

Chapter 4

Define the System Boundaries

To be effective, an information security/IA program must correspond to the reality of today's technology: distributed processing; client/server applications; mobile code; integrated audio, video, image, and textual data; PLCs; ASICs; embedded systems; wireless communications; and, of course, the Internet. An integrated methodical approach is needed: one that is comprehensive in scope, encompassing safety, reliability, and security engineering. Information security/IA is not just a software challenge; rather, it involves dynamic interactions within and among a multitude of hardware, software, telecommunications, and people.

Many organizations take a haphazard approach to information security/IA. If for no other reason than listening to the evening news, they are cognizant of the fact that something should be done to protect their IT base. So a firewall and virus scanner are installed, users are assigned passwords, and possibly e-mail is encrypted. However, the effectiveness of these measures is quite limited due to the lack of planning, analysis, and coordination that preceded them; a solution was implemented without defining the problem.

This chapter describes the initial component of an information security/IA program — defining the boundaries of the system to be protected. This component is comprised of four activities:

1. Determining what is being protected and why
2. Identifying the system
3. Characterizing system operation
4. Ascertaining what one does and does not have control over

These activities are straightforward. It is essential that they be performed — one must know what one is protecting before an effective strategy for

doing so can be developed. As Jesty³⁰⁸ reports, one of the first challenges in certifying the safety, reliability, or security of a system — particularly intelligent transportation systems (ITS) — is to define the boundaries and components of a system. Likewise, the first step in the U.K. Central Computing and Telecommunications Agency (CCTA) Risk Analysis and Management Methodology (CRAMM), developed in 1991, and its successors BS 7799^{20,21} and ISO/IEC 17799(2000-12),¹²³ is to define the boundaries of a system.²⁰⁸

4.1 Determine What is Being Protected and Why

Webster's dictionary defines *protect* as:

- vt. (1) to cover or shield from exposure, injury, or destruction, guard;
- (2) to maintain the status or integrity of.

The purpose of information security/IA is to protect critical systems and data. Before this can be accomplished, the systems and data being protected need to be identified. Not all systems or system components may need to be protected; nor will they all need the same level of protection. Therefore, the first step is to define what is being protected:

- Systems that process or generate data?
- Systems that display data?
- Backup, archival, or online storage systems?
- Control systems that act on real-time data?
- Communications systems?
- Voice, video, image, or textual data?
- Hardcopy output?
- Input devices?

The next step is to define why these items need to be protected. The specific rationale and purpose for protecting each system and component should be explained. Information security/IA activities should not be undertaken just because “everyone else is doing it.” Rather, information security/IA activities should be undertaken with the intent of accomplishing specific goals for specific reasons. This is common sense because goals must be articulated before they can be achieved. It also ensures that systems and components are not over- or under-protected. A clear, concise, unambiguous statement of the IA goals (what is being protected) and a justification for those goals (why these items are being protected) are needed. This statement should focus on what is to be accomplished — not how it is accomplished. The “how” is determined later. The goals should be stated succinctly and in a manner such that their achievement can be easily verified. For large systems, it may be useful to express these goals hierarchically, with a limit of three or four levels of detail (see [Exhibit 1](#)).

Exhibit 1 Sample Statement of IA Goals

<i>Goal</i>		<i>Justification</i>	
1.	Protect the privacy and integrity of customer records from accidental or malicious intentional unauthorized disclosure, manipulation, alteration, abuse, corruption, and theft.	1.a	Customer loyalty depends on sound business ethics.
		1.b	Local (or national) regulations require privacy protections.
1.1	Protect personal identifying information: name, address, phone number, e-mail address, account number, and fax number.	1.c	Liability lawsuits may result from a failure to protect customer records.
		1.d	Fraud lawsuits may result from a failure to protect customer records.
1.2	Protect customer payment information and history.		
1.3	Protect customer purchase history and preferences.		
1.4	Protect customer online, voice, fax, and hardcopy transactions.		

4.2 Identify the System

Webster’s dictionary defines a system as:

a regularly interacting or interdependent group of items forming a unified whole.

Similarly, IEEE Std. 610.12-1990* defines a system as:

a collection of components organized to accomplish a specific function or set of functions.

The common theme between these definitions is that a system is composed of smaller parts that cooperate to accomplish something. Within the IT domain, systems are generally considered to be composed of subsystems, components, and subcomponents, as shown in [Exhibit 2](#). However, what constitutes a system is relative to one’s vantage point. What one person considers a system, another person might consider a subsystem or a collection of systems. In other words, abstractions about systems and their constituent components can go to very high and very low levels, depending on one’s perspective and the purpose of the abstractions. The lack of specificity in terminology defining what is a system versus a subsystem or component is one reason why all stakeholders should be involved in defining the boundaries of a system.

In the IT world, it is common to think of systems as consisting of only hardware, software, and telecommunications equipment. Information security/IA takes a much broader view of systems, adding items such as people,

* ANSI/IEEE Std. 610.12-1990, Standard Glossary of Software Engineering Terminology.⁴⁴

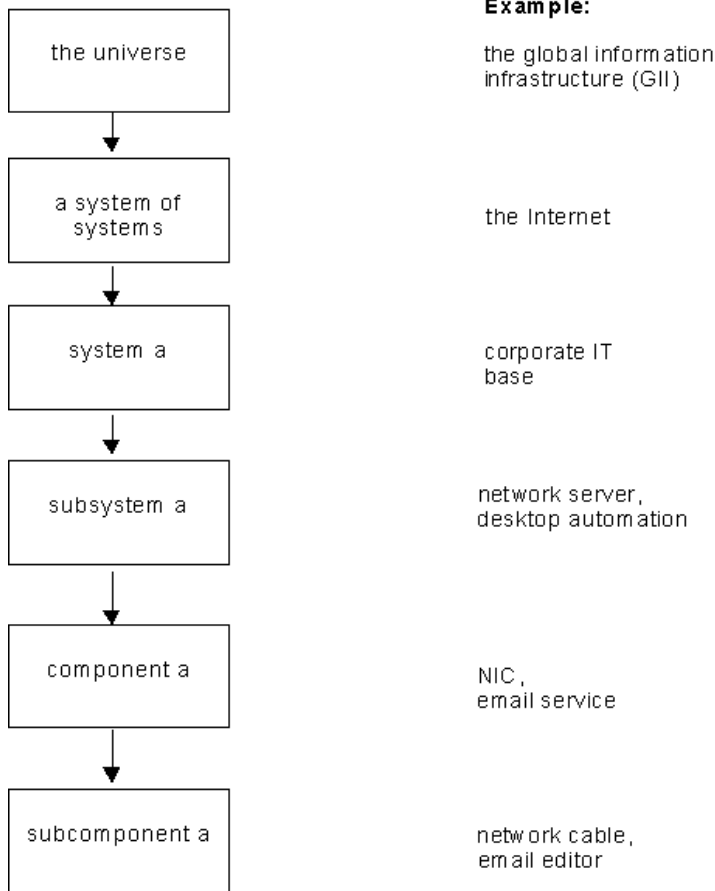


Exhibit 2 Standard Hierarchy Used in System Definition

operational procedures, and the supporting infrastructure systems to the equation. As such, systems include logical and physical, animate and inanimate, primary and support, dynamic and static entities. The following are examples of each type of system entity:

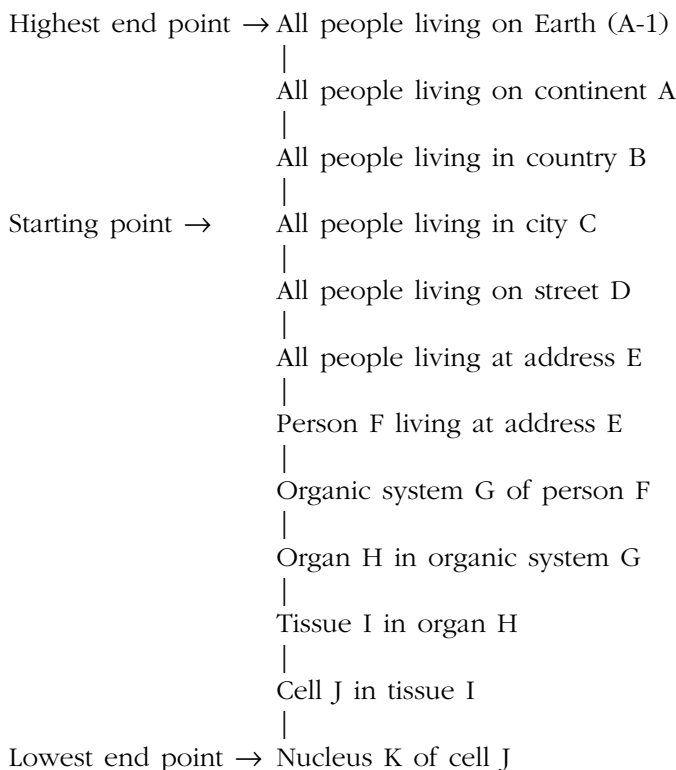
- **Logical:** Software is a logical entity.
- **Physical:** Software executes and is stored on physical entities such as computers, hard drives, floppy drives, PROMs, PLCs, and ASICs.
- **Animate:** Human users, system administrators, trainers, and maintenance staff are the animate entities within a system.
- **Inanimate:** All other system entities are inanimate; for example, system archives.
- **Primary:** Primary entities are those that contribute directly to accomplishing a system's function; for example the CPU, operating system, applications software, and end users.
- **Support:** The electric power grid and the telecommunications backbone are examples of support entities, as are most infrastructure systems.

They are essential but contribute indirectly to the accomplishment of a system's function.

- **Dynamic:** System configurations and operational procedures are dynamic entities. Both tend to evolve or be modified frequently over the life of a system, due to enhancements, maintenance, and changes in technology. A change in a dynamic entity should trigger the revalidation of protection strategies.
- **Static:** The entities that are static will vary from system to system. In one case, a maintenance schedule may be static; in another, the electromechanical components may be static.

Note that an item may fall into more than one entity type. For example, an item could be logical, primary, and dynamic. Only the pairs are mutually exclusive: logical/physical, animate/inanimate, primary/support, and dynamic/static. In general, different protection strategies are needed for different entity types.

To identify the boundaries of a system, one must first pick a starting point or prime entity from which to work upward to identify the outer limits of the system and downward to identify the constituent subsystems, components, and subcomponents. These are the two end points. To illustrate, suppose one is trying to define a demographic system in order to perform an epidemiological study. One picks the people living in city C as the starting point:



On a case-by-case basis, a determination is made about the level to which it is meaningful to carry the identification process, both upward and downward.

For example, epidemiological studies require a statistically significant group of people. Hence, items E–K, A-1, and A would not be considered meaningful.

Once the upper and lower limits of the system have been established, the system definition should be formally documented as shown in [Exhibits 3 and 4](#). In this example, the boundaries of a radiation therapy system are being defined. The graphical system definition in [Exhibit 3](#) helps establish which entities are inside and outside the system boundary. External entities may be optional or mandatory; hence, it is important to capture them. Often, but not always, links to external entities such as infrastructure systems will be through internal support entities. Some sources talk about unbounded systems, especially when discussing Internet applications or interaction between mission- or business-critical systems and infrastructure systems. The concept of an unbounded system is useful to denote the interaction and interdependency between systems. However, information security/IA activities must be focused on a system that has defined boundaries and distinguishes between internal and external entities.

The tabular system definition in [Exhibit 4](#) captures a lower level of detail about the system and characterizes the entities. These two charts reinforce each other and promote a thorough system definition. The formal process of documenting the system definition ensures that entities are not left out or overlooked. Defining the boundaries of a system also helps to designate organizational responsibility for information security/IA activities. In this way, responsibility can be assigned to the organizational component that can carry out information security/IA activities most efficiently. Duplication of effort is also minimized. For large organizations with geographically dispersed enterprisewide systems, defining system boundaries and information security/IA responsibilities is a crucial step. The system definition should be reviewed and approved by all stakeholders. It is common today to automatically assign total responsibility for security to the system or network administrator. By all means, they should be involved in information security/IA activities and analyses, but as one of many participating stakeholders.

4.3 Characterize System Operation

Thus far, we have (1) determined what is being protected and why, and (2) defined the boundaries of the system. The next step is to characterize the system operation. A system operation characterization takes two forms: operational modes or states and operational profiles. This information serves as input to the vulnerability and threat analyses discussed in Chapter 5.

An operational mode or state represents one of several states or modes in which a system or system entity can exist. Operational modes and states may or may not be mutually exclusive. Some operational modes and states are common to most systems, such as:

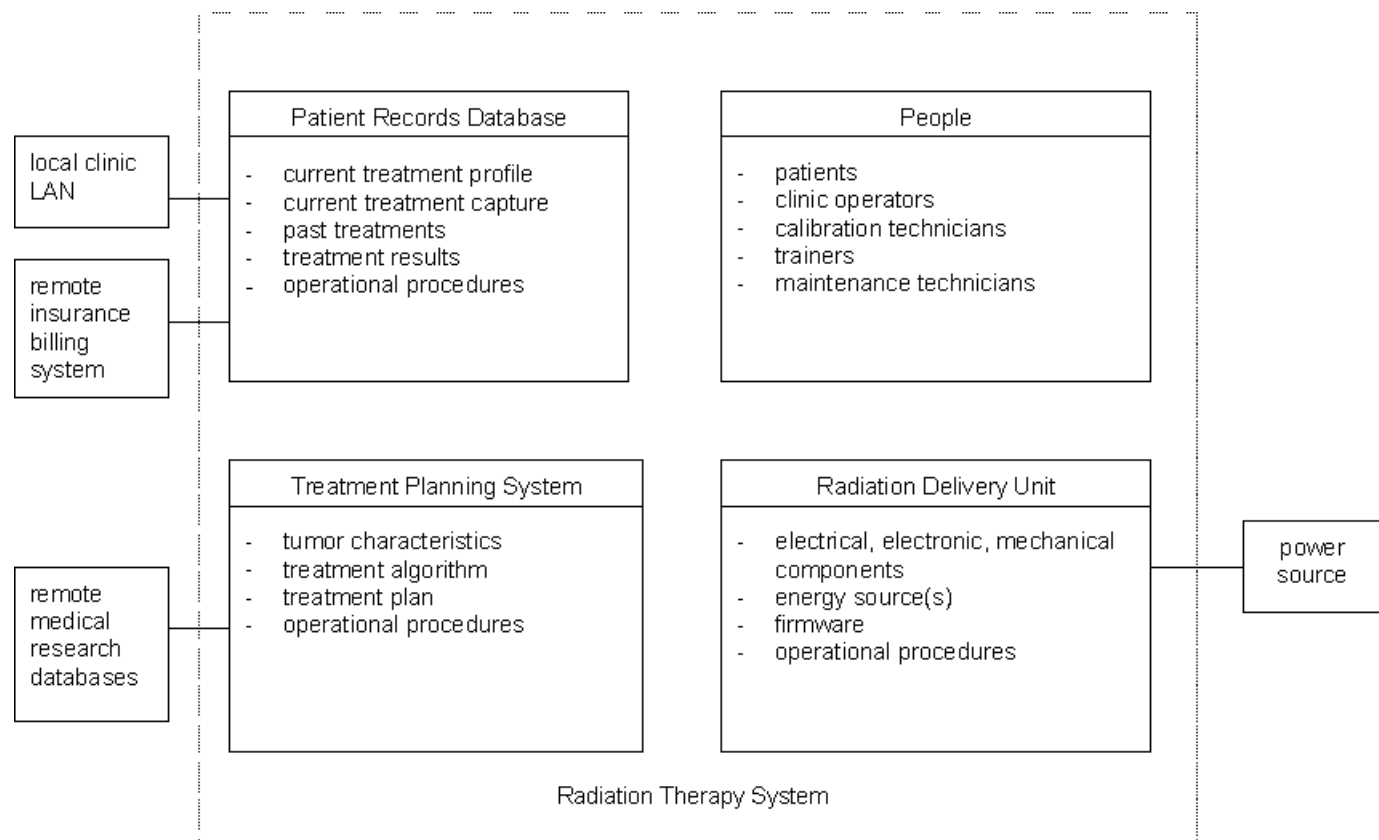


Exhibit 3 Sample High-Level System Definition

Exhibit 4 Sample High-Level System Definition

System: Radiation Therapy System as of: 20 March 2000

<i>Subsystem</i>	<i>Component</i>	<i>Subcomponent</i>	<i>L/P</i>	<i>A/I</i>	<i>P/S</i>	<i>D/S</i>
1. People	1.1 Patients	—	P	A	P	D
	1.2 Clinical operators	—	P	A	P	D
	1.3 Calibration staff	—	P	A	S	D
	1.4 Maintenance staff	—	P	A	S	D
	1.5 Training staff	—	P	A	S	D
2. Patient records DBMS	2.1 Treatment profile	2.x.1 Data records	L	I	S	D
	2.2 Current treatment capture	2.x.2 Record management capability	L	I	S	D
	2.3 Past treatments	2.x.3 Report generation capability	L	I	S	S
	2.4 Treatment results	2.x.4 Query/response capability	L	I	S	D
	2.5 Operational procedures	2.x.5 Backup/archive capability	L	I	S	D
	**2.6 Local clinic LAN	—	P	I	S	D/S
	**2.7 Remote insurance/billing system	—	L	I	S	D/S
	3.1 Tumor characteristics	—	L	I	P	D/S
3. Treatment planning system	3.2 Radiation therapy algorithm	3.2.1 Optional components or variations of algorithm	L	I	P	S
	3.3 Operational procedures	—	L	I	S	D/S
	3.4 Treatment plan x	3.4.1 Dosage	L	I	P	D/S
		3.4.2 Targeting information				
		3.4.3 Number of sessions				
	**3.5 Remote medical research databases	—	L	I	S	D/S
4. Radiation delivery system	4.1 Electrical, electronic, and mechanical components	4.1.x Subassemblies	P	I	P	S
	**4.2 Energy source(s)	4.2.1 Energy delivery system	P	I	P	S
		4.2.2 Power supply				
	4.3 Operational procedures	4.3.1 Maintenance schedule and procedures	L	I	S	D/S
		4.3.2 Calibration schedule and procedures	L	I	S	D/S
		4.3.3 Patient use procedures	L	I	P	D/S

Note: L/P, logical or physical entity; A/I, animate or inanimate entity; P/S, primary or support entity; D/S, dynamic or static; and **, external entity.

- Normal operations:
 - start-up
 - shutdown
 - reconfiguration
 - restart/reset
 - backup
 - standby
 - maintenance
 - decommission
 - perform normal system-specific functions
- Abnormal operations:
 - failure of system hardware
 - failure of system or application software
 - operator error
 - degraded mode operations
 - shutdown under abnormal conditions (e.g., an attack)

Operational modes and states can be further characterized by performance and reliability constraints, such as response times, processor load/capacity, bandwidth requirements, sequencing of state transitions, etc. The level and type of information that is useful is decided on a case-by-case basis.

Operational profiles are a direct corollary to operational modes and states. Operational profiles or scenarios represent the set of operations that a system can execute.³⁴³ While operational modes and states only consider the inanimate entities of a system, operational profiles also take into account the human component. Operational profiles depict how humans interact with a system to accomplish tasks, through an analysis of operational scenarios, user views, and system events. They capture domain knowledge about how a system can be (and in reality is) used.

Operational profiles are often developed to support reliability engineering analyses. These operational profiles focus on end users. For information security/IA purposes, operational profiles should also be developed for maintenance staff, trainers, system administrators, super-users, testers, and potential intruders. Sometimes it is helpful to devise operational profiles graphically, using a tree notation. Operational profiles developed for reliability purposes often assign a probability that each action will be performed, or alternatively prorate a user's time among all possible activities. Whether or not this additional level of detail is useful is decided on a case-by-case basis.

Exhibit 5 presents a sample high-level system operational characterization, continuing the radiation therapy system example. The operational modes and states are listed. An indication is given as to whether or not a mode occurs before, after, or during another mode. Constraints associated with activating or transitioning to a mode and what agents can initiate a mode are identified. Next, operational profiles are developed by type of operator. In this example, there are three types of end users, maintenance staff, a system administrator, trainers, and potential intruders. The primary activities performed by each

Exhibit 5 Sample High-Level System Operation Characterization

System: Radiation Therapy System as of: 30 March 2000

I. Operational Modes and States

<i>Mode/State</i>	<i>Occurs Before</i>	<i>Occurs After</i>	<i>Occurs During</i>	<i>Constraints</i>	<i>Initiated by</i>
<i>Normal Operations</i>					
Start-up	All other modes	—	—	Power availability, absence of system fault	System administrator, maintenance staff
Shutdown	—	All other modes	—	System has been safed, records saved	System administrator, maintenance staff
Reconfiguration	Shutdown	Start-up	—	No end users active	System administrator, maintenance staff
Restart/reset	Shutdown	Start-up	—	No end users active	System administrator, maintenance staff
Backup	Shutdown	Start-up	—	No end users active	System administrator, maintenance staff
Standby	Shutdown	Start-up	—	No end users active	System administrator, maintenance staff
Maintenance	Shutdown	Start-up	—	No end users active	System administrator, maintenance staff
Decommission	Shutdown	Start-up	—	System has been safed, no end users active	System administrator, maintenance staff
Perform normal system-specific functions	Shutdown	Start-up	Varies	System resources are available	All except intruders
<i>Abnormal Operations</i>					
Failure of patient records database	Shutdown	Start-up	—	Failure must not cause safety and/or security violation	Operator error, system HW/SW fault
Failure of treatment planning system	Shutdown	Start-up	—	Failure must not cause safety and/or security violation	Operator error, system HW/SW fault

Exhibit 5 Sample High-Level System Operation Characterization (continued)

<i>Mode/State</i>	<i>Occurs Before</i>	<i>Occurs After</i>	<i>Occurs During</i>	<i>Constraints</i>	<i>Initiated by</i>
Failure of radiation treatment unit	Shutdown	Start-up	—	Failure must not cause safety violation	Operator error, system HW/SW fault
Degraded mode operations	Shutdown	Start-up, system failure	—	Criteria for transferring to degraded mode operations must be defined and met	System software and/or system administrator

II. Operational Profiles

<i>Operator</i>	<i>Primary Activities</i>	<i>Time Distribution</i>	<i>Sequencing, Timing, or Other Restrictions</i>
End user a	Logon	5%	Patient records must be initialized before any other transactions can take place
	Access, enter, store, forward patient records	90%	
	Logoff	5%	

operator are discerned. Time on the system is allocated among these activities. Any restrictions related to performing these activities are noted.

4.4 Ascertain What One Does and Does Not Have Control Over

The final activity in defining the boundaries of a system is to ascertain what system entities one does and does not have control over. This information is crucial input to the vulnerability and threat analyses discussed in Chapter 5.

The level of control the system owner has over each entity is determined using the system definition charts ([Exhibits 3 and 4](#)) as input. The level of control is determined for all identified internal and external entities. The level of detail for which control status is identified corresponds to the level of detail in the system definition charts. As shown in [Exhibit 6](#), the first three columns of the system entity control analysis are taken directly from the system definition charts. Two new columns are added: control status and explanation. The control status records the degree of control or responsibility a system owner has over the accurate functioning of an entity. The control status can be either total, partial, or none. These terms are defined as follows:

- **Total control:** System owner has total control over and responsibility for an entity, the correctness and performance of its actions.
- **Partial control:** System owner shares control over and responsibility for an entity, the correctness and performance of its actions with one or more second parties, usually through a legal mechanism such as a contract.
- **None:** System owner has no control over or responsibility for an entity, but is dependent on the services it provides. One or more third parties have this responsibility and control. Infrastructure systems are a good example.

A brief rationale for the assigned control status is given in the explanation column. Most system owners are surprised to discover how few entities they have total control over. This discovery has a significant impact on the vulnerability and threat analyses, as well as the development of contingency plans.

4.5 Summary

The first component of an effective information security/IA program is to define the boundaries of a system. There are four activities involved in defining the boundaries of a system, as listed below and summarized in [Exhibit 7](#):

- Determining what is being protected and why
- Identifying the system
- Characterizing system operation
- Ascertaining what one does and does not have control over

Exhibit 6 Sample High-Level System Entity Control Analysis

<i>Subsystem</i>	<i>Component</i>	<i>Subcomponent</i>	<i>Control Status</i>	<i>Explanation</i>
1. People	1.1 Patients	—	None	Patients are not employees or otherwise under contract to the clinic.
	1.2 Clinical operators	—	Total	All legitimate operators are clinic employees.
	1.3 Calibration staff	—	Partial	Calibration staff are under contract to the clinic.
	1.4 Maintenance staff	—	Partial	Maintenance staff are under contract to the clinic.
	1.5 Training staff	—	Partial	Trainers are under contract to the clinic.
2. Patient records DBMS	2.1 Treatment profile	2.x.1 Data records	Total	Clinic owns patient records.
	2.2 Current treatment capture	2.x.2 Record management capability	None	DBMS application software is provided and maintained by vendor.
	2.3 Past treatments	2.x.3 Report generation capability		
	2.4 Treatment results	2.x.4 Query/response capability		
	2.5 Operational procedures	2.x.5 Backup/archive capability	Partial	Clinic owns backup/archive records. Vendor owns software that generates backups.
	**2.6 Local clinic LAN	—	Partial	Clinic contracts for LAN services.
	**2.7 Remote insurance/billing system	—	None	Third party maintains insurance/billing databases.

Exhibit 6 Sample High-Level System Entity Control Analysis (continued)

<i>Subsystem</i>	<i>Component</i>	<i>Subcomponent</i>	<i>Control Status</i>	<i>Explanation</i>
3. Treatment planning system	3.1 Tumor characteristics	—	Total	Clinic owns patient records.
	3.2 Radiation therapy algorithm	3.2.1 Optional components or variations of algorithm	Partial	Clinic implements specific instance of algorithm. Vendor owns application software.
	3.3 Operational procedures	—	Partial	Clinic is responsible for enforcing operational procedures. Vendor is responsible for developing operational procedures.
	3.4 Treatment plan x	3.4.1 Dosage 3.4.2 Targeting information 3.4.3 Number of sessions	Total	Clinic employee develops specific treatment plan.
	**3.5 Remote medical research databases	—	None	Clinic neither creates or maintains research databases; a third party does.
4. Radiation delivery system	4.1 Electrical, electronic, and mechanical components	4.1.x Subassemblies	None	Vendor has total responsibility.
	**4.2 Energy sources	4.2.1 Energy delivery system 4.2.2 Power supply	None	Power company and vendor have responsibility.
	4.3 Operational procedures	4.3.1 Maintenance schedule and procedures 4.3.2 Calibration schedule and procedures 4.3.3 Patient use procedures	Partial	Clinic is responsible for enforcing procedures. Vendor is responsible for developing accurate procedures.

Note: **, external entity.

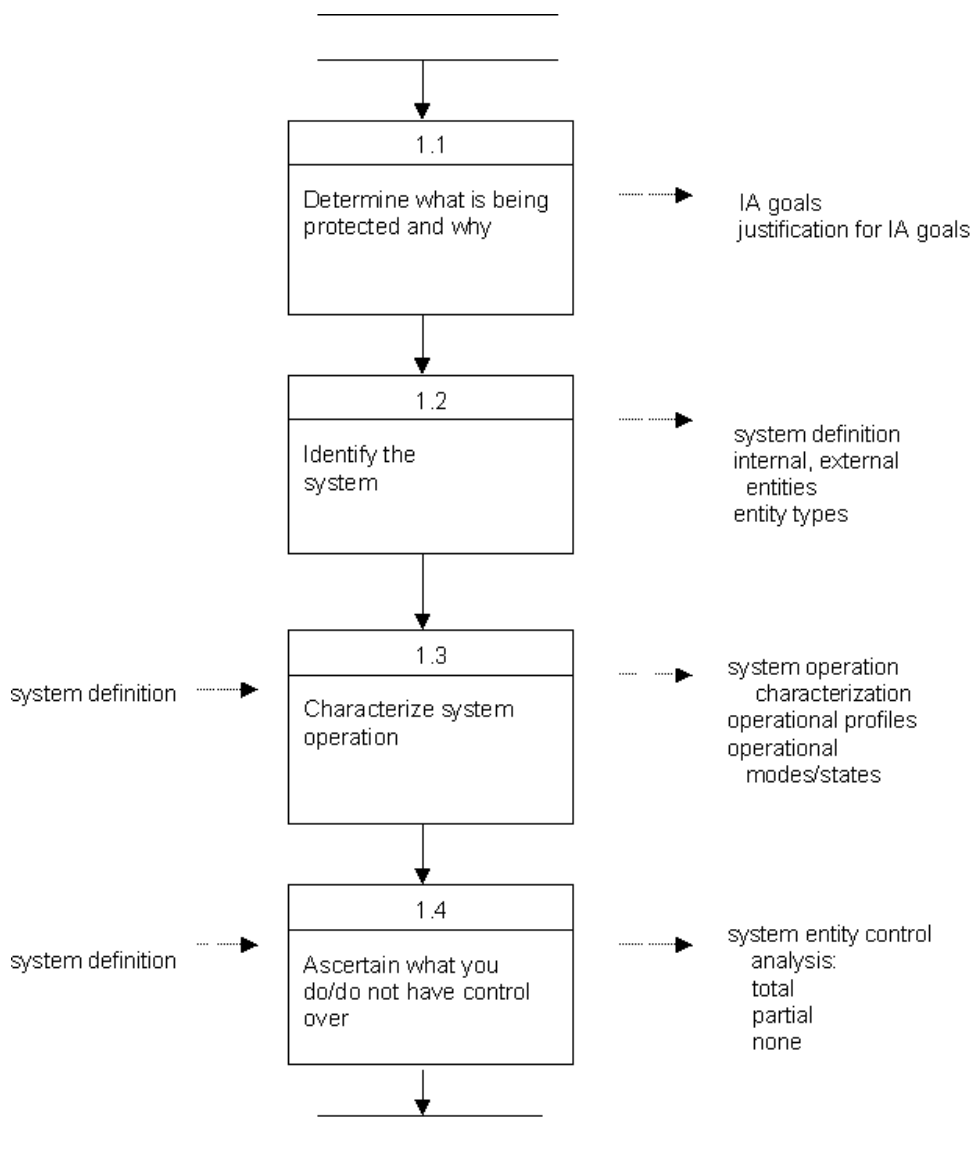


Exhibit 7 Summary of Activities Involved in Defining System Boundaries

This component is crucial; one must know what one is protecting and why before effective protection strategies can be developed. Expressed another way, an organized attack begins with a specific target in mind; hence, it is only logical that protection strategies should also be developed with specific targets in mind.

The results of the analyses conducted while defining the boundaries of a system provide essential input to other components of an information security/IA program. For example, the system definition, system operation characterization, and system entity control analysis are input to the vulnerability and threat analyses, while the information assurance goals are input to implementing threat control measures.

All stakeholders should be involved in defining the information assurance goals, developing the system definition and system operation characterization, and performing the system entity control analysis. This will ensure that all aspects of a system, its entities and operation, are included in the analyses. The formal process of conducting these analyses also helps identify organizational responsibility for information security/IA activities.

The boundaries of a system should be defined before a new system is deployed and whenever a system is enhanced or modified through a maintenance activity. The results of these analyses should be periodically reviewed to ensure that they remain accurate. In addition, the boundaries of a system are (re)defined as part of an accident/incident investigation.

Next, Chapter 5 explains how and why to conduct vulnerability/threat analyses.

4.6 Discussion Problems

1. Why should information security/IA activities be undertaken?
2. How and by whom are information assurance goals developed?
3. Describe the internal and external entities for a generic online business.
4. Why would or would not it be useful to include probability of occurrence in an operational profile?
5. What distinguishes quality IA goals?
6. How is information that is generated while defining the system boundaries used?
7. When is an accident/incident investigation performed?
8. Identify the entity type(s) for each of the following items and explain your rationale: (a) bar code reader, (b) DVD, (c) ISP, (d) e-mail system, (e) credit verification system, (f) WAN, (g) UPS, (h) antenna, (i) fiber optic cable, (j) user and maintenance manuals, and (k) company that publishes the manuals in item (j).
9. What is the difference between: (a) an operational mode and an operational profile; and (b) an operational mode and an operational state?
10. Why should or should not system operational characterizations include interaction with external entities?
11. What is the purpose of performing the system entity control analysis?
12. What commonalities exist between the development of protection strategies and an organized attack?
13. Discuss the fringe benefits of defining system boundaries.
14. Why is it possible for different people to define the boundaries of a system differently? How should these differences be resolved?