Chapter 4: INFOSEC – Protecting UAS Information Channels & Components

This chapter provides an overview of the basic concepts of information security to provide a common set of terms and concepts for discussion and analysis. This is only an overview: students interested in learning more should be aware that this discussion is cursory and there is a wealth of knowledge out there to be discovered.

Student learning objectives. After reading this chapter, students should be able to do the following:

- ❑ Identify, describe, and explain the three basic security policy questions
- ❑ Define the three commonly used security attributes
- ❑ Explain how security requirements can be systematically derived
- ❑ Identify and describe the three security engineering phases
- ❑ Explain the detection timeline
- ❑ Differentiate between the classes of problems that need to be detected
- ❑ Describe the types of activities that should be triggered by a detection event
- ❑ Extrapolate types of security challenges for UAS
- ❑ Identify how UAS information security challenges can be analyzed

Basic Concepts in Information Security

Information security is as old as information itself. Information has value; where the oasis is located, how many warriors are in the opposing force, how to safely prepare medicines, etc. Information security is not just about keeping secrets, although that is an important aspect. In this section, we will describe the basic concepts of information security and define important terms. Examples will illustrate concepts. Keep in mind that this is an abstraction of a very real problem space. What you will be exploring is a model of information security. Like any model, it simplifies and approximates reality to aid conceptualization. Just like an organizational chart does not capture the complexity of office politics or a data flow diagram does not capture the quality of the data, a security model has limitations. Its usefulness lies in that it gives you a way of understanding and analyzing something enormously complex.

Policy Questions

Before any analysis can begin, you must know your starting point and your goal. We refer to that as starting at first principles. **We can start with the most obvious question; what information needs to be protected?** To answer that question, you require substantial information. Depending on the size of the system or organization, distinct types of information can exist and require *varying levels of protection*. It seems like such a simple question, but it is deceptive in its simplicity. For example, consider yourself a target for a security analysis. What information requires protection? Your personally identifiable information (PII) is an obvious choice; after all, that is a target for identity thieves. What else requires protection? Usernames, passwords, bank account information, and medical records are all sensitive information. There may also be some relationships you would prefer to keep secret.

One side note: the information and systems considered in this question are not only the ones owned and controlled by the enterprise, but also *custodial files*. The responsibilities associated with having custody of data, equipment, or personnel require special consideration. When accepting custodial responsibility, you are tacitly or explicitly accepting the responsibility to exercise due care and control of those assets. These considerations must be considered in your analysis of the policy questions. In other words, it is not just the assets that establish your baseline operational capability that you need to consider, but also the set of assets that flow in and out of your enterprise. Two tools that can help you focus on where these challenges might lie are *data flow diagrams* and *functional decomposition diagrams*.

How Much Protection is needed?

Each asset has different security needs as well, so **another question needs to be asked; how much protection is needed?** Some information types require a great deal of protection, possibly layers of security. Others might require less. Another way of thinking about this question is what is the minimal amount of security required? Think of common types of information in your life. How much protection does your personal data (like your social security number) require as compared to information about your residence? For the majority (those who are not in the witness protection program, for example), personal data warrants more protection than their address and phone number. Understanding these distinctions helps determine how resources should be allocated in developing and ensuring protective mechanisms.

How long must the information be protected?

In addition to the question of 'how much?' **there is also the question regarding how long such protection must be maintained.** Some protections must remain in place for substantial time, while others can be allowed to expire relatively quickly. For example, suppose one proposes marriage to their partner, how long would that information need to be protected? It depends on the variables, but the author believes we can safely say that once married, the information is no longer sensitive and public access is acceptable. In contrast, consider an espionage agent that has infiltrated the highest level of an adversary government. How long should this information be protected? An argument could be made that the information should be protected perpetually.

To summarize, there are three policy questions that need to be addressed for competent security analysis. Both the positive and negative versions of these questions need to be considered.

These three questions are:
— What information requires protection? What information does not?
— How much protection does asset require? Conversely, what is the minimum amount of protection required?
— How long must security be kept in place? Conversely, how soon can it lapse?

There are risk implications to the answers, which will be explored in Chapter 6. Every decision is a risk decision, but decisions are informed by the availability of resources and operational circumstances. It is impossible to be risk-free, so the decisions on how and when risks are accepted need to be made thoughtfully.

Security Attributes

The previous section discussed protection of information. Various aspects of protection are called "security attributes." There are three commonly used security attributes in information security. They are **confidentiality, integrity, and availability,** abbreviated as CIA in many publications. (Some researchers have proposed emphasis on other security attributes; keep an open mind about what security attributes are important in your unique operational environment.)

Confidentiality refers to the need to keep information, operations, and transactions secret. This applies to more than simply data. There **are four distinct kinds of secrets** that need to be kept; ***actual secrets, relationship secrets, sources and methods secrets, and operational secrets***. We differentiate between these types because protecting them requires different approaches. A relationship secret is not protected the same way a sources and methods secret is protected. However, and this is a very important point, these types are not mutually exclusive. A secret may require several of these characteristics simultaneously.

Consider some examples of the types of secrets to understand this concept. An example of *an actual secret* might be something you did when you were seven years old. Perhaps you ate your sister's cookie or broke a lamp and blamed the dog. Keeping that information secret requires that access to the actual information be restricted; only those who are granted the highest level of trust are granted access to that information. The gate keeper considers the access request and either grants it or not, based on the adjudication of trust and need to know. The more people who know increases the probability that the secret will be breached, so the decision to grant access is made on a need to know basis.

Relationship secrets are secrets in which the relationship are what need to be kept secret. Some relationships are regulated; there are legal requirements to keep them secret. An example of this would be the relationship between two corporations considering a merger: before an agreement is made, the government requires that the relationship between the organizations for the purposes of negotiating the merger be kept secret. Other times, the relationship secret is a matter of protecting its illicit nature, such as an extramarital affair or membership in a crime syndicate. And still others, the relationship between types of information is what requires protection. For example, if a company has different pay scales by gender, the relationship between employee gender and pay would be regarded as a secret. A very famous example of this type of secret was the role of "Deep Throat" during the Watergate affair. It was obvious that the data being leaked was coming from an insider and the data itself was being made public. From the perspective of the participants, the most important secret was the relationship between the reporters and the insider. They took extreme measures to prevent the relationship from being discovered.

***Sources and methods secrets* refer to the need to protect sources of information and methods by which operational goals are achieved.** Protecting sources and methods secrets can be tricky. For example, consider an advanced secret imaging platform that provides extremely high-resolution images of targets. These images are useful, and secret, but an additional consideration is protection of the source. Both the existence of the source and any clues as to the existence of the source must be protected. The latter is the difficult part: steps must be taken to remove or obfuscate any hint that the source exists. For our example of the imaging platform, one way to protect the source would be to share only degraded

versions of the images. Similarly, methods for achieving some goal might need protection. For example, the exact process of mixing ingredients in controlled humidity and temperature environments could result in a superior product. In this example, the ingredients for the product are known. It is the method that is the secret. Protecting this secret requires layering of operational protective measures.

Operational secrets are those secrets that are situationally based. The operational situation may vary significantly in time or geography. For example, it may be well known that a bombing raid is being planned, but it is the actual time of the attack that is the secret. Similarly, it may common knowledge that drones are in a general area, but the actual location at any point in time may be secret. These types of secrets are considered separately from the others because of their transient nature; as soon as the operational circumstances change, the secrecy requirements become moot.

Consideration of these distinct types of secrets is extremely helpful when attempting to answer the three policy questions. It is easier to address the 'what', 'how much', and 'how long' aspects when possessing a sophisticated understanding of what type of confidentiality challenge needs to be addressed.

Integrity refers to the need to ensure the unchanging and unchangeable nature of data and transactions. There are three types of integrity challenges to pay attention to: *data, transaction, and communications integrity*. During this discussion, we will also discuss the concept of *nonrepudiation* as an integrity concept.

Data integrity refers to the allowable and unallowable variances or changes in the actual data. This may seem like a strange concept: why would there be allowable variances? Shouldn't all data be kept pristine and whole? Like everything else, there are tradeoffs to consider. Consider image compression algorithms. Some are lossy, some are lossless. A lossy image compression algorithm loses data by design, to balance between image quality and file size. The higher resolution the image, the more bandwidth required to transmit it and the more storage is required to store it. It may be acceptable to lower data integrity for the image file if the result is operationally adequate. The key here is to understand what level of data integrity is required, what level of variance is allowable, and how to detect malicious variations to data integrity.

Transaction integrity refers to the unchanging and provable nature of a transaction. A well-formed transaction has several qualities; the initiator of a transaction provably initiates the transaction, the transaction is provably received by the intended and authorized recipient, and the transaction is provably unchanged during the exchange. For example, suppose one deposits a check into an account. The provability of the origination of the transaction is supported by one's signature on the check, identifying it for deposit to a specified account. The provability of the recipient of the transaction is supported by the bank receipt, further supported by the updated ledger entry showing the exact amount of the deposit. Each one of these steps could be subverted; there are many scams that do exactly that. For high transactional integrity significant protections are implemented to ensure a high degree of transaction correctness, consistency, and completeness. An example of a transaction that needs very high integrity protection would be the authentication protocol for sensitive area access.

A special case in integrity is referred to as *nonrepudiation*. The root word 'repudiate' means to deny. When someone repudiates something, they deny it. If either party can deny their participation in the transaction or its content, it would not be a well-formed transaction. Consider, what if one went to the bank, deposited a check, got a receipt, and then a few days later the bank claimed that it never happened? The bank repudiates the transaction. It is serious enough with one's own money but becomes even more serious when the transaction is a lawful order to execute a military mission. Neither party would want there to be an ability to repudiate the transaction. The person executing the order would want to prove that they are operating under a legitimate order from a superior, while the superior would want to prove that they ordered a lawful action in accordance with policy. As a result, engineering nonrepudiation into a system is an important consideration.

Communications integrity refers to the unchanging nature of data while it is being transmitted from one entity to another. In this aspect, our concern is that the message itself is not corrupted. Keep in mind that this aspect of integrity is not just for digital communications, but for all communications. The challenges to integrity can result from many different problems, including the system itself. A fun way to explore this problem is to play the game of telephone. In this game, many people sit in a circle. The first person whispers a message to the next person, as clearly as possible. The second person whispers the message to the third, the third to the fourth, and so on. Even if every single person tries as hard as they can to be clear and enunciate, the message at the end of the chain will bear little resemblance to the message that originated. The point being that to ensure the integrity of the communications, you need to pay attention to not only the message but also to the channel.

Availability refers to the need to be able to access and use data and systems when operationally required. Obvious? It is the most straight forward of the security attributes, but that does not mean that it is simple or trivial. What does it mean to have access when needed? For some organizations, it is simple. Maximum availability during operational hours and maintenance as required. But what if there is no down time? Then it becomes complicated. Consider a very large enterprise spanning multiple time zones; how does one calculate availability requirements across the entire enterprise? Is it 100% availability required? That is almost impossible without a very large investment. Even highly reliable phone systems only measure availability in "five nines", 99.999% available. When considering availability, one needs to account for the availability of many various aspects, including the data itself, the tools (including both computational and other, such as pens and paper), and the infrastructure components (including, again, both computationally based, such as networks, and physical, such as secure areas and safes).

In summary, there are three commonly used security attributes:

- Confidentiality;
- Integrity, with the special case of nonrepudiation, and
- Availability.

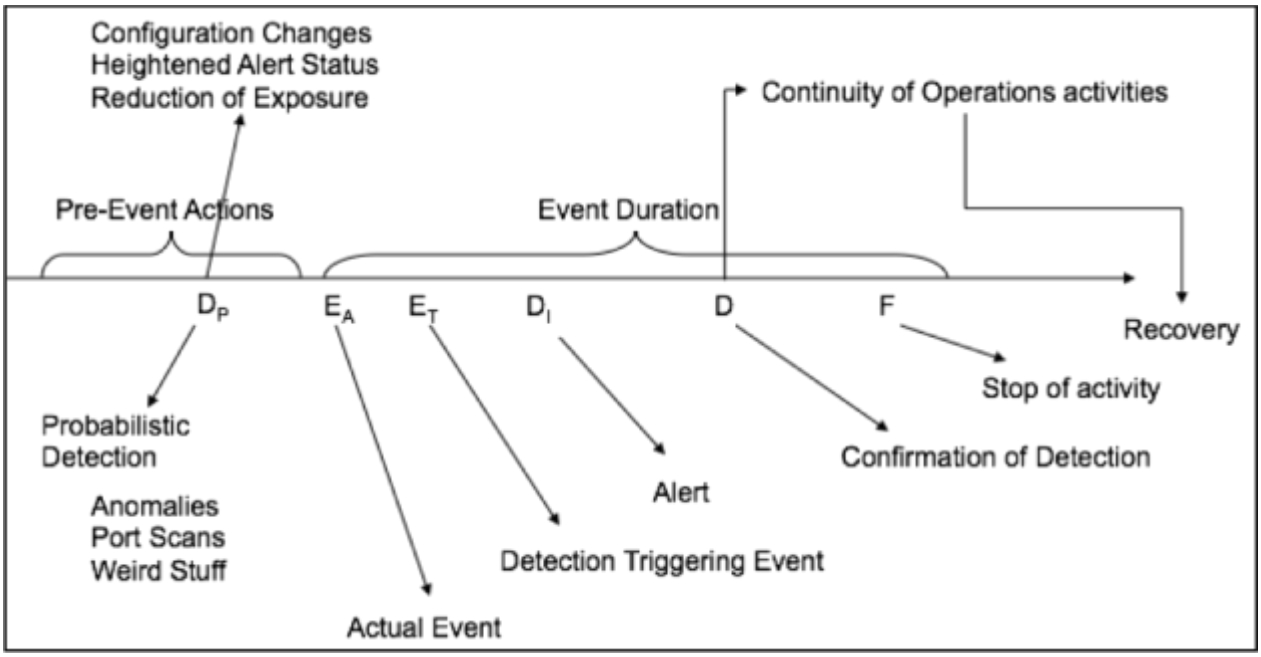
Security Phases

There are three security phases that we think about; protection, detection, and reaction/correction. There are two functions for these phases. The first use is to identify the viable solutions for the security requirements for an enterprise. The second use is to manage security operations for an enterprise.

Protection is the security engineering phase in which protective measures are implemented to meet security requirements. Protective mechanisms can consist of product technologies, processes, and methods. They come from the specialty areas of physical, personnel, administrative, electronic, communications, and computational security. No one protective measure can suffice to meet the needs of an enterprise, so it is important that the protective measures be integrated to meet the widest range of requirements within company resource limitations. Examples of protective mechanisms include locks, curtains, computer access controls, and cryptography.

Detection is the term in which the processes and methodologies associated with discovering potentially damaging situations. Detection works on a timeline and every activity on that timeline triggers the occurrence of one or more additional activities. The amount of elapsed time between activities is critical, but shortening that time requires additional resources. Figure 4-1 shows an abstraction of the detection timeline. Note that detection is closer to the middle of the timeline than expected. This is a crucial point: there may be a significant amount of time between the start of an event and the detection of phenomenology that indicates an event has occurred (or may still be underway).

Figure 4-1 The Detection Timeline



Source: Ryan, J.J.C.H., (summer, 2018). Detection Timeline.

Detection is the default status for fully operational systems. Once the system is designed, developed, and implemented, all the security controls are in place, hopefully operating correctly and consistently, and the security team is in place, on the alert for a negative incident to happen. This is detection. Once a negative

incident happens, then the status transitions to reaction and correction, and eventually back to protection, and once more to detection. How fast the team can transition between phases is a function of training, experience, capabilities, and the severity of the problem(s).

A design consideration for detecting problems is the detection mechanism must be in place before the event occurs. In some cases, it is possible to detect problems without have specific detective mechanisms (like reading corporate secrets on the front page of the newspaper), but in most cases that simply is not true. If you are not collecting real time data or you do not have intrusion detection systems in place, you will not be aware that a problem is occurring in near real time nor will you be able to reconstruct the forensics evidence for evaluation. It does no good to install an alarm on a door after the door has been forced open and the Crown Jewels stolen.

There are *five categories of problems* that require detection. The first is the occurrence of problems against the Protected Class. The second category is the occurrence of problems against the Unprotected Class. The third category is the occurrence of problems on the Unknown Class. The fourth category is the misuse or abuse of privilege by insiders: The Insider Class. Finally, the fifth category of problem is the use of activities, methods, or technologies designed to reduce the probability of detection: The Counter-Detection Class. Each of these classes requires different approaches to detection, which is why consider each individually.

When you are designing a security architecture, scarce resources will require that you make compromises. Some problems can be protected against with a reasonable expenditure of resources, some cannot. Some problems may require extremely expensive protective mechanisms; other problems may have such a low likelihood of occurrence such problems are not addressed at all. *This choice creates the distinction between Protected Class detection challenges and Unprotected Class detection challenges.*

Protected Class

The Protected Class detection challenges fall into two subclasses. The first is *failure of the protective mechanism*: detecting the occurrence of problems even though the protective mechanisms are in place. This is significant; if one invests time and resources to address potential problems, those problems are clearly a priority (otherwise, why bother protecting against them?). If the problems occurs anyway, it means that the protective mechanisms have failed in some manner. Problem detection both alerts one to the fact that security has been compromised and the need to review the protections in place. The second subclass is *degradation of the protective mechanism*. Simply stated, protective mechanisms must be continually assessed to ensure they are correctly implemented, working correctly, and remain uncompromised. Locks rust, software updates can subvert exploit counter-measures, and people can get lazy. Ensuring protection functionality is critical.

The Unprotected Class

The Unprotected Class detection challenge is that the first line of defense against the problems that you chose not to protect against is, in fact, detecting the occurrence of the problem. The possibility of a data

breach was then but intentionally ignored. Placing detective mechanisms in place to trigger a reaction to the problem is the only thing you can do to minimize potential damage.

Unknown Class

The final three classes are more complicated. For the Unknown Class, you do not know what you are seeking. So simply monitoring your environment for operational aberrations becomes your first detection method. It seems simple, but one must understand they are looking for unknown or newly conceived scenarios.

For example, consider a spike in help desk phone calls: one management reaction might be to shrug and think, "must be a glitch in the system," while a different reaction might be to alert and start analyzing what might be going on. Mindset makes all the difference in detecting the Unknown Class of problems.

Insider Class

This is true for the Insider Class as well: insiders are, by definition, trusted to operate within the enterprise. There are varying levels of trust, but each insider has at least some level of trust. Detecting abuse or misuse of that trust requires the same sort of awareness, proactive analysis, and operational surveillance that the Unknown Class requires, but with a difference; the detection activities should be engineered to be able to establish agency of purpose.

Counter – Detection Class

Finally, for the Counter-Detection Class, the focus of the detective efforts is to determine when protections and detective mechanisms are being subverted. The subversion can occur in many ways. In a way, this class of detection effort combines all the aspects of the other four problems. To tackle this problem, the use of red teams periodically testing the detective mechanisms is useful, as is constant analysis of detective mechanism performance.

The **Reaction/Correction** phase consists of the set of activities required to continue operating while executing the activities needed to fully recover operational capabilities. There can be significant overlap between detection activities and reaction/correction activities, particularly for complicated problem sets, so best practice is to have each activity have dedicated people, resources, and management structures. Reaction/correction activities include (at the very least) the following efforts: investigations, forensics analyses, business continuity, crisis communications, and business recovery.

To summarize, there are three engineering phases for security architectures:

- ❑ Protection
- ❑ Detection
- ❑ Reaction/Correction

Risk

All decisions are risk-based decisions, even if one is not actively considering risk when deciding. When walking down the street at 2 A.M., one makes a decision-based risk assessment associated with the activity. When a decision to purchase a certain type of computer system, one considers several types of risk, including the risk that the company may go out of business and leave the company with no support. Risk minimization is essential.

There are two elements to risk decisions: the assessment of risk in the current time state (Now Risk) and an assessment of potential changes to the risk elements over time (Future Risk). As an entity requiring evaluation, one can think of risk as a combination of the probability that a negative incident will happen and the impact that such a terrible thing would have. For example, the risk of your secrets being exposed is analyzed by determining the probability that implemented protective mechanisms could be defeated plus the impact of data exposure. Note that this use of the term risk is slightly different than as it is used in common parlance. Typically, the word risk is used informally as a substitute for probability without including the impact assessment. However, when assessing risk associated with a potential decision, including the impact assessment gives you the additional data needed to make the subsequent management decisions on where to expend resources to reduce risk.

Now Risk is an evaluation of the elements of risk using the information that exists currently. In the information security community, it is hard to impossible to get actual data on the probability of terrible things happening. So, it is common to perform the analysis by estimating the chances of terrible things happening by investigating the elements of Threats, Vulnerabilities, and Counter-Measures. Threats exploit vulnerabilities to do terrible things. Vulnerabilities are exploitable by threats. Counter-measures reduce the ability of threats to exploit vulnerabilities successfully.

Threats come in two varieties: natural and man-made. Natural threats include things like fires, hurricanes, floods, earthquakes, and the odd meteorite. Yes, security planning does need to address these elements. For example, if you are operating in a flood zone, you should make the risk decision to put your server farm on a floor that is above the flood plain. If you live in an earthquake-prone area, like Japan, you might want to invest in earthquake mitigation technologies. Man-made threats are problems that originate in the mind of humans. They can be realized though the actions of people or through the agents of people, including animals, robots, and software.

When considering the elements that constitute viable threats, there are some aspects we can tease out. A credible threat requires both capability and intent. If the potential threat has capability but not intent, we call that a trusted insider. If the potential threat has intent but no capability, there is very little the threat can accomplish. Intent can be either organic to the individual or programmed into the threat agent (through software, training, or design). Further, capability requires several elements. To be truly capable, a threat requires three traits:

- ❑ the knowledge, skills, and ability needed to act;
- ❑ the resources to plan, develop, and execute actions;
- ❑ and access to the target.

Understanding those elements informs as to how to design counter-measures; what can be done to reduce or eliminate any or all the components? For example, what can be done to reduce the probability of a trusted insider developing the intent to act maliciously? How to reduce the probability that a threat can acquire the resources needed to act? When considering these elements, it quickly becomes obvious that there are easy and difficult things that can be implemented. A very summary of what can be done to counter risk activities are listed here:

Intent:

- ☐ Motivate threat to not form intent
- ☐ Intimidate threat through implication of personal harm or danger
- ☐ Scare threat through implication of harm to reputation, livelihood, etc.
- ☐ Discourage action through psychological motivations

Resources: Not much.

Knowledge, Skills, and Ability:

- ❑ Limit knowledge of security details --keep the details and engineering data secret; keep plans and procedures secret
- ❑ Use technologies, tools, and equipment that require highly specialized skills or training
- ❑ Combine security elements in ways that limit the probability that a threat could easily develop the needed KSA
- ❑ Pay for the development of unique systems that are not commercial off the shelf

Access:

- ☐ Deny access through barriers
- ☐ Deny access by use of technology such as locks
- ☐ Deny access through checkpoints
- ☐ Control access through I&A procedures
- ☐ Minimize access available to all people
- ☐ Limit speed of access
- ☐ Limit speed of egress

Vulnerabilities, similarly, can be considered to come in two types: accidental and by design. Accidental vulnerabilities are the result of lack of understanding, sloppiness, or unintended consequences. The danger here is you do not know that they exist until the analysis is completed. Efforts must include a search for such problems. One method used by large corporations is to search for vulnerabilities by having "bug bounties". This may work well for some organizations but may be absolutely the wrong thing for other organizations. Vulnerabilities that exist by design, on the other hand, should be the result of risk-based decision making. These include decisions to connect to vulnerable networks to conduct business or using a less than optimal engineering solution because a better one is not affordable. These types of vulnerabilities must be monitored, and detective measures should be considered to alert on any attempt of a threat to exploit these vulnerabilities.

Taking the threat, vulnerabilities, and countermeasures into account gives one insight into possible attack probability. Considering the impact of what the effects of an attack might be provides a way to measure the relative importance of each potential. Decisions can then be made about what to do, or not do, regarding each aspect of risk in the environment.

Now Risk decisions are based on knowledge of what is known regarding the current and past. **Future Risk** analysis looks at the potential for change in the situation. Each element of a risk decision has a change potential that varies over time. To do a competent analysis of what may change that can cause a decision's effect to be moderated, it is necessary to do a futures analysis. For example, a decision to plagiarize a dissertation made in 1987 might have been a fully logical decision, while the impact of detection would have been very high, the probability of the plagiarism being detected was relatively low. Fast forward to the Internet age, where plagiarism checking is both automated and crowd-sourced, and the situation has changed tremendously. The decision that seemed logical in 1987 has been turned on its head because of the unforeseen technological evolution.

In considering the impact future decisions made now, one should take into consideration how difficult it would be to change a decision should it become necessary or desirable to do so. Some decisions simply cannot be changed, these are point decisions. The example of the plagiarized dissertation is a point decision. Once it is published, there is no taking it back. Other decisions can be changed, but at a cost. Some costs are manageable. If one hires a spy, they can be fired and then a cleanup. Some damage will be incurred, but the decision can be changed. Some decisions more difficult to change due to costs. For example, making the decision to standardize on a software suite for accounting and fiscal management has long term implications that can be pervasive. A more dramatic example of a long-term risk decision that can have substantial future risk variability might be the decision to fund a new aircraft carrier: a dramatic change in technology could make aircraft carriers obsolete.

To summarize, risk decisions are made now for future events and so all potential futures should be considered while deciding. Over the life of a decision, future events should be monitored to see what is emerging and what potential impacts of change might be.

Systems Engineering an Information Security Solution

Using the ten pieces to the puzzle presented above, one now has the tools to think through an architectural approach to developing and managing an information systems security solution. To review, the ten pieces of the puzzle are:

- ❑The three policy questions
- ❑The three security attributes
- ❑The three security phases
- ❑The risk decision elements

Building on that, one can derive security requirements for the enterprise and then systematically identify a set of processes, technologies, and engineering approaches to address them. All of these are done within a risk management envelope: all decisions are risk decisions.

Identifying Security Requirements for an Enterprise

Understanding what the security requirements are for an enterprise is the first step in developing and managing security solutions for that enterprise. One cannot possibly meet all requirements; that would take more resources and have more operational impact than can be tolerated. However, knowing which requirements are not being met gives one the ability to monitor the unmet needs. Putting together a solution is an exercise in engineering a system, or a system of systems. Further, every solution is time limited, so the system must be periodically re-evaluated, and the set of solutions revised, updated, and improved. This is not a point solution effort; it is a life-cycle effort.

There are **three basic types of requirements** in systems engineering. These are referred to as *explicit, implicit, and derived requirements*. These requirements are identified for the information security of a system by systematically considering the three policy questions against the three security attributes. By going through this exercise, one can concretely describe what is required to protect the information in an enterprise. Table 4-1, a 3 x 3 matrix, specifies what is needed in terms of policy questions and security attributes. In this table, the policy questions are abbreviated to make it easier to read. But remember that it is important to consider the full meaning of the questions for each cell in the matrix. Cells are renumbered to make it easier to refer to in the discussion about requirements. Refer to this table to explore examples of security requirements for a system.

Table 4-1

	What?	**How Much?**	**How Long?**
Confidentiality	C-1	C-2	C-3
Integrity	I-1	I-2	I-3
Availability	A-1	A-2	A-3

When each cell is considered, requirements are identified. For example, in cell C-1, the full policy question consideration is:

> *What data, systems, and operational elements need to have aspects of confidentiality protected? What elements do not need confidentiality protection?*

Then, as one proceeds through C-2 and C-3, one can elaborate on how much (or how little) confidentiality protections are needed and how long they need to be kept in place (or when they can be allowed to lapse). While this is a systematic and straightforward process. There are many complexities in systems, addressed in previous examples, so a strong suggestion is that this exercise be conducted by a diverse team with multi-faceted knowledge of the system under analysis.

Explicit Requirements

The first requirements that will be identified will be the e*xplicit* requirements. These are the ones that are defined as things that are needed in a system, independent of any implementation or technology solution. For example, an explicit requirement that may be identified in cell C-1 could be that the existence of a sensor must be kept secret. That is an obvious type of requirement to specify, as it reflects a specific secrecy need.

Implicit Requirements

For all explicit requirements, *implicit* requirements also exist. These are the requirements that are implied by the need that is identified in the explicit requirement. What is implied by a need to keep the existence of a sensor secret? There are many implied requirements to consider. First and foremost, the sensor must be hidden from observation. Observation may be accomplished though vision, imaging, signals interception, and other means. It is important to then complete the analysis of the types of observation that the sensor might be subjected to. Another implied requirement is that all individuals who observe or work with the sensor or the sensor products must be approved to do so. Yet another implied requirement is that the sensor products themselves be protected.

In the matter of moments, three implied requirements have been created that come from the single explicit requirement; more can develop as required. This is the power of working through these analyses; by extrapolating what it would take to meet an explicit requirement, we identify the systemic elements that also need to be considered. Nothing lives in an isolated space; surrounding system components must be considered.

Derived Requirements

Beyond explicit and implicit requirements, there are *derived* requirements. Derived requirements are the things that are necessary conditions for the environment. These, unfortunately, are sometimes simply assumed in a system. That can lead to hilarious outcomes, such as delivering a fully-compliant system to a customer that is totally unusable because it lacks a user interface.

Derived requirements are important to consider; it is usually best to start by reviewing the implicit requirements. One of the implicit requirements for the secret sensor system was that all individuals who observe or work with the sensor or the sensor products must be authorized. There are many requirements that can be derived from this single implicit requirement. One is that a system exists to vet individuals for trustworthiness. Another is that only vetted individuals are authorized to work or with the sensor. Another is that there are controls that check to make sure that only vetted and authorized individuals get access to the sensor and to the area in which the sensor is housed.

There is power in this approach to identifying explicit requirements and then identifying the underlying implicit and derived requirements. Going through the exercise of thinking through all these things can not only help one understand the management needs of the environment, it can also point to possibilities for efficiencies in operational processes. At the very least, it provides one with a comprehensive understanding of what the security needs are for a system, a system of systems, or an enterprise.

Security Solutions Consideration

Once a set of requirements is created, one can start to parse out how to develop and implement solutions for each requirement. Note well: no enterprise can afford all potential solutions, so this will start out as a list of options, which would then need to be winnowed down based on enterprise priorities, operational realities, and resource availability.

Table 4-2 presents a very simplified view of starting the process. Simplified because information exists in many divergent phases. A trivial way of thinking about the information phases include processing, storage, and transmission. That is a normal abstraction, but it overlooks some very specialized states that may warrant separate consideration according to enterprise needs. A more layered approach to information states is that of input, output, processing, local communications, external communications, temporary storage, permanent storage, and display. In any event, when considering how to select security solutions, it is important to consider which states the solution works for and which ones it fails.

Table 4-2 Information Security Parameters and Process

	Protection	Detection	Reaction/Correction
Confidentiality	C-P	C-D	C-RC
Integrity	I-P	I-D	I-RC
Availability	A-P	A-D	A-RC

Look at an example of how this table can be used to explore security solutions. For cell C-P, assume that requirements have been identified for protecting the confidentiality of information in the display state. Some technologies that can contribute to meeting this requirement include putting curtains on the windows and restricting access to the area. But these solutions would be useless if the operational environment were an economy seat on a commercial flight. In this case, other solutions would be considered. The crucial point? The solutions need to match the operational environment as well as meet the statement of requirement.

UAS Security Challenges

Now with a basic understanding of how to think about the information security challenges in an operational enterprise, consider what that mean for UASs. Again, any specific security solution will need to be unique to the enterprise; solutions appropriate for commercial delivery drones may not be sufficient for national security operations of a sensitive nature. Some elements that may require security consideration; communications, data processing systems, sensors, location, and control systems.

Communications may need to have confidentiality, integrity, and availability protected. Data processing systems may have high integrity needs as well as some availability needs. Sensors may need high confidentiality and integrity requirements. Location may be sensitive based on mission and so may have contextual security needs. Control systems may need to have strong integrity and availability needs.

Discussion Questions

Test your understanding of the material presented in this chapter by thinking through the following questions. Discuss them with other people. There are some subtleties that are fascinating to explore.

1. Explore the differences and similarities of confidentiality and privacy. How do the concepts overlap? How are they distinctly different? What does that mean in terms of managing privacy?
2. How is the concept of integrity different from truthfulness?
3. How is the concept of ownership different from availability?
4. How does custodial responsibility for information translate into security requirements? What are some of the explicit, implicit, and derived requirements?
5. What security requirements are needed during the acquisition of sensitive equipment, material, or information? What are some of the explicit, implicit, and derived requirements?
6. What security requirements are needed for the management of systems, particularly information processing systems? Think about installing patches or software updates: what security controls are needed during these processes?

Bibliography

Ryan, J. (2018, August 26). *Dr. Julie J.C.H Ryan Research Page* . Retrieved from GWU SEAS: https://www2.seas.gwu.edu/~jjchryan/research.html

References

"Nichols, Randall K, Daniel J. Ryan, and Julie J.C.H. Ryan, *Defending Your Digital Assets Against Hackers, Crackers, Spies, and Thieves,* New York: McGraw Hill, ISBN: 978-0072122855, 2000" (Ryan, 2018)

"Ryan, Julie J.C.H., *Teaching Information Security to Engineering Managers*, Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference, Boulder, Colorado, November 2003" (Ryan, 2018)

U.S. Department of Defense (1970). "Security Controls for Computer Systems (U): A Report of the Defense Science Board Task Force on Computer Security". Rand Corporation: Santa Monica, California. Available online at https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/ware70.pdf

"Pettigrew, J. Andrew, and Julie Ryan, Making Successful Security Decisions: A Qualitative Evaluation, IEEE Security Privacy Magazine, Vol 10 (1), 2012" (Ryan, 2018)

"Amin, Rohan, Julie Ryan, and Johan Van Dorp, Detecting Targeted Malicious Email Using Persistent Threat and Recipient Oriented Features, IEEE Security Privacy Magazine, 2012, vol 10n3 (64-71)" (Ryan, 2018)"

"Ryan, Julie J.C.H., Thomas A. Mazzuchi, Daniel J. Ryan, Juliana Lopez de la Cruz, Roger Cooke, Quantifying Information Security Risks Using Expert Judgment Elicitation, Computers and Operations Research, April 2012, v39 n4 (774-784)" (Ryan, 2018)"

"Ryan, Julie J.C.H. and Daniel J. Ryan, Performance Metrics for Information Security Risk Management, IEEE Security and Privacy, vol. 6 no. 5, Sep/Oct 2008, pp. 38-44" (Ryan, 2018)

"Ryan, Julie J.C.H., Use of Information Sharing Between Government and Industry as a Weapon, Journal of Information Warfare, Journal of Information Warfare" (Ryan, 2018)

"Ryan, Julie J.C.H., Cyber Security: The Mess We're In and Why It's Going to Get Worse, The Cyber Security Policy and Research Institute, Report GW-CSPRI-2100-4, April 11, 2011, 2010-2011 Seminar Papers" (Ryan, 2018)

"Ryan, Julie J.C.H. (ed), Leading Issues in Information Warfare and Security Research, Reading, UK: Academic Publishing International, Ltd, ISBN : 978-1-908272-08-9, 2011" (Ryan, 2018)

"Ryan, Julie J.C.H, Information Warfare: A Conceptual Framework, Seminar on Intelligence, Command, and Control (1996), ISBN : 1-879716-39-9,Proceedings of the 1996 Seminar on Intelligence, Command, and Control. Boston, MA: Harvard University Press, 1997" (Ryan, 2018)"

"Ryan, Daniel J. and Julie J.C.H. Ryan, Protecting the NII, Information Warfare: Protecting Your Personal Security in the Electronic Age, ISBN : 978-1560251323, (New York: Thunder's Mouth Press, 1994), 626" (Ryan, 2018)

Websites of Interest

https://www2.seas.gwu.edu/~jjchryan/research.html
https://www2.seas.gwu.edu/~jjchryan/curricularmaterials.html
https://www2.gwu.edu/~usjpciip/