zero in on the weakest links in the chain — the events most likely to lead to a system compromise. For example, not all

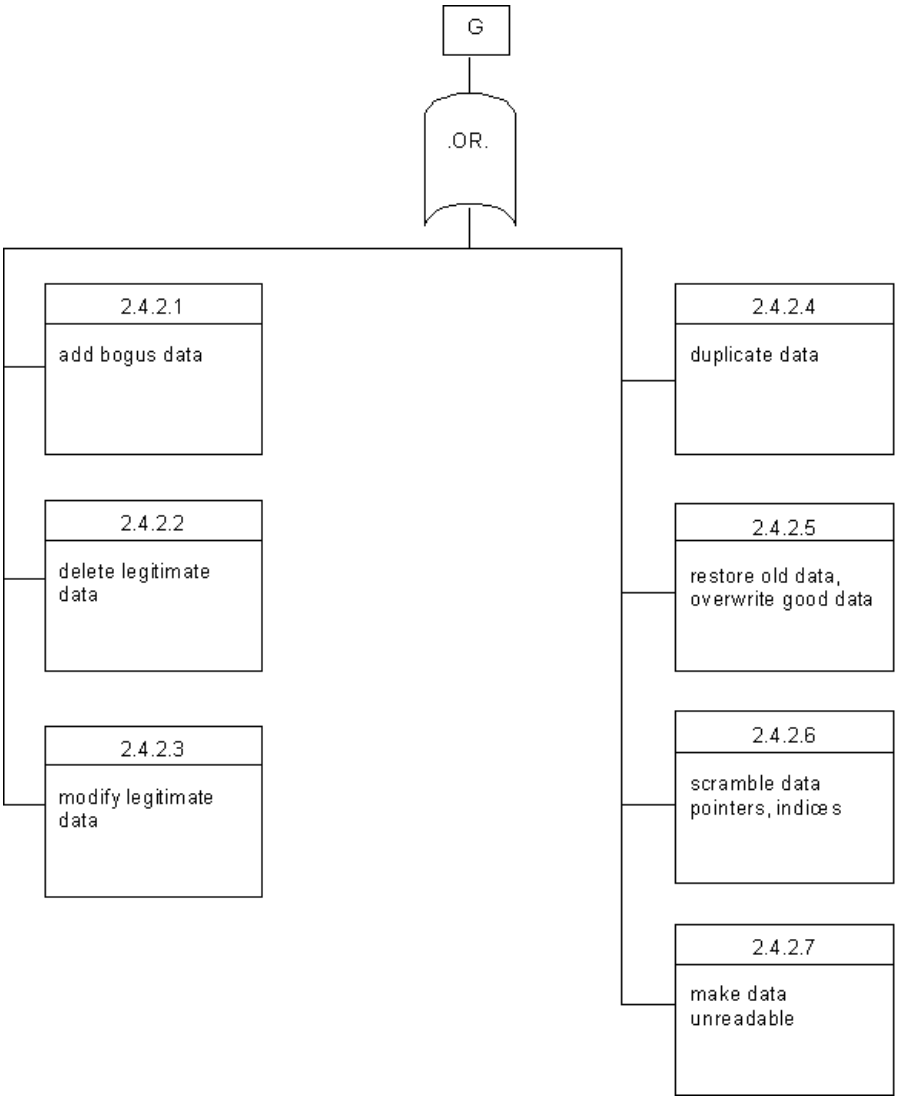

Exhibit 24 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

61 potential transaction paths that could lead to the compromise of the ATC system example are equally: (1) as likely to occur, or (2) as easy to accomplish, accidentally or intentionally. Rather, for each system, specific combinations of a transaction path segment, operational mode/state, operational profile, time of day/year, etc. are more likely to lead to a compromise. Opportunity, motive, and intent for an attack must also be considered when isolating critical threat zones.[248,391] Once the critical threat zones have been identified and ranked, attack points can be fortified *a priori* and loss prevented.

Overall system risk exposure is derived from several interactive factors, as shown in Exhibit 27:

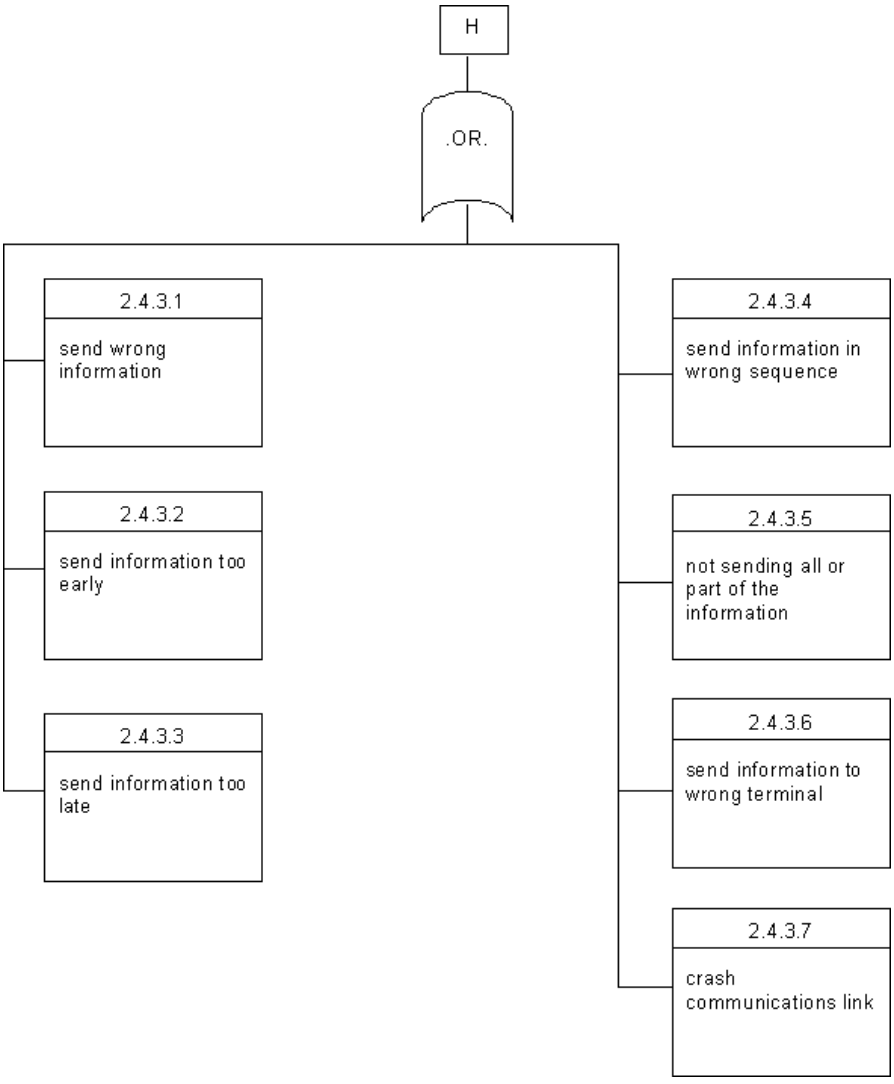

Exhibit 25 Potential Transaction Paths Leading to the Compromise of a Hypothetical ATC System (continued)

- Correlating vulnerability severity and threat likelihood
- Analyzing transaction paths from different threat perspectives
- Isolating critical threat zones

It is essential to remember in all cases — whether evaluating vulnerability severity, threat likelihood, transaction paths, or threat zones — that loss can occur as a result of accidental ***or*** intentional action ***or*** inaction. At the same time, all entities and factors which effect the design, operation, and operational environment of a system should be analyzed. Concerns traditionally associated with OPSEC and physical security should be scrutinized as thoroughly as those associated with INFOSEC. Safety, reliability, and security concerns should be assessed in tandem. In short, the effectiveness of the implementation of threat

Exhibit 26 System Compromises Examined from Different Threat Perspectives

Threat Perspective	*Tamper with Communication between Aircraft and Radar*	*Tamper with Pilot/Air Traffic Controller Voice Communication*	*Tamper with Radar Communication with ATC*	*Tamper with ATC System*
Pilot/crew	Pilots rely on an accurate location signal being sent to the radar. If radar signal and onboard instrumentation disagree, how will pilots know which is correct? In the worst case, pilots may not know signal has been altered.	Pilots assume they are talking with real air traffic controllers. They have no way of knowing otherwise. If information is not received/relayed correctly, severe consequences could result.	Pilots rely on an accurate signal being sent from the radar to the ATC system so that controllers can provide correct landing information.	Pilots are directed by air traffic controllers based on information from the ATC system. If that information has been altered or is incorrect, it could affect the safety of one or more aircraft.
Passengers	Passengers rely on an accurate signal being sent to the radar so that their plane can land safely. They have no way of knowing otherwise.	Passengers rely on the accurate exchange of information between pilot and air traffic controller so that their plane can land safely. They have no way of knowing otherwise.	Passengers rely on accurate information being sent from the radar to the ATC system so that their plane can land safely. They have no way of knowing otherwise.	Passengers rely on the ATC system to provide accurate information to the controller so that their plane can land safely. They have know way of knowing otherwise.
Air traffic controllers	Air traffic controllers assume that information displayed on their screen is correct. They have no way of knowing that is incorrect unless voice communications so indicate. Critical decisions are based on this information, which may affect more than one aircraft.	Air traffic controllers assume that they are talking with real pilots. They have no way of knowing otherwise. If information is not received/relayed correctly, severe consequences could result.	Controller relies on an accurate signal being sent from the radar to the ATC system so that they can provide correct landing information to the pilot.	Air traffic controllers direct pilots and ground crews based on information from the ATC system. Conflicting voice communication is the only way they would know if the ATC information is in error. Erroneous information could lead to severe consequences.

Exhibit 26 System Compromises Examined from Different Threat Perspectives (continued)

Threat Perspective	*Tamper with Communication between Aircraft and Radar*	*Tamper with Pilot/Air Traffic Controller Voice Communication*	*Tamper with Radar Communication with ATC*	*Tamper with ATC System*
ATC system maintenance technicians	—	—	Maintenance technicians only verify that a signal is received. They have no way of knowing if the contents have been altered.	Maintenance technicians only verify that the system functions correctly. They have no way to knowing if the information content is correct.
Radar system maintenance technicians	Maintenance technicians only verify that a signal is received at the correct frequency, timing, etc. They have no way of knowing if the signal content is correct.	—	Maintenance technicians only verify that a signal is sent at the correct frequency, timing, etc. They have no way of knowing if the signal content is correct.	—
Airport ground crews	—	Ground crews are directed to perform certain functions based on information received from the pilot. If that information is incorrect, the consequences could be anything from wasted time to injuries.	Ground crews are directed to perform certain functions based on information received from the radar. If that information is incorrect, the consequences could be anything from wasted time to injuries.	Ground crews are directed to perform certain functions based on information received from the ATC system. If that information is incorrect, the consequences could be anything from wasted time to injuries.

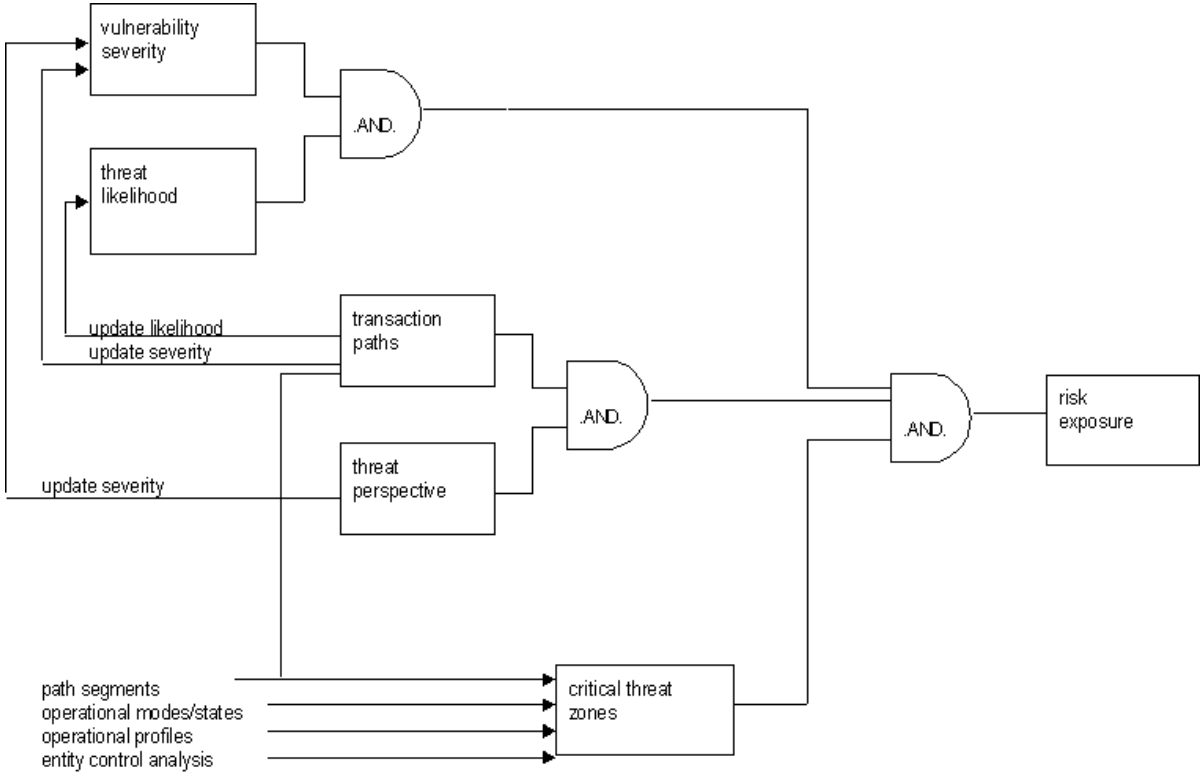

Exhibit 27 Components of Risk Exposure and Their Interaction

control measures is dependent on the thoroughness of the analysis of risk exposure which preceded it.

Risk exposure is determined and reviewed in stages:

1. Initial risk exposure is ascertained to prioritize threat control measures (Chapter 5).
2. Threat control measures are implemented in accordance with these priorities (Chapter 6).
3. The effectiveness of threat control measures (residual risk exposure) is verified (Chapter 7).

The problem remaining is how to decide what constitutes acceptable risk. There are few absolutes to guide the determination of risk acceptability. While laws and regulations provide some guidance, most often the decision has to be made on a case-by-case basis. One approach, developed by the U.K. Health and Safety Executive, is to correlate threat likelihood and vulnerability severity. (This is similar to the approach discussed in an earlier Chapter section ("Identify Threats, Their Type, Source, and Likelihood) for prioritizing threat control measures.) The intent is to discover whether each threat likelihood and vulnerability severity pair has been reduced: (1) to a level that is acceptable, and (2) as low as reasonably practicable (ALARP). A variety of factors can influence the acceptability decision, including the system's mission, availability of alternatives, state of technology, social considerations, economic considerations, contractual requirements, etc.

A second approach has been developed by Nordland[367] that models risk acceptability for rail transportation systems. In addition to the parameters mentioned above, Nordland notes that risk aversion varies by region, country, time in history, and current political situation. The basic model is[367]:

$$
\begin{aligned}
T &= \text{Risk acceptability} \\
&= b/\ (C/A * A/J * dA/dt * f(c)) \\
&= b/\ (C/J * dA/dt * f(c))
\end{aligned}
$$

where:

C/A	=	Average number of casualties per accident
A/J	=	Average number of accidents per journey
dA/dt	=	Distribution of accidents over time
f(c)	=	Differential risk aversion factor
b	=	Factor describing the benefit of the system

This model can be used as-is for other transportation systems, such as air traffic control systems, marine navigation systems, and intelligent vehicle systems. This model can also be easily applied to other situations by substituting appropriate factors for journey, such as operational time, and expanding

casualty to include items such as financial loss and environmental damage. To illustrate, the model could be adapted as follows for a financial system:

$$
\begin{aligned}
T &= \text{Risk acceptability} \\
&= b/\ (C/A * A/J * dA/dt * f(c)) \\
&= b/\ (C/J * dA/dt * f(c))
\end{aligned}
$$

where:

C/A	=	Average number of errors per transaction
A/J	=	Average number of transactions per 24-hour day
dA/dt	=	Distribution of errors over time
f(c)	=	Differential risk aversion factor
b	=	Factor describing the benefit of the system

In addition, the model could be further refined by distinguishing the severity of the errors C_1 … C_4 (insignificant … catastrophic). Similar modifications can be made to adapt this model to the telecommunications industry, the nuclear power industry, etc.

Optimally, a combination of both approaches would be used to determine risk acceptability. In the final analysis, risk acceptability should tie back to specified IA goals.

5.6 Summary

The second component of an effective information security/IA program is the vulnerability and threat analyses. Four activities are performed during the vulnerability and threat analyses, as shown in Exhibit 28:

- IA analysis techniques are selected and used.
- Vulnerabilities, their type, source, and severity are identified.
- Threats, their type, source, and likelihood are identified.
- Transaction paths, threats zones, and risk exposure are evaluated.

The terms "vulnerability," "threat," "hazard," and "risk" are often (incorrectly) used interchangeably. These four concepts are related, as shown in Exhibit 1; however, they have distinct meanings. This distinction must be maintained when performing the vulnerability and threat analyses or the results will be confusing and misleading.

Vulnerabilities and threats are identified and characterized so that resources can be applied to the most critical need(s) to prevent loss. The identification and analysis of vulnerabilities and threats considers accidental **and** intentional action **and** inaction. Individual events as well as unusual unplanned combinations and sequences of events that could lead to a failure/compromise are analyzed.

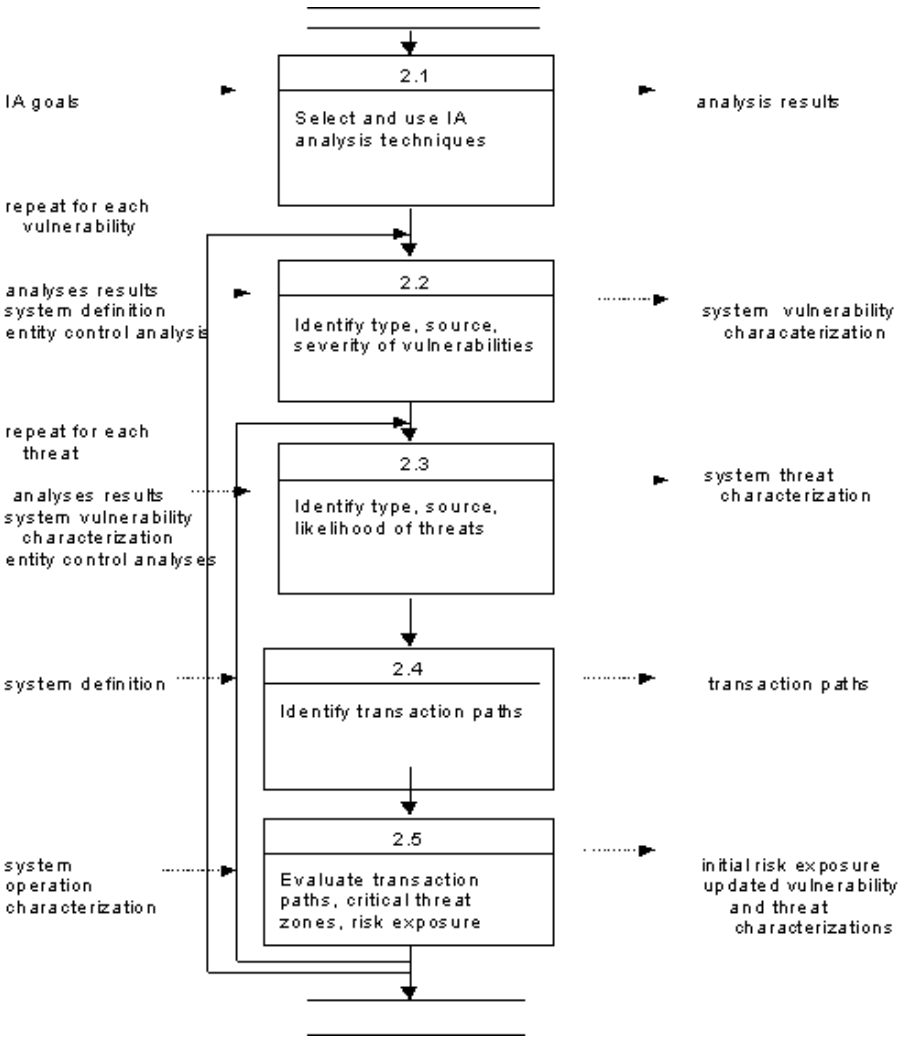

Exhibit 28 Summary of the Activities Involved in Performing Vulnerability and Threat Analyses

Transaction paths are developed to identify all logically possible combinations of discrete activities that could cause a system to be compromised or rendered inoperable. Transaction paths can be developed *a priori* to determine the need for threat control measures, or *a posteriori* as part of an accident/incident investigation. Transaction paths are analyzed from the perspective of multiple stakeholders to refine vulnerability/threat assessments and uncover critical threat zones. Risk exposure is derived from correlating vulnerability severity and threat instantiation likelihood, analyzing transaction paths, and isolating critical threat zones.

Next, Chapter 6 explains how to proactively defend systems and data against loss through IA design techniques and features.

5.7 Discussion Problems

1. When are IA analysis techniques performed? By whom are they performed?
2. For what type of threats are quantitative likelihood estimations most appropriate? For what type of threats are qualitative likelihood estimations most appropriate? Explain your reasoning.
3. What criteria should be used to select IA analysis techniques?
4. Explain the relationship, if any, between: (a) hazards and vulnerabilities, (b) hazards and risk, (c) threats and risk, and (d) vulnerabilities and threats. Give an example of each.
5. What causes vulnerabilities?
6. Identify an example of a potential system failure or compromise for each severity and likelihood level.
7. Which is more important to analyze: (a) an individual event, (b) a sequence of events, or (c) a combination of events? Why?
8. Postulate failure scenarios for the telecommunications backbone, power source, and environmental controls for the online banking example.
9. Would an intentional vulnerability be direct or indirect? Why?
10. Characterize the vulnerabilities associated with routers listed in the chapter section "Identify Vulnerabilities, Their Type, Souce, and Severity."
11. Which is easier to prevent: a vulnerability or a threat? Why?
12. Cite potential examples of a safety vulnerability, a reliability vulnerability, a security vulnerability, and a vulnerability caused by a combination of factors for an air traffic control system.
13. How is the entity control analysis used during the assessment of threat likelihood?
14. What does a transaction path represent? How does it relate to risk exposure?
15. Which threat perspective should be considered when evaluating threat zones? Why?
16. Develop a threat characterization summary for the COTS vulnerabilities listed in Exhibit 10.