Chapter 10
Protecting Information with Cybersecurity

10.1 The Cybersecurity Challenge

This chapter provides a brief, high-level summary of a very large and complicated topic: protecting information-intensive systems against attack, compromise, corruption, theft, unauthorized use, and other malicious acts. The overall term for this is cybersecurity, which was referred to in the past as Information Assurance.[1] Scarcely a day passes without a major news report on computer crime or other incident, from things as basic as defacing Web sites to identity theft, crippling virus attacks, diverted bank accounts, ransomware, and compromise of sensitive operational data and intellectual property. Systems with large numbers of users, geographically distributed locations, and networked access are especially vulnerable. Security professionals, many of whom have specialized training and hold credentials like Certified Information Systems Security Professional (CISSP), fight a never-ending battle to defeat criminals, hackers, terrorists, foreign intelligence services, and deranged people who derive perverse satisfaction from releasing viruses, Trojans, worms, and other "malware." Even a system with robust safeguards against external attack may be vulnerable to the "insider threat" from personnel with authorized access who have become disaffected, have taken money from a criminal or hostile agency, or are simply poorly trained and careless. Any organization or enterprise that relies on information processing is a potential target, and the reality is that most systems of any size or significance have already been penetrated. Figure 10.1 suggests the range of threats confronting secure systems.

Appendix G tabulates some common attack methods and possible mitigations. A few examples from recent reports on cyberattacks and data breaches will serve to highlight the challenge [1].

[1] We will not use the obvious CS acronym since this is so widely taken to mean computer science.

J. M. Borky, T. H. Bradley, *Effective Model-Based Systems Engineering*,
https://doi.org/10.1007/978-3-319-95669-5_10

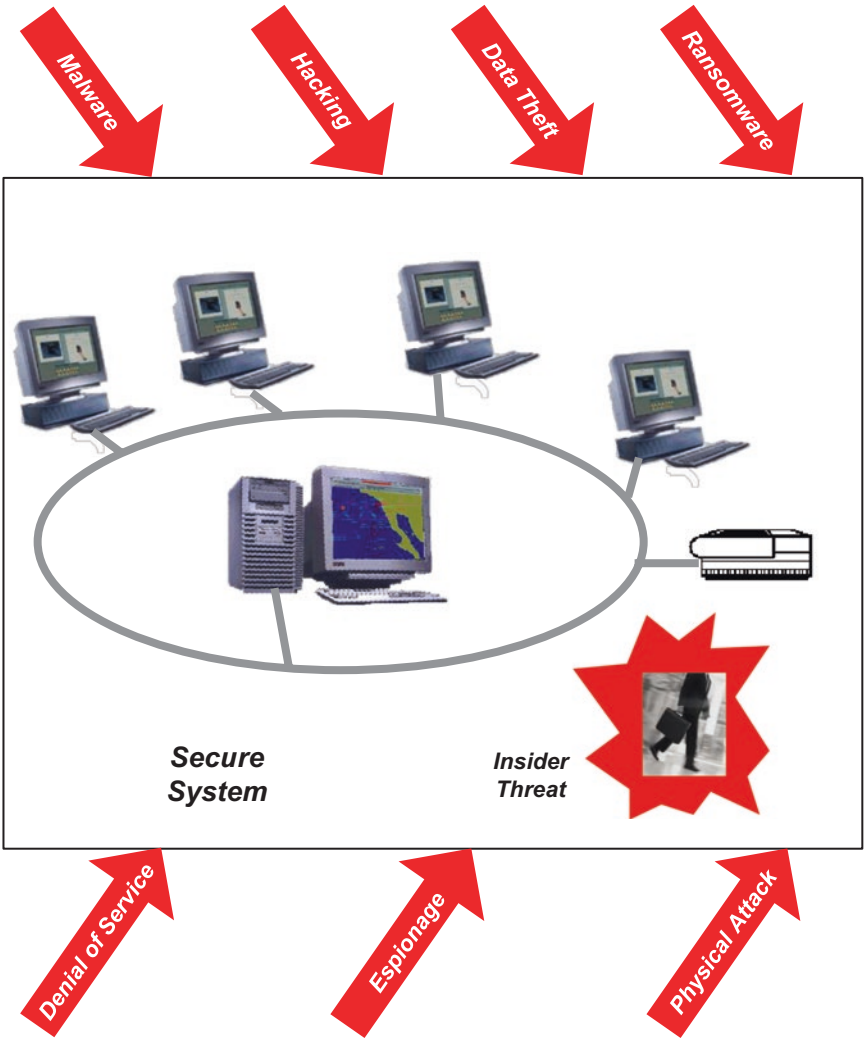


Fig. 10.1 Multiple threats to secure systems

- Common estimates are that the number of new threats is doubling every year, and the global cost of data breaches is measured in trillions of dollars per year.
- Many, perhaps most, common attacks could be defeated with available and affordable safeguards that victims simply fail to deploy; as an example, the average time from publication of a software patch or other mitigation to close a newly discovered vulnerability to the actual installation of the protective measure by both private and public sector organizations is months or even years, when it should be hours or days.

- The top ten system vulnerabilities account for approximately 85% of data breaches; some have been around for as long as several years but continue to be exploited.
- Roughly 80% of successful attacks originate with external threat agents, but the majority also involve either deliberate or accidental actions by "insiders," i.e., members or employees of the victimized organization; a common example is an attack that starts with "phishing" to trick an insider into revealing information or downloading malware, giving the attacker access to the system.
- A cybersecurity provider recently reported that ten out of ten mobile devices and applications used for payments at stores, theaters, hotels, etc. had major security weaknesses such as failing to encrypt Social Security Numbers.
- Relatively new and more sophisticated threats are proliferating; one such is a "ransomware" attack in which a criminal attacker encrypts a victim's databases and demands payment to restore them.
- Online security across the board is so bad that most successful hackers don't actually need to employ hacking methods; they simply enter the victim's network through unprotected public interfaces.

Just as a perfectly safe airplane is one with so much strength and redundancy that it's too heavy to fly, a perfectly secure information system would simply deny access to anyone. Cybersecurity is therefore a balancing act involving an adequate level of protection against known or postulated threats while still allowing systems and their users to carry out their legitimate functions and accomplish their business objectives or operational missions. For example, a classic dilemma facing a security architect today is the unanticipated user, an external party who is not known to a system a priori, and may very well not even be physically located at the system, but who nevertheless has permissions granted by a higher authority to access functions and data. The security design must then ensure that such legitimate users are granted access, while imposters are blocked with acceptably low error rates.

Since this is a book about system architecture and architecture-centric systems engineering, the emphasis of this chapter is on ways to implement adequate safeguards against cyberattacks within the context of an overall balanced solution to customer needs. The discussion is primarily about architectural approaches, especially layered defense design patterns for security, as well as selection of security controls (safeguards or countermeasures) and the products that implement them. We necessarily pay less attention to procedural aspects of security such as enforcing strong passwords and providing security awareness training, although these are absolutely essential elements of an effective cybersecurity posture.

Given that absolute security is impossible, the approach taken today by security policy-makers and architects is based on risk management. A typical security architecture begins with the most thorough analysis possible of the sources and methods of system attack. The security architect then follows an orderly process of selecting and applying protective measures while ensuring that a system can still accomplish its purpose with acceptable impacts on performance, cost, reliability, and long-term evolution. Security design often uses products with defined assurance levels and

employs design patterns, security protocols, and standards to achieve an acceptable level of security risk. This chapter describes a typical security architecture process and introduces some terminology and design approaches. The objective is to enable a general system architect to appreciate the dimensions of the security challenge, to interact more effectively with security specialists, and to ensure compliance with security policies. There are frequent references to Department of Defense (DoD) security policies and practices both because of the importance of protecting classified information and because much of the security technology and many standards and implementation methods, which are used throughout information technology (IT) systems, derive from the experience and investments of the Defense community.

10.2 Basic Concepts

10.2.1 Elements of an Attack

Increasingly sophisticated attacks on information systems are often characterized as cybercrime, cyber-espionage, and even cyberwarfare. Figure 10.2 is a graphical portrayal of the main elements of a cyberattack and introduces some standard terms. An *Asset* is any information, process, or other system content that requires protection. Assets are potentially at risk if the system has *Vulnerabilities* that an attacker can exploit. Collectively, these vulnerabilities constitute the system's *Attack Surface*, and a primary goal of secure architecture is to minimize this. An asset such as a database typically has *Attributes* that are the specific items sought by an attacker. A *Threat* is created by a *Threat Agent* who seeks to exploit a vulnerability to steal, corrupt, delete, or otherwise harm an asset. Then if the system has safeguards, properly called *Security Controls*, that adequately mitigate the vulnerability, the asset is protected; otherwise, there can be an *Exposure* or data breach. The structure in Fig. 10.2 is the basis for the Vocabulary for Event Reporting and Incident Sharing (VERIS) [2] which asks "What Threat Actor took what Action on what Asset to compromise what Attribute," referred to as the 4 As, and is widely used in reporting and compiling information on cybersecurity events.

Published work on information security, of which a paper by Whitmore [3] is a typical example, generally highlights the following as the principal factors in the success or failure of a system that is trusted to handle sensitive information:

- Clarity and completeness of requirements, including validation by stakeholders, especially the ultimate system users
- Trustworthiness of the components and policies used in system implementation
- Satisfaction of requirements by the system design and implementation
- Correct operation and maintenance of the system to preserve security characteristics

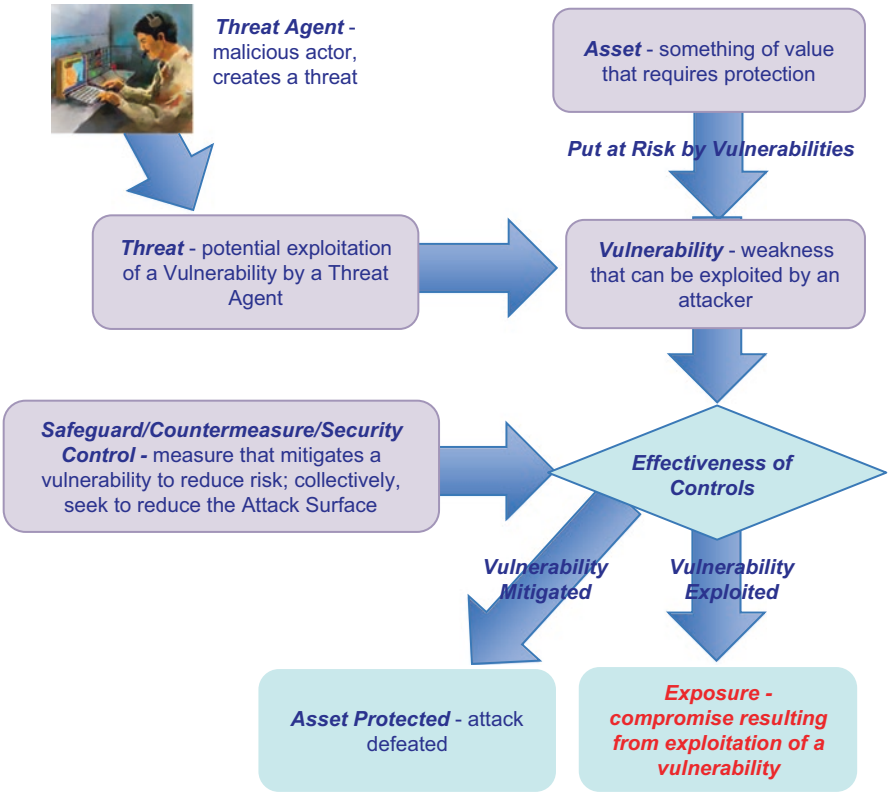


Fig. 10.2 Elements of a cyberattack

- Understanding of the environment in which the system operates, including the threats that security mechanisms must counter

The remainder of this chapter explores these aspects of system security and discusses some of the current approaches in use by security architects and engineers.

Although classified information is most often associated with military systems, any organization that has sensitive data needing protection should apply a consistent scheme for sorting that information into levels of importance or sensitivity and clearly labeling it so that appropriate safeguards can be applied. The US DoD defines classification levels, such as For Official Use Only (FOUO, also called Controlled Unclassified Information), Confidential, Secret, and Top Secret, primarily on the basis of the damage to national security that would result from information compromise. Intelligence agencies typically require more stringent control and protection for their information, which is often achieved through "compartments" that have very strict access requirements and higher levels of physical, administrative, and technical security measures. Civil agencies and commercial enterprises typically define sensitivity levels, e.g., Restricted, Confidential, Private, etc., for

information such as intellectual property, personal and financial data, strategic planning, and compliance with laws governing information protection. A good data classification system concentrates the most stringent security controls on the most sensitive information, especially when it is impractical to give everything the highest level of protection.

10.2.2 Categories of Threat Agents

The threat agents in Fig. 10.2 come in a wide variety of forms. The following is a representative tabulation.

- *Insiders* – people within the targeted organization who may be either malicious (deliberately seeking to do damage, commit theft, etc.) or inadvertent (careless, poorly trained, etc.); these are the most dangerous because they are already inside system defenses and have access to targeted assets.
- *Hackers, Thrill Seekers, and Individual Criminals* – individuals or small groups whose motivation may range from ideology to financial gain to an adrenaline rush from cracking a system.
- *Organized Crime* – organizations looking to compromise systems for purposes of theft, blackmail, data ransom, or other criminal objectives; stolen data is commonly traded on the Dark or Black Web, a group of clandestine peer-to-peer networks ("darknets") using the public Internet but with measures to control access and prevent users from being identified or traced.
- *Terrorists* – a variety of criminal organization that, in addition to cybercrime, may seek to compromise target systems as part of a political, ideological, or simply psychopathological campaign.
- *Advanced Persistent Threat (APT)* – the most sophisticated category of attackers, often state-sponsored for purposes of military or commercial espionage; APTs commonly have extensive financial and technical resources to execute elaborate campaigns extending over long periods, employing mixtures of tactics, and seeking to thoroughly infiltrate, and even take control of, target systems.

10.2.3 Fundamental Security Concepts

System, network, and enterprise security is customarily discussed in terms of a set of fundamental concepts. These include:

- *Confidentiality* – the ability to preserve secrecy by preventing unauthorized access to sensitive data, whether stored ("at rest"), being communicated ("in transit"), being processed ("in use"), or being created (e.g., by being intercepted by an attacker's tool such as a keyboard sniffer)

- *Integrity* – the ability to preserve content by preventing an unauthorized party from modifying, corrupting, inserting, deleting, or duplicating data
- *Availability* – the ability to ensure timely access to protected data and functions by authorized recipients
- *Authentication/Identity* – the ability to prove with adequate certainty that a party is who the party claims to be and has an identity that is known to the system or enterprise for purposes such as Access Control
- *Authorization/Access Control* – the ability to restrict access to sensitive data to authenticated recipients who possess the necessary access permissions and need-to-know
- *Non-Repudiation* – the ability to prove that a given party took part in an information transaction despite that party's attempts to deny involvement
- *Audit* – the ability to detect, inspect, record, analyze, and report events associated with security mechanisms, which may be important in determining that a compromise or other unauthorized action has occurred, verifying that security procedures have been followed, and detecting events or trends that may reveal hostile activity

Confidentiality, Integrity, and Availability are the primary cybersecurity concerns and are, with a certain amount of irony, referred to as CIA. An additional term and concept that is increasingly stressed is *Resilience*, which is the overarching characteristic of a secure system that implements all of the listed concepts, is able to defeat both existing and emerging threats, can maintain essential functions during and after an attack, and can maintain an acceptable level of risk as the threat environment evolves.

10.2.4 Elements of Resilient Cybersecurity

To achieve a cyber-resilient system architecture and implementation while maintaining acceptable system performance, cost, and reliability, the security architect employs proven designs, mechanisms, products, and procedures. All too often, organizations and managers assume that cybersecurity begins and ends with technical safeguards such as firewalls. In reality, an effective security solution requires all three of the elements sketched in Fig. 10.3. They include:

- Personnel who are trained, motivated, and empowered to perform their duties in a secure and reliable fashion, including all levels of management, system users, and system support staff
- Processes and procedures that implement a security policy and maintain the effectiveness of safeguards
- Technologies that implement security controls and that evolve to keep pace with an ever-changing threat environment

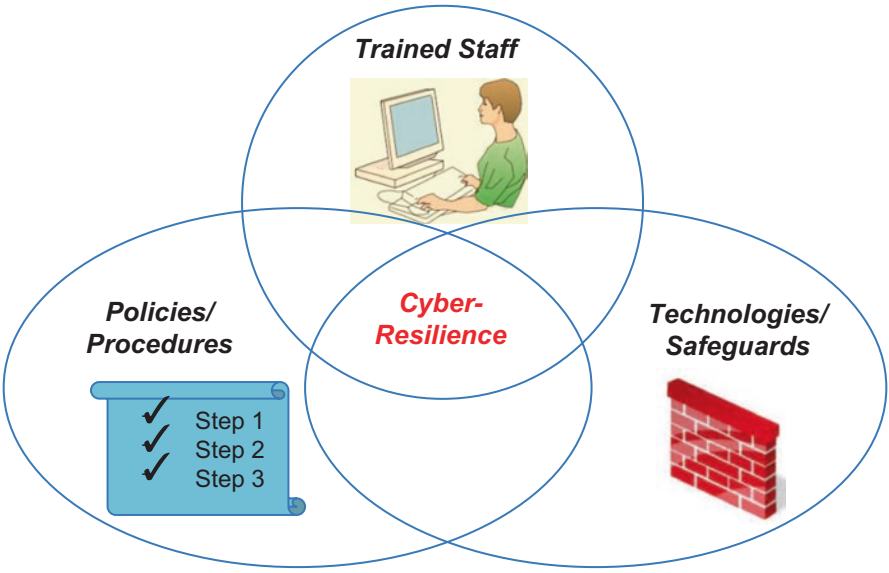


Fig. 10.3 Essential elements of an effective cybersecurity solution

A second threefold taxonomy is useful in describing the security controls or safeguards employed to protect a system. Securing a system that has human involvement normally requires a mix of technical, physical, and procedural security measures.

- *Technical measures* – these can be quite diverse and may include:
  - Firewalls and other protective devices at the system boundary to protect against unauthorized information flows in or out of the system
  - Intrusion detection devices to detect, log, and analyze unauthorized attempts to access a system, often resulting in alarms so that timely defensive measures can be taken
  - Public Key Infrastructure (PKI)
  - Auditing of activity logs, system configuration changes, and other events to detect and respond to suspicious or prohibited actions
  - Security testing, including simulated hostile attacks, to verify the robustness of information protection and identify deficiencies that must be corrected
  - Policy servers that automate the application of rules governing the operation of security mechanisms
  - Access control mechanisms to enforce user privileges and restrict unauthorized use of resources and data
  - Lockdown ("hardening") of servers and other computers by disabling any operating system functions, ports, utilities, or other capabilities that are not absolutely necessary and that may create security vulnerabilities

- *Physical measures* – these seek to place sensitive information or other protected resources behind barriers ranging from facility access controls such as badges, door locks, and guards to locked enclosures and backup power and air conditioning. These can range from property fences and controlled access points to alarms and locked equipment enclosures.
- *Procedural measures* – these deal with secure operations and practices aimed at maintaining the effectiveness of security controls and eliminating vulnerabilities caused by human error. A typical example is a requirement to erase sensitive software and data from a computer at the completion of a work period or task. Others include automatic locking of computers after a period of inactivity and regular, timely system scans to detect suspicious behavior or altered system configurations. Such procedures may be documented in an Automated Information System (AIS) directive. A very important category of procedural security involves training system users to practice good security "hygiene" such as frequently changing passwords and detecting and defeating "phishing" attacks that try to trick a user into disclosing sensitive information or loading malicious software.

Any protective measure is likely to eventually be compromised or circumvented. As a result, traditional defensive configurations that tend to be static and to protect only the system boundary are evolving toward more dynamic and sophisticated approaches. Layered defense approaches such as Defense-in-Depth (DiD) and Zero-Trust Architecture, described below, are an important advance over previous security implementations. The basic idea is that an attacker, to reach sensitive system content, must penetrate a series of barriers, which can be individually managed and updated to deal with newly detected threats and methods of attack. Another important cybersecurity trend involves anti-malware tools that go beyond detecting a threat based on comparing a message or file to a library of threat signatures and seek to analyze a message and any attached files to determine its functions, spot the presence of malicious code or Web sites, and block an attempted attack.

Yet another evolving security approach seeks to be proactive and is based on continuous, even real-time, monitoring of user behavior and cybersecurity events coupled with dynamic responses to minimize or contain attempted intrusions [3, 4]. This could involve deploying agents (see Chap. 14) at various locations within an information enterprise to measure events such as password change attempts (which may indicate a password breaking attack), failed log-on attempts, blocked data packets, or data objects failing integrity checks. Agent reports can alert operational personnel to an abnormal situation and can trigger automated responses such as blocking suspected network addresses. Logging and analysis of messages and other traffic, particularly amounts, combinations, or timing of data downloads that have not been previously seen, can reveal patterns that may indicate the early stages of a cyberattack. A number of commercial products based on inspection and sophisticated statistical analysis of network activity are now available and have proved effective. Continuous testing of security controls and application software is essential to detect and mitigate vulnerabilities, especially new ones from the ever-evolving threat environment.

10.2.5 Cybersecurity Domains

Security professionals typically organize the cybersecurity discipline using the domains of the CISSP Common Body of Knowledge (CBK). The following paragraphs briefly summarize these[2] and give the equivalent of an executive summary of the subject.

- *Access Control* – measures to ensure only authorized persons or other entities ("subjects") can get possession of protected content ("objects"). They are based on file permissions, program permissions, and data rights that are tailored to each individual system user's need for such content and embedded in a user account. Access control includes the process by which individuals receive authorizations to access particular system content and authentication of the identity of a subject requesting access to an object.
- *Telecommunications and Network Security* – measures to ensure cybersecurity when protected objects are transmitted. This includes network architecture and design, trusted network components, secure channels, and protection against network attacks.
- *Information Security Governance and Risk Management* – security policy, guidance, and direction that a system must follow to attain an Authority to Operate (ATO) with sensitive data. This commonly dictates security constraints and procedures that must be followed, security functions vs. operational or business goals and missions, policy compliance and enforcement, phases of the information life cycle, and governance of third parties such as component vendors, personnel security, education and training, metrics, and resources, especially budgets and skilled personnel.
- *Secure Software Development Life Cycle (SSDLC)* – building cybersecurity into system and application software from the outset. This includes determining information protection needs, establishing security requirements, developing secure software architecture and design, software security testing, and assessing protection effectiveness. Software security controls must be applied and tested in a development environment that is isolated from production systems.
- *Cryptography* – converting plaintext into unreadable ciphertext using complex algorithms. This domain includes tailored applications of cryptography, the life cycle of cryptographic materials such as encryption keys, basic and advanced cryptographic concepts, encryption algorithms ("ciphers"), public and private keys, digital signatures, non-repudiation (generally using digital signatures), attack methods, cryptography for network security, cryptography for secure application software, PKI, issues with digital certificates, and information hiding (what and when to hide). The protection of the crypto keys is more important than the strength of the cryptography itself; hence a good Key Management Plan means more than the strength of a cipher.

[2] Actually, this discussion uses the CBK Domain structure that precedes the most recent revision because it is much more logical and useful for tutorial purposes.

- *Security Architecture and Design* – embedded features and functions that implement security controls and eliminate or mitigate vulnerabilities. This includes models and concepts, model-based evaluation, security capabilities, vulnerabilities, countermeasures, selection of trusted components, and mechanisms to establish and maintain the required level of system security over time.
- *Security Operations* – ongoing activities to enforce policies and procedures, detect and prevent or mitigate attacks, and maintain a system security posture. This includes operations security, resource protection, incident response, attack prevention and response, management of software vulnerabilities and patches to mitigate them, change and configuration management, and procedures to preserve system security resilience and fault tolerance.
- *Business Continuity and Disaster Recovery Planning* – measures to minimize the organizational and mission impacts of cyberattacks and natural disasters. This includes requirements, impact analysis, backup and recovery strategies, disaster recovery, and testing of plans for continuity and recovery.
- *Legal, Regulations, Investigations, and Compliance* – legal issues, ethics, investigations and evidence, forensics, compliance procedures, contracts, and procurements.
- *Physical Security* – measures to establish and maintain a secure environment for sensitive resources and processes. This includes site and facility design, perimeter security, internal security, facility security, equipment security, privacy, and safety.

10.2.6 Cybersecurity Foundations

An effective, affordable, and resilient cybersecurity solution begins with two fundamental principles.

Security Policy and Requirements The first essential is a policy that concisely describes the goals assigned to the security function of an organization and how those goals will be met with available or planned resources, together with specific, verifiable security requirements. A good starting point is a Security System Concept of Operations (SECOPS), which complements an overall Concept of Operations (CONOPS) by providing a focused definition of security requirements levied on the components or functions of a system with rationale for each. For example, the SECOPS would describe the required level of access control and what the components responsible for user account management, privilege control, and user authentication must do. A SECOPS often includes "Abuse Cases," which are analogous to normal Use Cases except that they describe behaviors associated with a cyberattack.

The SECOPS may incorporate or be the basis for a Security Policy Document (SPD). A security policy should be stated in a form that is useful to security stakeholders and includes an overall security policy and governance strategy. The policy

is typically elaborated in a Security Plan (SP) that spells out specific policy for all system elements, defines an acceptable risk threshold, and gives an overview of system security requirements and the existing or planned controls to satisfy those requirements. Special Publication SP 800-18, *Guide for Developing Security Plans for Federal Information Systems*, Rev. 1, from the National Institute of Standards and Technology (NIST) [5] has general guidance applicable to security planning in both public and private sectors.

Ultimately, a secure system needs clear, unambiguous, effective, and verifiable requirements for security features and functions as part of an overall requirements baseline to support development, acquisition, integration, test, operational support, and other activities. Federal Information Processing System Publication 200 (FIPS 200) [6] lays out minimum acceptable requirements in 17 areas and can be used as a template for developing security requirements. Figure 10.4 suggests the flowdown from overall organizational policy and goals to focused security policy, a SECOPS, and ultimately security requirements and implementation guidance.

Secure System Life Cycle The best and least expensive cybersecurity solution results when security is "baked in" to a system design from the outset. A secure life cycle starts with making security requirements a core part of the overall system requirements baseline and continues through all phases of development, integration, test, production, fielding, and operational support. All too often, an attempt is made to graft cybersecurity safeguards onto an existing system, but this is virtually guaranteed to be much harder to do and to yield a less resilient and more costly result. A secure life cycle is particularly important for application software and is further discussed under that topic later in this chapter.

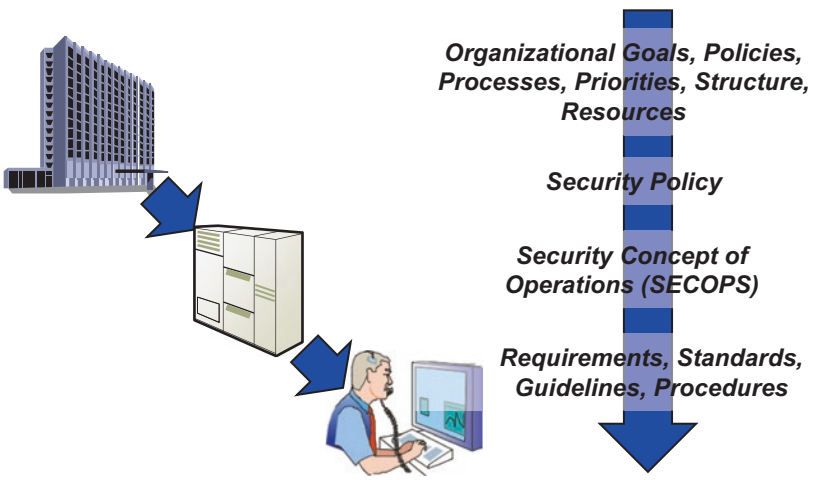


Fig. 10.4 Effective cybersecurity originates in overall organizational policy and flows down to security requirements and implementation

These principles may seem obvious but they are, in fact, widely ignored. A recent analysis of the cybersecurity of US Air Force systems uncovered two pervasive trends in risk management [7]:

- Failing to adequately capture the impact to operational missions, a violation of the first basic principle
- Implementing cybersecurity as an add-on to existing systems, versus designing it in, a violation of the second

The report goes on to recommend better definition of cybersecurity goals; clear roles and responsibilities with adequate authority to act; more comprehensive and better prioritized security controls across systems and enterprises; improved access to cybersecurity expertise; continuous assessment and reporting of security status; better threat data; and, perhaps most significantly, holding individuals responsible for violation of cybersecurity policies. None of this is new or surprising. Similar findings come up repeatedly in reviews of nominally secure systems in Government, industry, and elsewhere, especially after cyberattacks. Clearly, the security community has a long way to go in getting system owners to take the cyber threat seriously and to deal with it effectively.

It should be clear from this preliminary survey of cybersecurity that this is a multidimensional problem and a process that continues for the entire operational lifetime of a system or enterprise. Just a few of the matters requiring diligent attention and adequate resources are promptly installing software patches, managing user accounts, acting quickly to detect and mitigate new vulnerabilities, maintaining the security awareness and defensive training of personnel, validating the trust level of new system components before installation, and monitoring system resources and activities to detect and defeat hostile actions. A good basic agenda for establishing and maintaining cyber resilience is spelled out in the Critical Security Control (CSC) list maintained by the Council on CyberSecurity [8]. The remainder of this chapter explores the aspects of cybersecurity that are most prominent in system architecture and in MBSE.

10.3 Cybersecurity Risk Management

Risk management is a core activity of Systems Engineering, based on assessing the probability of occurrence of potential adverse events or conditions and the consequences of such occurrences. A risk management process usually consists of:

- Risk identification
- Risk assessment in terms of probability of occurrence and consequence of occurrence
- Identification of risks requiring mitigation
- Planning and budgeting for risk reduction activities

- Tracking the progress of risk mitigation until a risk is retired or a decision is made to accept it

In dealing with cybersecurity, risk management is focused on protected assets, threats, vulnerabilities, and risk mitigation using security controls. It begins with risk identification and analysis, which then supports selection of security controls to achieve an acceptable risk level. This, in turn, is the basis for securing approval to operate the system. Accordingly, risk analysis is an essential component of defining security policy and requirements. The goal is to apply security controls intelligently to mitigate the most important risks while preserving an operationally effective, affordable, and supportable system. Risk management continues over the life of a system to deal with changes to the system itself and with the emergence of new vulnerabilities and threats.

Cybersecurity risk management deals with four basic elements.

- *Risk* – a measure of the likelihood of occurrence of an adverse event such as loss of sensitive information and of the potential resulting consequences to a system or organization that is the target of an attack. This is conceptually equivalent to the probability and consequence of occurrence used in general risk management.
- *Threat* – the potential danger of a compromise that is associated with a vulnerability. The term is commonly associated with an entity (Threat Agent), event, or circumstance that has the potential to adversely affect an organization/target by exploiting a vulnerability.
- *Vulnerability* – a flaw in a target that can potentially be exploited by a threat. This is often due to the absence or weakness of a security control or countermeasure.
- *Exploitation* – an event in which a threat agent leverages a vulnerability to carry out an attack or otherwise induce adverse consequences for a target. An exploit typically results in an Exposure/Compromise/Data Breach, i.e., an instance of the target being exposed to loss.

The basic risk relationship can be expressed by considering an individual risk as a function of a specific vulnerability and a threat that arises from the possibility that the vulnerability may be discovered and exploited. We can state this simply as:

$$\text{Risk}_n = f\left(\text{Threat}_n, \text{Vulnerability}_n\right) \tag{10.1}$$

where the nth risk, Risk_n, results from the nth threat associated with the nth vulnerability.

Cybersecurity risks are associated with information assets (data, systems, processes, etc.) whose value is such that they require protection. The sum of vulnerabilities (or, equivalently, of potential attack vectors) constitutes an Attack Surface. Managing cyber risks then comes down to minimizing the Attack Surface. This is made difficult by the reality of a diverse, growing threat environment; a full listing of known vulnerabilities and attack methods would fill a small book, and a complete description would fill a large one.

The first step in cybersecurity risk management is risk identification (RI). This begins with asset valuation, meaning a consistent approach to determine both quantitative (cost) and qualitative (relative importance) values. Factors include costs to acquire or develop an asset, initial and recurring data maintenance costs, importance of assets to the organization and to others (including criminals), and public value in terms of things like intellectual property. Next comes threat analysis, involving identification and definition of known and potential threats, consequences of an exploitation, the estimated frequency of threat events, and the probability that a potential threat will materialize. The third element of RI is vulnerability assessment. This can draw on a variety of sources, including published lists of vulnerabilities, a history of security events, and results of vulnerability testing.

Now comes risk analysis (RA), which can be quantitative or qualitative. Quantitative RA has advantages in devising an optimum cybersecurity solution but requires more information and more effort. It attempts to assign a numerical value or cost to assets and threats to support cost/benefit analysis and more clear and concise characterization of particular risks. NIST SP 800-30, *Risk Management Guide for IT Systems* [9], defines a nine-step risk assessment process. The analysis begins with the asset valuations and assessments of threats and vulnerabilities from RI. For each combination of an asset, a vulnerability, and a threat, an Annualized Loss Expectancy (ALE) is calculated as:

$$\text{ALE} = \text{SLE} \times \text{ARO} \tag{10.2}$$

where SLE is the Single Loss Expectancy and ARO is the Annualized Rate of Occurrence, stated as the expected number of occurrences of a specific threat in a 12-month period. SLE is determined as:

$$\text{SLE} = \text{Asset Value} \times \text{EF} \tag{10.3}$$

where EF is the Exposure Factor, consisting of the fraction of total Asset Value expected to be lost in a single security event. Obviously, many of the values used in these calculations are estimates, although careful tracking of published security data and analysis of an organization's security history and of recent event logs can make the estimates more accurate.

The next step is to evaluate the feasibility and cost of possible security controls. It can be difficult to arrive at the true *total* cost of a safeguard. In addition to the cost of developing or purchasing a product that implements a control, there are likely to be costs associated with planning, implementation, and maintenance; operating costs (including skilled personnel); and costs associated with economic impacts such as effects on productivity. Once a reasonable total cost is obtained, the value of a given control against a given threat can be computed as:

$$\begin{aligned}\text{Control Value} = {} & \text{ALE}(\text{without control}) - \text{ALE}(\text{with control}) \\ & - \text{Total Cost of Control}\end{aligned} \tag{10.4}$$

Qualitative RA is a heuristic method (Delphi technique) based on scenarios. It does not attach absolute numerical costs, but typically uses scoring by knowledgeable people on a 1–5 or 1–10 scale of threat severity, potential loss, effectiveness of protective measures, etc. These subjective assessments can be compiled to get some insight into the relative importance of risks and the relative cost and effectiveness of countermeasures. When quantitative RA is not feasible, qualitative analysis is better than nothing.

With the results of RI and RA in hand, risk management moves to decisions about how each risk should be treated. There are four basic options, representing the available safeguards:

- *Risk Reduction* – application of security controls or countermeasures to mitigate vulnerabilities and thereby reduce risk
- *Risk Assignment or Transference* – movement of a risk to a third party, e.g., by using insurance
- *Risk Avoidance* – action to eliminate the situation that gives rise to a risk, e.g., by disposing of assets or modifying vulnerable processes
- *Risk Acceptance* – deciding to live with a risk when the cost of mitigating it exceeds the potential consequences of a compromise

Several factors may bear on risk treatment decisions. In addition to the cost/benefit associated with implementing a security control, which may indicate accepting the risk or pursuing risk avoidance, some of these considerations are:

- *Legal Liability* – becomes important when there are statutory requirements or civil liabilities associated with failure to implement security controls.
- *Standards* – publications from NIST and other standards bodies offer guidance on optimum risk treatments.
- *Operational Impacts* – may be a factor if failure to mitigate a risk has potential operational impacts such as added cost for security surveillance, reduced productivity, or the need to implement work-arounds in affected processes.
- *Technical Factors* – a situation in which a proposed security control to deal with one risk actually introduces new vulnerabilities and thus leads to other collateral risks.

From all these considerations, the last step is to decide which risk treatments to apply based on an integrated assessment of the most effective protection strategy within the constraints of available resources and system impacts. Another way to say this is that we seek to reduce a system's Attack Surface to the feasible minimum. The remaining *residual risk*, which might be quantified by summing the ALEs of the untreated risks or simply described in qualitative terms, must correspond to acceptable risk level as defined in policy and approved by senior management of the organization. Security policy and requirements are ultimately based on achieving this level, and it is essential that responsible authorities in the organization approve it as being acceptable. This is then the basis for granting or withholding permission to operate the system with sensitive data and processes. RI, RA, security controls, and assessment of residual risk should be repeated as necessary in the face of the

constantly changing threat environment, specifically whenever new threats, new vulnerabilities, and changes to system content and processes make the previous analysis no longer valid.

A number of systematic threat modeling approaches are available. One is the Microsoft Threat Modeling Process, which is supported by a free tool and is complemented by threat analysis tools [10]. Once a set of security objectives for a system has been established, this process goes through a sequence of characterizing and decomposing applications to discover trust boundaries, data flows, entry and exit points, and external dependencies that potentially introduce vulnerabilities. These are then analyzed in terms of known threats and attack vectors, using methods such as attack tree analysis, resulting in a prioritized threat list. Finally, actual and potential vulnerabilities are identified, and the analysis results are fed back to the application characterization stage to support design changes, addition of countermeasures, or other steps to mitigate the threats. Other threat modeling methodologies include:

- *STRIDE* – this model uses six basic threat categories: Spoofing Identity, Tampering with Data, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege; it then identifies mitigation strategies for each.
- *DREAD* – this approach scores each identified threat in terms of Damage Potential, Reproducibility, Exploitability, Affected Users, and Discoverability from 1 to 10 and then computes overall risk by dividing the sum of these scores by 5.

10.4 Cybersecurity Guidance and Resources

Organizations that develop, own, and authorize secure information systems commonly establish policies and other directives governing these systems and their operations. The flowdown of security policy from overall organizational policies and objectives was described earlier. These policies often draw upon a growing body of standards published by Government, industry, and other bodies. System architects and engineers need to be familiar with this aspect of cybersecurity.

Table 10.1 lists some of the more prominent sources of cybersecurity data and guidance. The list contains Government agencies, industry consortia, professional societies, and others. Collectively, they offer current cybersecurity intelligence, tools and methods, standards, training, and many other resources of great value to the secure system architect, as well as later to the security staff that must support and maintain adequate security over a system's operational lifetime.

As Table 10.1 suggests, there is a large and diverse community striving to cope with the ever-growing and constantly changing cyber threat environment. As part of these activities, a number of standards and taxonomies have been developed.

- *Vocabulary for Event Recording and Incident Sharing (VERIS)* – a set of metrics designed to provide a common language for describing security incidents in a

Table 10.1 Sources of cybersecurity information and guidance

Source	Information provided
Web Application Security Consortium (WASC)	Best practice security standards, tools, resources, and information, e.g., Web application scanner evaluation criteria
Open Web Application Security Project (OWASP)	Tools, articles, other resources; development/testing/code review procedures, list of top risks
Department of Homeland Security (DHS)	Best practice tools, guidelines, rules, principles, other resources; Build Security In (BSI) initiative; provides CWE (below) as a service; SW assurance Common Body of Knowledge; comprehensive guidance on secure SW development at buildsecurityin.us-cert.gov
Institute of Electrical and Electronics Engineers (IEEE), Computer Society (CS), Center for Secure Design (CSD)	Recent initiative aimed at providing a variety of artifacts for secure system development, e.g., "How to Avoid the Top Ten Software Security Flaws"
ISO/IEC 27034	Standard guidance on integrating security in processes for managing applications
System Administration, Networking, and Security Institute (SANS)	Education, certification, reference materials, conferences dealing with information security
Council on CyberSecurity	Established in 2013; publishes a list of prioritized Critical Security Controls (CSCs) that are widely used as a measure of cyber resilience (www.counciloncybersecurity.org)
National Institute of Standards and Technology (NIST)	Federal Government lead for cybersecurity, e.g., Risk Management Framework (RMF); multiple standards and best practice publications, including SP 800 series and Federal Information Processing Standards (FIPS)
National Security Agency (NSA)	Published the original "Rainbow Books" on computer and network security; operates the Center for Assured SW
SW Assurance Metrics and Tool Evaluation (SAMATE)	Provided by DHS and NIST
Common Attack Pattern Enumeration and Classification (CAPEC)	Sponsored by DHS and MITRE (http://capec.mitre.org/index.html); lists common attack patterns, comprehensive schema, and classification taxonomy; ~500 entries (so far)
National Vulnerability Database (NVD); NIST SP 800-53	Federal Government repository of standards-based vulnerability management data; uses the Security Content Automation Protocol (SCAP) which includes the Extensible Configuration Checklist Description Format (XCCDF); supports automation of vulnerability management, security measurement, and compliance; includes databases of security checklists, security-related software flaws, misconfigurations, product names, and impact metrics; supports the Information Security Automation Program (ISAP); provides a list of Common Vulnerabilities and Exposures (CVEs) and scores CVEs to quantify the risk of vulnerabilities, calculated from a set of equations based on metrics such as access complexity and availability of a remedy; includes a Computer Platform Enumeration (CPE) dictionary

structured and repeatable manner [2]. VERIS helps organizations collect and share useful incident-related information.

- *Common Weakness Enumeration (CWE)* – a common language for describing software security weaknesses in architecture, design, or code; this is the standard measure for SW security tools targeting these weaknesses and a common baseline standard for identification, mitigation, and prevention of software weaknesses [11].
- National Vulnerability *Database (NVD)* – a Government repository of standards-based vulnerability management data, described in NIST Special Publication 800-53.
- *Vulnerability Description Standards* – some of these are listed in Table 10.1; they include:
  - Common Vulnerabilities and Exposures (CVE)
  - Common Configuration Enumeration (CCE)
  - Open Vulnerability Assessment Language (OVAL)
  - Computer Platform Enumeration (CPE)
  - Common Vulnerability Scoring System (CVSS)
  - Extensible Configuration Checklist Description Format (XCCDF)
- *Common Attack Pattern Enumeration and Classification (CAPEC)* – a dictionary and classification taxonomy of known attacks that can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.
- *Other Public Materials* – multiple publically available security taxonomies, research reports, checklists, newsletters, blogs, and other data sources.

10.5 Secure Architecture and Design

Given an effective cybersecurity policy and associated set of requirements, a subdiscipline of Systems Engineering that is generally referred to as System Security Engineering (SSE) comes into play as part of the overall MBSAP flow from requirements to an OV, LV, and PV. At every step, security requirements are allocated to a progressively more detailed system structure, and security functions and processes are modeled as part of the Behavioral Perspective. The overall strategy recommended in this book can be called "layered defense," and the layers of this structure are defined in parallel with other architecture development activities. Layered defense has two complementary elements:

- *Defense in Depth (DiD)* – deployment of diverse and mutually supportive security controls at various points in the architecture to create, in effect, concentric shells of protection that an attacker must defeat to reach protected assets. Later paragraphs and Appendix F go into more detail on this approach. DiD can be thought of as security in depth.

- *Zero-Trust Architecture* – a strategy that extends and refines DiD architectures with fine-grained segmentation, strict access control, strict privilege management, and advanced protective devices, all based on a rule that no person and no resource is trusted without adequate verification. Zero-Trust represents a broad approach to secure systems.

10.5.1 Secure Architecture and Design Process

Figure 10.5 shows the primary steps of a typical end-to-end security architecture process. As with architecture strategy in general, this flow is highly iterative, and results at any step may cause a return to an earlier step to correct a problem, refine a requirement, perform additional analysis on a discovered issue, and so forth. By making this an integral part of an MBSAP effort, security architecture becomes an integral part of the overall architecture of a secure system or enterprise, not something that is ad hoc or independent. Cybersecurity requirements, functions, and standards will impact many aspects of the system and may be the dominant factor in decisions such as product selections. Security testing involves a distinct set of events, but it must be integrated with the overall test plan to ensure all required data is collected at minimum cost and effort. Similar considerations apply throughout the MBSAP process.

The following paragraphs summarize the primary actions at each step in Fig. 10.5.

- *Planning and Initiation* – as part of basic program planning, establish the methodology, resources, schedule, and other basics of the security architecture development; this includes securing commitment from program management, customers, and other stakeholders.
- *Threat Analysis and Risk Assessment* – perform the analysis activities described earlier under Risk Management.
- *Cybersecurity Policies and Requirements* – once the RI, RA, and risk treatment analysis is complete, create a security policy, SP, SECOPS, and specific security requirements.
- *Cybersecurity Architecture* – define the security architecture under MBSAP, including:
  - Selecting and prescribing security boundaries and mapping them to system structure and partitioning.
  - Defining and modeling the implementation of selected security controls at each level of a layered defense structure.
  - Defining and modeling specific security functions and services; for example, security services like user authentication and access control are usually made available to both infrastructure and application components of a system.
  - Defining metrics to be used to assess the quality and policy compliance of the security architecture; another NIST Special Publication, 800-55 [12], offers guidance on the selection and application of security metrics.

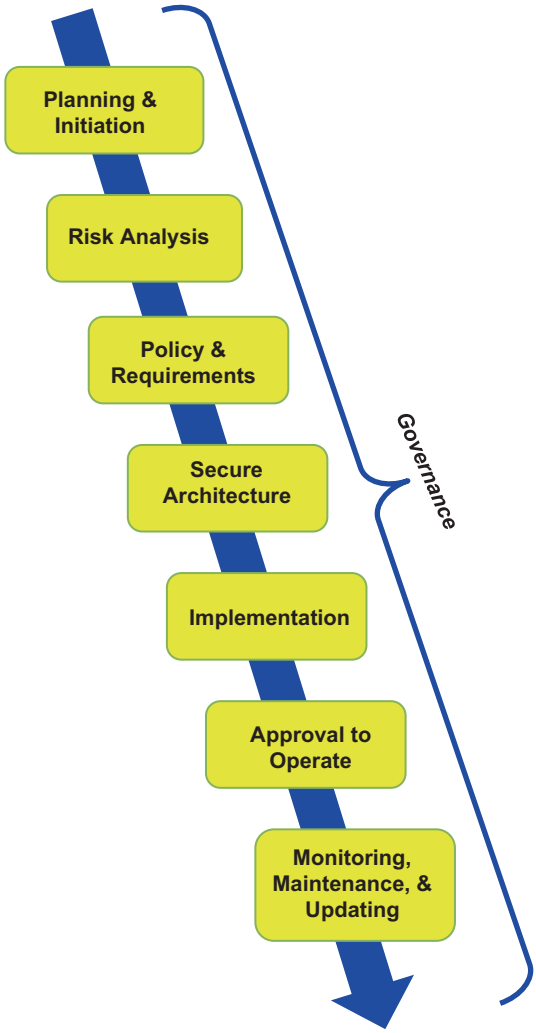


Fig. 10.5 Basic flow of a security architecture process

  - Defining roles and responsibilities of system users, security managers, and others.
  - Defining other aspects of the security architecture such as required trust levels for various system components; this must obviously be done in close collaboration with the system architecture development as a whole, and there will be security content in the OV, LV, and PV.

- *Implementation* – as described in more detail in the next section, implement the security architecture within the PV, including:
  - Selecting and configuring components and processes that implement security controls

  - Verifying that system components have appropriate required trust levels based on the nature of the threat and the sensitivity of protected information and that selected products comply
  - Developing security-related architecture artifacts and documentation
  - All other aspects of building the system in conformity to requirements and the security architecture

- *Approval to Operate* – once the implementation is complete, accomplish the analysis and testing to verify that the required risk level has been achieved. Regardless of whether a system has a Government or a private customer, adequate testing must be planned and conducted as an integral part of the total test program. Ultimately, a body of evidence is presented to an appropriate authority to support an approval to operate decision. This is further discussed under System and Component Trust Evaluation below.
- *Monitoring, Maintenance, and Upgrading* – during system operation, provide the resources and procedures for monitoring functions such as event logging, configuration audits, and vulnerability testing. Since technology, requirements, and threats continuously evolve, even the best security architecture and implementation will require periodic updating and retesting to maintain assurance that the required risk level is still being met. Plans for long-term system support and upgrading need to account for the inevitable updates and problem fixes associated with system security components and functions. Specific activities in this area include:

  - Configuration management to protect against unknown or unapproved content
  - Prompt installation of software patches as soon as released to deal with new vulnerabilities
  - Technology refreshment as cybersecurity components, like any other products, evolve, become obsolescent, and require replacement or upgrading
  - Addition, upgrading, or reconfiguration of protective measures to deal with new or changing threats and policies
  - Education, training, and security awareness of personnel to ensure the human element of the cybersecurity posture does not degrade as personnel and system functions change

- *Governance* – as shown in Fig. 10.5, implement governance at every step and throughout a system's lifetime. Valuable guidance is provided in NIST Special Publication 800-100 [13], which is targeted to Government agencies and policies but provides useful insight for private security architects and managers as well; Chap. 3 is especially relevant to secure system development. This document stresses the importance of clear individual and organizational roles and responsibilities, regular monitoring of security features and policy compliance, periodic reassessment of the effectiveness of security controls and procedures using measurements and metrics, rigorous configuration management and

change control, and continuous analysis of incidents, supported by auditing of system logs.

A number of Government and private sector organizations issue guidance on design of a resilient cybersecurity solution. Rozanski and Woods [14], writing about software architecture from a largely commercial point of view, suggest the following elements of a security approach:

- *Principle of Least Privilege* – grant system users only those accesses and capabilities required to perform their roles.
- *Secure Weakest Link* – focus protection on system elements that are most vulnerable and most likely to be attacked.
- *Defend in Depth* – this is essentially layered defense.
- *Compartmentalize Content* – structure and manage information such that access to a given data area does not accidentally expose information that should be separately controlled.
- *Keep Security Simple* – avoid implementations that are hard for system users and administrators to work with and therefore invite cheating, shortcuts, or work-arounds.
- *Establish Secure Defaults* – ensure that if failures or anomalies force a system to revert to default configurations and settings, these still provide adequate protection.
- *Fail Secure* – similarly, ensure that if a system fails or shuts down, it does so in a state that does not expose sensitive content; a typical example is to encrypt all data at rest (i.e., in nonvolatile storage); this is the security equivalent of "fail-safe."
- *Start Up Secure* – the inverse principle is that upon startup, a system must activate all protective measures before accessing or exposing sensitive information.
- *Ensure Trusted Externals* – implement protections that provide adequate assurance of the identity, trust level, and access privileges of any external entity trying to use the system.
- *Log, Analyze, and Audit* – provide the capability to detect and log all security-related events, and implement a rigorous process of analysis and auditing to make full use of the recorded event data.

The Department of Defense focuses on the following essential aspects of cybersecurity policy for the DoD Information Network (DoDIN), supporting the Global Information Grid (GIG); these apply to most private sector cybersecurity situations as well:

- *Identification and Authentication* – the policy requires both non-forgeable credentials and the use of a Strength of Mechanism assessment to ensure acceptable risk that unauthorized individuals will be able to gain access to classified systems and data.
- *Access Control* – the policy directs that traditional Role-Based Access Control (RBAC) give way to more powerful and flexible approaches such as Risk-Adaptive Access Control (RAdAC) as described later in this chapter.

- *Privilege Management* – the policy calls for adaptive and flexible privilege management to ensure all authorized users have timely access to the resources needed for mission accomplishment in accordance with the local operational situation.
- *Trusted Computing Platforms* – the GIG requires that participants implement TCPs, also described below.
- *Secure End-to-End Communications* – the policy prescribes migration to a "black core" in which sensitive information is always encrypted when it is outside a secure environment; ideally, data is encrypted at the computer that produces it and decrypted at the computer that consumes it.

Such policy requirements could drive anything from the selection of biometric means for user authentication to the development of new algorithms for real-time adjustment of individual privileges. For example, most security guidance documents now call for two-factor or multifactor authentication, involving two or more independent means of verifying identity, in place of sole reliance on passwords that are all too frequently compromised. In a private sector context, a security architect might well be called upon to implement controls that give citizens of multiple countries in a global company or enterprise access to appropriate data and processes while ensuring no export control laws or company policies on information release are violated. These examples illustrate the importance of very thorough analysis of policies and security requirements early in the architecture cycle, followed by allocation to the elements of an emerging system architecture and design, to achieve the objective of embedding security features and functions in the system.

10.5.2 Layered Defense Architecture

To provide a basis for discussing layered defense, Fig. 10.6 shows a notional networked enterprise to which a variety of risk mitigations can be applied. This is at least representative of the contents of a wide assortment of real-world information systems. As used in MBSAP, layered defense has two components: a DiD structure of successive layers of protective mechanisms and Zero-Trust Architecture, which enforces segregation and strict access controls for all sensitive content.

Layered defense begins as a dimension of the secure system architecture and continues in the selection and deployment of products and processes for security controls at various points. This is a much larger topic than can be fully treated in a single chapter. Accordingly, the emphasis here is on key aspects of secure system implementation that are likely to be important to a general systems engineer charged with satisfying the full scope of system requirements, including security. The following paragraphs describe some important considerations in devising a DiD structure for a system like that in Fig. 10.6. Table 10.2 lists a typical set of layers in a DiD structure for such a system with representative safeguards and functions. Appendix F gives an extensive set of security control definitions for these layers, many of which are referenced in Fig. 10.6.

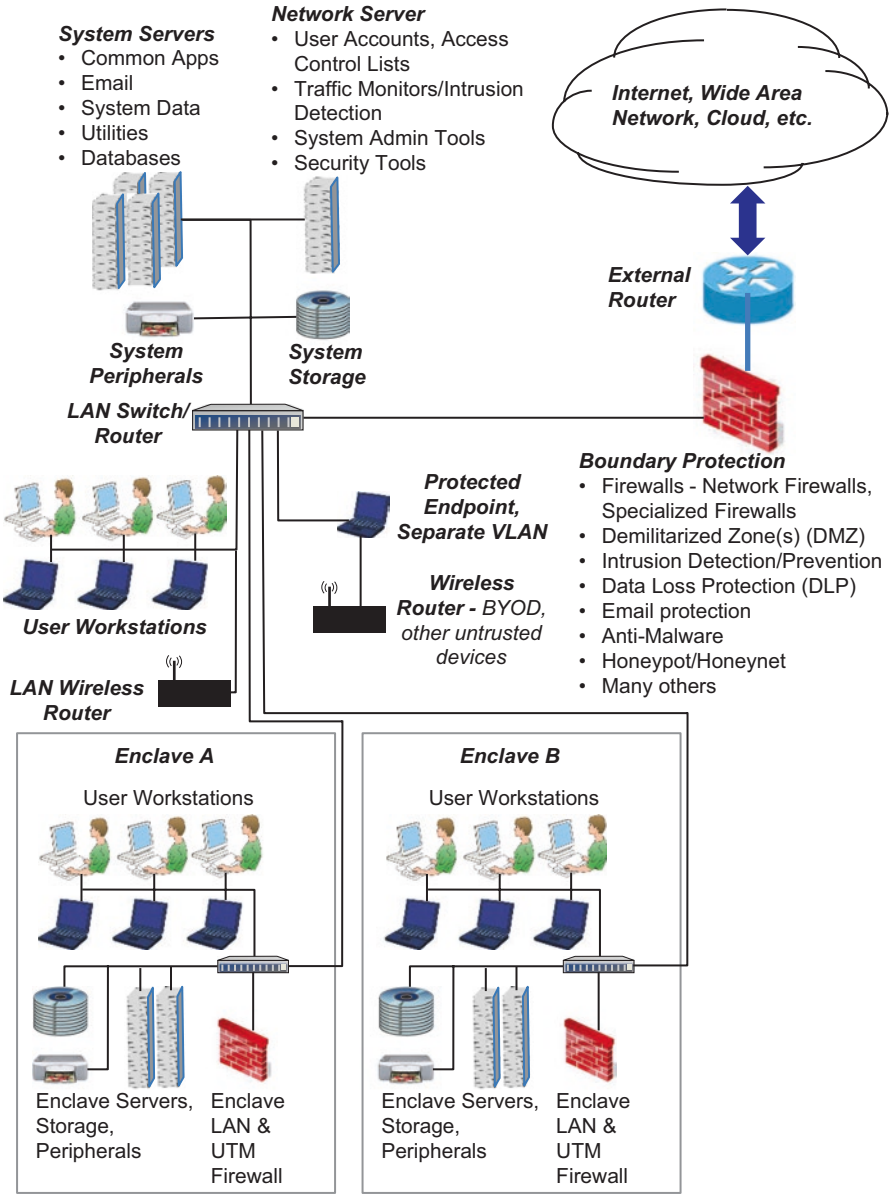


Fig. 10.6 Notional networked enterprise requiring defense-in-depth protection

Boundary Security A Boundary Security System (BSS) protects the point of entry from the external network environment and implements functions like those listed in Fig. 10.6. Some specific examples include:

- *External (Boundary) Router* – may provide some protective functions such as address filtering on messages.

Table 10.2 Examples of defense-in-depth protective measures

Protective layers	Protective measures	Protective functions
System Boundary or Perimeter	• Firewall(s) • External Networking • Demilitarized Zone (DMZ) • Intrusion Detection System/ Intrusion Prevention System (IDS/IPS) • Encryption/Decryption in boundary Protection Devices • Data Loss Prevention • Honeypot/Honeynet • Anti-Virus/Anti-Malware	• Block unauthorized information transfers and scan for malicious or prohibited content • Allow connections only to trusted addresses • Isolate system resources from external access; provide address space for protective devices and tools • Monitor external activity and log unapproved traffic; IPS blocks untrusted messages • Ensure sensitive ("red") content is converted to unreadable ("black") before release from a protected environment • Detect and block unauthorized efforts to extract ("exfiltrate") data • Trap would-be attackers in a "sacrificial" server; gather data for attack analysis • Detect/defeat malware at system boundary
Network	• Network-Based Intrusion Protection/Access Control • Enclave Segregation with Firewalls • Anti-Virus • Labeled/Protected Messaging • Message Integrity Checking • Encryption	• Monitor network traffic, detect suspicious or prohibited use, block unauthorized use • Provide customized protection for separate network enclaves • Detect/defeat malware in network traffic • Restrict traffic to authorized participants, prevent/detect message corruption or tampering • Detect tampering with network traffic • Eliminate unprotected data on the network
Endpoint	• Operating System (OS) Lockdown/Hardening • Host System Security (HSS) • User Accounts • OS Control File Protection • Administrative Separation of Privileges • Firewalls/Intrusion Detection	• Disable functions not required for system operations that may create vulnerabilities • Detect/prevent unauthorized or suspicious access attempts, malware, other hostile actions • Enforce access controls, user authentication, principle of least privilege, etc. • Prevent unauthorized changes to controls • Restrict access to security controls and settings • Customize protection for endpoint devices

(continued)

Table 10.2 (continued)

Protective layers	Protective measures	Protective functions
Application Software	• Logging and Auditing • Software Testing and Trusted Software • Specialized Firewall(s) e.g., Web Application and XML • Content Monitoring/Filtering	• Detect suspicious/prohibited access to apps • Rigorously analyze and test software to preclude malicious code and other vulnerabilities; detect/prevent unauthorized modifications • Apply customized protection to applications • Detect suspicious or unauthorized processing
Data	• Logging and Auditing • Encryption of Data at Rest • Data Loss Prevention (DLP)	• Detect suspicious or prohibited access to data • Ensure even theft or other physical compromise cannot obtain sensitive data • Block unauthorized data access/ exfiltration

- *Firewalls* – a variety of types can be deployed to inspect various aspects of in- and outbound traffic and block suspicious or unauthorized messages.
- *Demilitarized Zone (DMZ)* – internal and external firewalls can be used to create a "buffer zone"[3] between the untrusted external environment and sensitive system resources; this can host a variety of protective measures.
- *Intrusion Detection System/Prevention System (IDS/IPS)* – specialized device or software that detects and (if an IPS) blocks and records suspicious activity.
- *Data Loss Protection (DLP)* – identifies, monitors, and protects data in use, data in motion, and data at rest through deep content inspection and other sophisticated techniques.
- *Proxy Servers* – servers that allow external users to request and receive services such as Email and applications without being granted direct access to the actual servers and data.
- *Malware Protection* – software that tries to detect malicious software and prevent it from executing.
- *Honeypot* – a "sacrificial" server that mimics real system resources to attract an attacker and gather information; multiple honeypots create a "honeynet."

System Resources There is a set of overall system resources, including servers, workstations, storage, peripherals, and a local area network (LAN) router. These provide functions needed by all system users such as account management and common software applications. All these components are candidates for protection using tailored security controls.

[3] The implied parallel to the situation between North and South Korea is quite deliberate.

Protected Enclaves The specialized functions of the enterprise are isolated in enclaves, each with its own LAN and a Unified Threat Management (UTM) firewall. Only two enclaves are shown, but there would typically be an enclave with restricted access for each individual department or other organizational element of the enterprise. This segmentation is a core principle of Zero-Trust Architecture. As with all cybersecurity devices, it's important that candidate UTM products be carefully evaluated for their protective capabilities, especially against the assessed threats to a system.

Protected Wireless Connection The system LAN has a wireless router for connection of approved (trusted) devices. There is also a wireless router for other access, such as by personal (Bring Your Own Device, or BYOD) devices. This router is protected by a dedicated processor that implements a variety of security controls and establishes a virtual LAN (VLAN) as described in Chap. 9 that is separate from the system network.

The other key element of layered defense is a Zero-Trust (ZT) Architecture [15], and Table 10.3 gives some of the main features for a system like that in Fig. 10.6. The name "Zero-Trust" derives from the basic principle that no one and nothing is trusted beyond a single authenticated instance. ZT complements and strengthens DiD by adding stringent controls, especially for access to sensitive content. For example, a system that makes an authentication and authorization decision for every request for access to sensitive content provides a higher level of assurance than one that requires only one-time access approval for an entire user session.

All of the DiD and ZT features in Tables 10.2 and 10.3 could be applied to a system like the one in Fig. 10.6, depending on the assets to be protected and the risks to be mitigated. One of the major objectives of adding a Zero-Trust network to a DiD architecture is to minimize the damage an attacker can do after successfully

Table 10.3 Features of a Zero-Trust Architecture

Protective features	Protective functions
Zero-Trust	• No system users or resources (applications, databases, etc.) are trusted a priori • Every resource access, installation action, etc. requires explicit authentication and authorization
Address control	• All resources are protected by strong access controls, regardless of location (e.g., multifactor authentication of users) • Enforces highly granular access to resources (specific permissions)
Network segmentation	• Network is divided into multiple segments enclosing related resources and functions • Enclaves have distinct protected perimeters and enforce access controls • Enclaves are protected by multifunction devices such as a Unified Threat Management (UTM) firewall[a]
Network security	• Implements a secure packet forwarding engine • Monitors all network traffic for suspicious activity

[a]Forrester Research has coined the term "Segmentation Gateway" for this device

penetrating a system. Every attempt at "lateral movement" from one resource area or enclave to another must defeat a layered defensive structure within the security perimeter of that enclave. Additionally, every attempt to access a protected asset requires strong authentication of the party requesting access and fine-grained authorization (access permission) for every specific asset. DiD with Zero-Trust, using advanced safeguards such as those which apply statistical behavior analysis and artificial intelligence, represents the current state-of-the-art in secure architecture.

As an example of secure architecture strategy based on deployment of multiple, complementary safeguards, Jones and Horowitz [16] have published an approach to protecting critical utility infrastructure, especially nuclear power plants. Basic elements of their proposal are:

- Use of diverse and redundant subsystems, increasing the effort an attacker must make to defeat a given layer of protection; this can also draw out a "signature" by which an attacker can be identified and recognized.
- Configuration "hopping" to restore a compromised system by switching to an alternative configuration.
- Data consistency checking using techniques such as voting among parallel diverse components.
- Tactical forensics to distinguish true attacks from false alarms or natural failures.

10.6 Secure System Implementation

Once an approved security architecture is in hand, the next stage of the process in Fig. 10.5 is implementation. We assume here that a set of security controls and other risk treatments has been selected and embodied in an architecture with the appropriate features and functions. These must now be realized with hardware and software components, procedures, staff training, and other elements of a cybersecurity solution. The system must then be tested to confirm that security controls function as designed, that no known vulnerabilities remain open, and that the required residual risk level has been achieved. The following paragraphs discuss implementation of these controls in the context of the enterprise in Fig. 10.6.

10.6.1 Boundary Security

A DiD design begins with protective measures at the system or enclave perimeter as described in Sect. 10.5.2. In an older security architecture, boundary security may well be the primary defense. This approach has been called "crunchy outside, chewy inside" and is now thoroughly obsolete. There is no such thing as a standard BSS design, but Fig. 10.7 shows some typical components in an Internal Block Diagram.

The Blocks associated with the BSS have a corresponding stereotype. The system connects to the outside world, which might be a public network like the Internet, a private wide area network (WAN), a cloud environment provided by a third party, or some other external environment, through a Boundary Router. There may also be other channels such as encrypted point-to-point links, and Fig. 10.7 includes a Virtual Private Network (VPN) server used by authorized parties outside the enterprise. This and other networking approaches are discussed in Chap. 9. There may be an external Network Switch if the Boundary Router needs this functionality. The Boundary Router normally uses the Border Gateway Protocol to establish a session with another autonomous system on the network. It may participate in the security scheme by filtering incoming traffic, e.g., by applying Time of Day limits to exclude messages that fall outside established time parameters.

Perhaps the most ubiquitous security component is a firewall, which is a specialized computer, running trusted software, that applies a range of techniques to

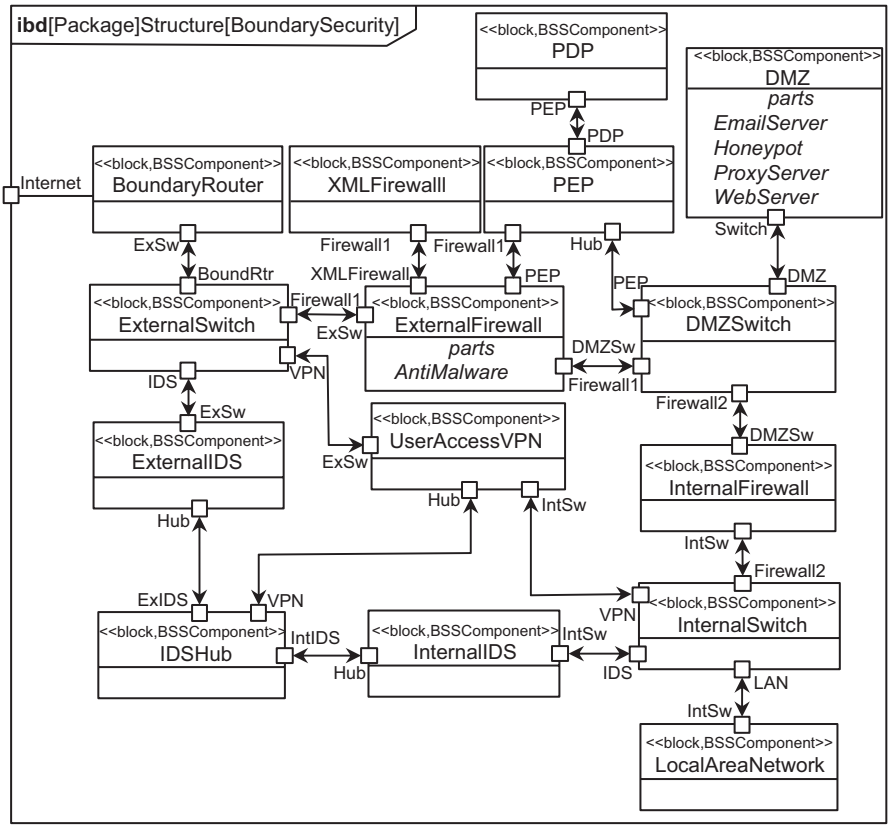


Fig. 10.7 Typical elements of a boundary security system, forming the perimeter security layer of a DiD implementation

incoming and outgoing traffic to identify and block unauthorized or threatening messages. The firewall has rule sets that can be programmed by security managers to establish the basis for allowing or preventing messages from passing, as well as an Access Control List (ACL) that identifies authorized external sites and users.

Firewalls apply a variety of filtering techniques, usually at the packet and protocol or application levels, based on user-supplied rules and parameters. A packet filtering firewall, which can be a software component running in a sophisticated boundary router, permits, rejects, and drops packets based on parameters such as IP address and port number. A proxy firewall runs proxy gateways for mail and other kinds of traffic using rules for context, authorization, and authentication. These, and other firewall types, can be combined to meet specific security requirements. A sophisticated boundary security system may have a DMZ with both external and internal firewalls, which is further described in a later paragraph. We also note that any firewall will have a certain number of false negatives (legitimate access requests that are blocked) and false positives (the opposite mistake). System and Network Administrators may be required to "tune" the firewall rules to achieve acceptable levels of these as defined by organizational security policy. For example, it might be acceptable to have 1% of legitimate traffic denied if that's the consequence of blocking 99.9% of malicious traffic.

As Fig. 10.7 indicates, a firewall may be supported by a variety of specialized security controls. The External Firewall is shown as including an Anti-Malware Part and interacting with an XML Firewall that specifically scans XML traffic and protects XML-based interfaces. The Internal Firewall is shown as incorporating DLP, which is often deployed at multiple layers of a DiD structure. The dual firewalls in Fig. 10.7 are complemented by Internal and External Intrusion Detection Systems, connected by a Hub. An IDS detects unauthorized traffic by determining patterns of activity and matching them to known signatures that are identified with known vulnerabilities or attack vectors. It can apply a variety of algorithms at the packet, message, protocol, and application layers and may be set up to respond to network attacks, data-driven attacks, attempts to subvert access controls and privilege management, unauthorized log-ons, and unauthorized access requests. The IDS may also supplement malware protection software in dealing with Trojans, worms, and other attacks. A more powerful component is an Intrusion Protection System (IPS) that actively blocks "bad" traffic. To remain effective, an IPS must receive regular updates as new threats are identified; this is analogous to keeping malware detection libraries current.

Both public and private sector systems have begun to implement the DMZ concept as a way to make information available to authorized external users while protecting core resources. A DMZ typically provides a data buffer and task queue. It resides in a separate enclave with its own address space and may have an external Domain Name System (DNS) server to process external user identities. A DMZ can be implemented with one or two firewalls, the latter being more secure. Figure 10.7 shows the DMZ, accessed through a DMZ Switch, lying between the External and Internal Firewalls and hosting functions like an Email Server, a Web Server, and one or more Proxy Servers that expose data and processes from system servers without

allowing direct access to those resources. An Email Gateway is a server that accepts, forwards, delivers, and stores messages on behalf of system users. It typically also scans messages and attachments for viruses and other malicious content and can perform filtering to eliminate unwanted traffic (the notorious "spam"). A DMZ typically creates a "holding pen" for traffic to which the firewall can send suspicious incoming messages for additional processing before delivery or deletion. A further feature shown in the figure is a Honeypot that mimics actual system resources to trap malicious intruders and allow defenders to record information and to observe the details of an attempted penetration.

For many years, cybersecurity was based on applying rigid rules governing which users could access what information and processes. This makes it hard to accommodate the dynamic situations common in modern enterprises subject to sophisticated cyberattacks and can create a large administrative burden to fine-tune the privileges of a large and constantly changing user base. The current trend is toward policy-based security management in which rule sets take into account a variety of circumstances and make near real-time adjustments. For example, a specific individual might need access to specific data for a specific period of time, after which this privilege should be revoked. Or the access rules might need to become more stringent in response to a detected hostile penetration attempt. One implementation uses a Policy Enforcement Point (PEP), shown interacting with the External Firewall in Fig. 10.7, which is usually a dedicated computer that mediates user access requests, message traffic, and other activities on the basis of a set of policy attributes. In a typical situation, a user would request access to an application or data set, and the PEP would obtain the user's certificate and other attributes (see Access Control below); would apply current policy, provided by a Policy Definition Point (PDP), to determine if the requested access is allowed; and, if so, would authorize the requested activity. The PDP server gives System and Security Administrators important tools to manage access control.

On the inner edge of the BSS, an Internal Switch connects traffic that passes the Internal Firewall, along with VPN traffic and data from the IDS, to the system LAN. This is the next DiD level, with security controls like those listed in Table 10.2. The other layers have obvious mappings to endpoint devices connected to the network, to application software running on servers in the enclaves, and to data, wherever it exists and whether it is in use, in motion, or at rest. Space prevents a description of these layers as detailed as that just given for the Boundary or Perimeter Security Layer. However, Appendix F gives more detail about the candidate controls at each level that are summarized in the table. Because many cyberattacks seek to exploit weaknesses in Web-facing software, an important example is the use of a Web Application Firewall (WAF) using Deep Packet Inspection (DPI) either as part of the overall system BSS or in conjunction with the access point to an enclave.

10.6.2 Trusted Computing Platform (TCP)

The term TCP identifies a class of resources that satisfy stringent cybersecurity criteria and thus can contribute to highly trusted system designs. It was originally defined for Defense systems but can be applied to any secure system implementation. A TCP implements the following characteristics:

- The system boots to a secure state before accessing any applications or data.
- The system is trusted to load only signed software that is encoded to ensure that no modifications have been made to the code from the original developer.
- The system authenticates each client or user to the infrastructure.
- The system implements host-based (endpoint) intrusion protection.
- The system enforces separation of content by level of sensitivity/classification.
- The system encrypts data at rest and in motion.
- The system enforces security policies.
- The system sanitizes memory and storage upon shutdown or other defined conditions.
- The system performs client/user attestation.
- The system labels data with sensitivity and access control parameters.

The Trusted Computing Group (TCG) [17], composed of a number of prominent IT providers,[4] requires a TCP to enforce security features in hardware and software, including a secret encryption key that protects a computer even from its owner. The TCG has defined a Trusted Platform Module (TPM), which is a specialized chip using encryption to provide a hardware-based approach that is "a secure cryptographic integrated circuit (IC) [that] provides a hardware-based approach to such sensitive functions as user authentication, network access, and data protection." For example, the TPM allows the computer to furnish a certificate to a third party proving that trusted software and hardware have not been altered. To fully comply with the TCG specification, a computer must implement the following five specific concepts:

- *Endorsement Key* – a public/private key pair (see Encryption, below) that is embedded in the TPM
- *Secure Input/Output* – techniques that minimize security vulnerabilities associated with system I/O
- *Memory Curtaining* – extensions to normal memory protection to fully isolate sensitive memory areas
- *Sealed Storage* – binds private information to a specific hardware/software platform
- *Remote Attestation* – allows authorized third parties to detect changes to the hardware/software platform

[4] It is important to note that the TCG also seeks to improve data rights management, especially to prevent unauthorized copying and dissemination of copyrighted material.

10.6.3 Quality of Protection (QoP)

By analogy to Quality of Service (QoS, Chap. 9), security architects have defined a QoP construct that defines attributes of a data object describing how it is to be protected at rest and in transit. Interestingly, QoS and QoP are often in conflict, because the steps taken to secure information generally add overhead processing that can increase the latency in communicating the data. A QoP profile will typically define requirements for encryption, allowable routing, access controls, and destruction of a sensitive data object.

10.6.4 Encryption and Certificates

Schemes for converting information into a form that is only intelligible to an intended recipient using some form of encryption are as old as civilization. The original message or information is generally referred to as clear text or plaintext, and the encrypted version is called ciphertext. The encryption algorithm is commonly known as a cipher. "Code breaking" has played a decisive role in warfare for millennia, and commercial espionage that involves breaking into secure communications among companies and nations is widespread.

Cryptography has three primary uses:

- *Confidentiality* – make plaintext into unreadable ciphertext.
- *Integrity* – use hashing and message digests to ensure integrity.
- *Authentication and Non-Repudiation* – use digital signatures, certificates, and PKI with public key cryptography to prove the identity of the source of a message or transaction.

Cryptography is an entire discipline in its own right, and only a brief overview is possible here. Details of the types and classes of ciphers, the mathematics of encryption algorithms, tools and methods used to create and break codes, and other aspects are beyond the scope of this book. The following paragraphs discuss some aspects of particular interest to system architects and engineers.

Fundamentally, encryption methods are divided into secret key and public key (also called asymmetric key) approaches as summarized in Fig. 10.8. With a secret key, the recipient of secure communications provides an encryption key to any potential sender using secure means such as using a courier or sending parts of the key through multiple independent channels. Keys might be valid only for specific time windows or geographical areas and should normally be changed frequently and randomly to complicate the task of a code breaker. This can be very secure if a high-quality key is used, but it has significant overhead. By contrast, public key schemes eliminate the need for secure key distribution. The recipient publishes a public key while retaining and safeguarding a private key. The encryption mathematics is such that only the private key can decrypt ciphertext produced with the public key.

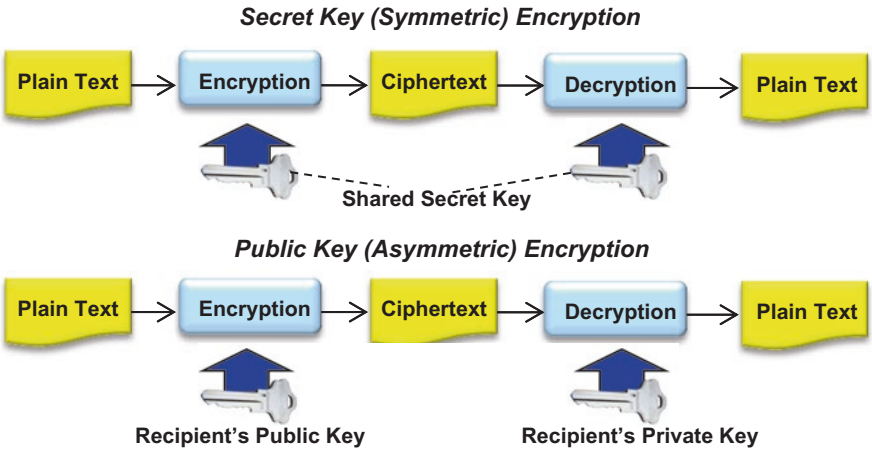


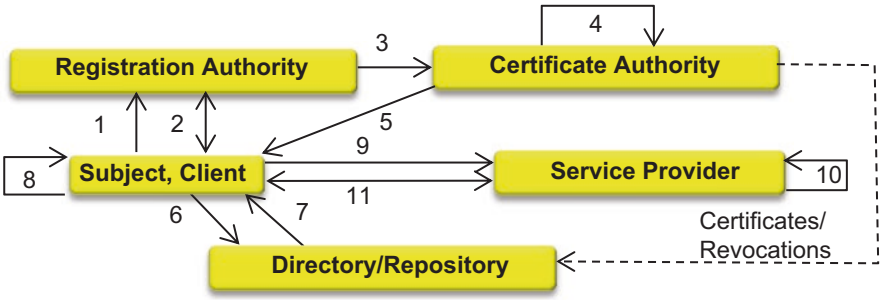
Fig. 10.8 Basic encryption schemes

A digital signature uses public key encryption to allow a recipient of a message to verify the identity of the sender and confirm that the content has not been altered. The sequence is:

- The sender computes a hash value[5] for the message and encrypts it with a private key; this is called "digital signing."
- The recipient also computes a hash value and decrypts the hash value that was sent, using the sender's public key.
- The recipient compares the hashes to confirm the sender's identity and message integrity.

Many encryption schemes involve trusted third parties, especially in the form of a Certificate Authority as sketched in Fig. 10.9. The Kerberos protocol, defined in RFC 4120 from the Internet Engineering Task Force (IETF), is the most common standard for mutual authentication between computers [18]. A Digital Certificate (or Public Key Certificate) is a document that binds a public key to one or more identification attributes of the key issuer, who may be the recipient of secure messaging or the client of a service provider. An enterprise can create a PKI that provides for issuing authentication credentials, such as an identity card with an embedded electronic certificate, to participants and can provide services such as certificate revocation if a credential is withdrawn or compromised, using a Certificate Revocation List (CRL). An alternative to PKI is called a Web of Trust scheme in which individual participants in an enterprise digitally sign their own certificates which are then attested by a third party. A popular implementation of this approach is Pretty Good Privacy (PGP), which allows use of email digital signatures and lets participants publish their public keys. A Web of Trust can interoperate with a PKI if

[5] Hashing is the application of an algorithm to a digital data object of arbitrary size to create a value of fixed size that can serve as a kind of fingerprint of the original; many hash algorithms are used.



1. Request certificate, provide subject public key
2. Request and receive ID credentials, verify subject ID
3. Forward request
4. Create and sign certificate with subject public key and ID
5. Send subject certificate
6. Request provider's public key
7. Receive provider's digital certificate, extract public key
8. Create session key, encrypt with provider's public key
9. Send session key and subject's digital certificate
10. Verify subject's certificate, decrypt session key
11. Communicate using session key

Fig. 10.9 Digital certificate process

all parties trust the Certificate Authority. Multiple CAs can be used to create a distributed ledger of certificates. Certificates can be used as part of a Non-Repudiation implementation, e.g., by using proof-of-origin and proof-of-receipt, as evidence that an individual participated in a transaction. There is precedent that an RSA signature (see next paragraph) with non-repudiation is legally binding.

Many encryption algorithms exist, and NSA develops and applies suites of algorithms for protecting US Government information. Two of the most widely used in private information systems are the Advanced Encryption Standard (AES) [19] and the RSA encryption algorithm, named for its inventors Rivest, Shamir, and Alderman [20]. An algorithm is often described in terms of the length of time a code breaker with high-end computing resources would need to penetrate it, which may range from minutes to years.

AES and RSA allow keys of varying lengths, the longer keys being harder to break, to allow system designers to trade off encryption overhead vs. level of communications security. A cryptographic checksum can be used to detect attempts to modify ciphertext in transit. Exotic new encryption schemes are emerging such as Quantum Key Distribution (QKD) which uses quantum mechanical effects to create secure keys and detect any attempts at tampering. It seems likely that the means to

securely encrypt even the most sensitive information will be adequate for the foreseeable future, at the cost of implementing complex and expensive encryption devices. FIPS 140-2, *Security Requirements for Cryptographic Modules* [21], is the Government standard for encryption devices.

NSA classifies encryption devices on the basis of the algorithms they incorporate and the level of sensitive information they are trusted to protect. The basic categories are:

- *Type 1* – trusted to cryptographically secure classified information
- *Type 2* – endorsed for telecommunications and information technology systems to protect unclassified national security information using classified NSA algorithms
- *Type 3* – endorsed to protect sensitive but unclassified information using unclassified algorithms
- *Type 4* – exportable devices using less robust encryption algorithms and keys

Encryption can be applied to multiple layers of a network protocol stack described in Chap. 9. At the Network or Internet Layer, the IETF defines the IP Security (IPSec) protocol suite that encrypts and authenticates each packet in an IP data stream. IPSec protocols can also perform mutual authentication and key negotiation between network participants, including individual users, servers, firewalls, routers, and others. At the Transport Layer, IETF defines the Transport Layer Security (TLS) protocol, successor to the earlier Secure Sockets Layer (SSL). In addition to securing the Transport Control Protocol (TCP), TLS has variants for web browsing, mail, instant messaging, and voice over IP (VoIP) digital voice communications. At the Application Layer, the Hypertext Transmission Protocol – Secure (HTTPS) combines HTTP with TLS using RSA public key encryption to create secure channels over nonsecure networks like the Internet. A number of schemes have been devised for securing Email, including Privacy-Enhanced Email (PEM) and PGP.

The black core concept mentioned earlier involves packet-level encryption and can be applied at the system boundary, at enclave boundaries within the system, or at individual computers, workstations, and storage devices. One approach uses individual encryption/decryption devices that conform to the NSA High Assurance IP Encryptor (HAIPE) specification and can, at some cost, be applied at any of the boundaries just listed.

10.6.5 Access Control

Like other aspects of cybersecurity, the mechanisms used to control the data and processes a user can access have become more sophisticated and flexible as threats have increased. Access control has major impacts on Confidentiality, Integrity, and Availability. A secure architecture can incorporate administrative or procedural, physical, and technical security controls to achieve acceptable risk that only

authorized subjects can gain access to protected objects. Access control can be centralized for a system or enterprise as a whole, or decentralized, e.g., at the level of individual enclaves. A closely related security control, noted in Fig. 10.7, is the use of IDS/IPS. A cybersecurity best practice, mentioned earlier, is to enforce the principle of least privilege, limiting each subject's access to categories of objects required by the subject's role or duties, and the even more stringent concept of need-to-know, which literally imposes access rules for individual objects. This is an important protection against the most damaging access control errors. Important elements of an access control strategy include:

- *Identity Management* – using mechanisms such as directories, password and other credential management, and user account management to verify that a subject has an identity known to the system
- *Authentication and Authorization* – verifying that a subject is who the party claims to be and determining the subject's permissions to a given object
- *Accountability and Auditing* – recording access control activities and analyzing these logs to support non-repudiation, threat warning, enforcement of policies and procedures, and other security goals

For many years, the standard was Role-Based Access Control (RBAC) under which a user who successfully authenticated his or her identity was given access according to the role contained in a user profile. More recently, RBAC has begun to give way to Attribute-Based Access Control (ABAC), especially for large systems with many unanticipated users [22]. Under ABAC, both users and resources have defined attributes, and policies, which can be adjusted dynamically, support access decisions. It was mentioned earlier that two-factor authentication, or more generally multi-attribute access control, which might combine a password with a biometric factor such as a fingerprint or with possession of a physical credential such as a token, is highly recommended and growing in use to strengthen access control. NIST has published a model for Risk Adaptive Access Control (RAdAC) in which an estimate of security risk and an estimate of operational need are computed in each situation and then used to make an access control decision [23]. Still another variant uses ABAC first to authenticate a user to a system, followed by RBAC to control the specific privileges the user will have.

The Organization for the Advancement of Structured Information Standards (OASIS) supports the eXtensible Access Control Markup Language (XACML) for defining schemas for authorization policies and access requests and responses. OASIS also supports the Security Assertions Markup Language (SAML), which is a profile of XACML that provides needed assertion and protocol mechanisms [24]. A secure system can use SAML tokens to authenticate the source of a message or file.

A party (a person or system) requesting access to protected data and resources under ABAC can have a variety of attributes, including:

- *Identity* – established by authentication, "who you are"

- *Certificate or Token* – used for authentication but can also contain access information, "what you have"
- *Password or Passphrase* – also used for authentication but can access a user profile with access parameters, "what you know"
- *Role* – "what you do"
- *Location or Source* – "where you are"

Data attributes can include classification or sensitivity, releasability rules, timeliness, a digital signature, and many others.

The standard model for an ABAC implementation has the following elements, some of which are illustrated in Fig. 10.7:

- *Policy Enforcement Point (PEP)* – receives access requests, forwards them for decision, and enforces decisions
- *Policy Decision Point (PDP)* – maintains current policy, compares requests to policy, and returns access decisions to the PEP
- *Policy Administration Point (PAP)* – stores and manages policies
- *Policy Information Point (PIP)/Policy Repository* – delivers policy data to the PDP
- *Attribute Authority (AA)/Attribute Repository* – provides attribute data

As enterprises become more networked and dynamic in their functions and participants, continuous evolution of access control mechanisms is likely. In particular, means to dynamically adjust access policies, for example, to accommodate unanticipated users or to clamp down system access in response to an attack, will be increasingly important in maintaining required levels of security while allowing systems and enterprises to function effectively. The ramifications of a networked enterprise and its environment include the need for effective access controls for a SOA and strong authentication and authorization for remote system users.

10.6.6 Virtual Private Networks (VPNs)

A VPN is popular way to create secure communications channels over nonsecure networks, especially the Internet. It is essentially a virtual network composed by using IPSec to tunnel through the unsecured transport layer of the public network. Users have a VPN client on their computers and have a token (often in the form of a key fob) that provides a frequently changing encryption key for a session. A basic VPN may simply consist of a set of point-to-point channels, but a modern VPN supports more complex and changing virtual network topologies. A VPN can be used within a system or node as a way to segregate communications among a certain set of users from the overall network. There are many variations on these basic VPN concepts. Increasingly, VPN functionality is available as a feature of browsers and operating systems.

10.6.7 Separation Kernels

One approach to using off-the-shelf hardware and software components in accreditable designs of secure systems is called a separation (sometimes partition) kernel, which was first mentioned in Chap. 8 [25]. Combined with the concept of virtualization (see Chap. 14) to create a Separation Kernel Hypervisor (SKH), this approach isolates hardware resources, host software, and guest operating systems in independent partitions with controlled information flows. The goal is to enable secure processing on commodity computing platforms whose design details are generally not available because they are proprietary to their vendors.

10.6.8 Multiple Levels of Security

In discussing Risk Identification, a key point was the need to characterize information by its sensitivity and required level of protection. Many systems are required to handle information at multiple levels and to ensure that more sensitive data is not exposed by being commingled with less sensitive data that has a lower level of protection. Such a system must be accreditable, and that means that there must be high confidence that information at a higher level of classification cannot be exposed at a lower, less secure level. Because an individual system user is likely to need to see data at multiple levels, the ideal solution would be a full Multi-Level Security (MLS) architecture with trusted mechanisms to ensure that every internal and external system user can only see data for which they have access permission and a need-to-know. This has presented an intractable challenge for many years, and the problem becomes even more difficult in situations where a system has international users and where legal considerations such as export control laws apply.

Full MLS is notoriously difficult, and the most common solution with current technology is a Multiple Independent Levels of Security (MILS) or Multiple Security Levels (MSL) architecture. In this approach, the system or enterprise is partitioned into enclaves or domains, each of which is trusted to store, process, and display information at a given level of classification and releasability. Such an enclave is said to run "system high." Each enclave or domain has a network, and it's common to use the network to label the domain. Thus, a company network might have enclaves for Company Private, Personally Identifiable Information, Public Information, and other kinds of restricted or confidential content.

When information must be exchanged across domain boundaries between different levels of sensitivity or trust, each having specific rules for user access, the normal approach uses specialized computers called guards. A guard applies a set of rules to a message, file, document, or other entity to be exchanged to validate that the exchange is allowed. Passing information from a lower to a higher level is much easier to deal with than the reverse. However, sensitive information can often be "downgraded" by removing especially sensitive content such that it can be released in a less trusted domain. Guard rule bases, sometimes supplemented by human

examination of the information, are the basis for this, and it is often highly desirable that the overhead processing and latency associated with a high-to-low transaction be minimized to maintain system performance. Increasingly, the trend is to base guards on XML documents, taking advantage of the fact that:

- XML documents are well structured, which facilitates automated rule processing.
- Technologies are available to describe XML messages and documents and to validate content against descriptions.
- Relatively mature security mechanisms exist for XML, e.g., for SOAP messaging, which may apply to cross-domain information exchanges.

In a MILS architecture, a user who needs to see data at multiple levels may need a separate computer or terminal for each domain network. In an MLS system, the user would have transparent access to any authorized information at any security level on one screen.[6] Accrediting such a system has historically been too challenging and expensive to be feasible. However, technology advances such as increasing use of trusted XML mechanisms may eventually solve the problem. This is an area where message and system access logging are also helpful.

10.6.9 TEMPEST

Because digital electronics can emit radio-frequency signals associated with the rapid switching of their logic, there exists a possibility that an adversary might detect and decode such signals and thereby obtain sensitive information. TEMPEST is the name usually given to design and test activities aimed at preventing this. Commercial computing equipment may require special enclosures with improved shielding and grounding. Buildings, aircraft, and vehicles that house classified computing environments may require testing to ensure detectable emissions are not present. When TEMPEST requirements are levied on a system, the architect has the responsibility to ensure that they are properly flowed down to the appropriate components and that TEMPEST test requirements are accounted for in overall test planning.

10.6.10 System and Component Trust Evaluation

The approach to specifying and verifying a level of trust for an information system has evolved over time in response to both the growing complexity of these systems and the increasing sophistication of cyber threats. Many of today's principles and

[6] It's been suggested that the litmus test for MLS would be the ability to drag and drop from a window at one security level to one at another level.

practices, in both the public and private sectors, originated in early Government work to protect sensitive information [26]. An early, widely used attempt to achieve defined levels of trust in information systems was documented in the "Rainbow Books" from the National Security Agency (NSA), so called because the books, more than 20 in all, each dealing with a specific aspect of security, had distinctive cover colors. The central volume, the Orange Book, or more formally *A Guide to Understanding Audits in Trusted Systems* [27], defined a set of security classes, with A1 representing the highest level of trust and D2 the lowest. A set of Trusted Computer System Evaluation Criteria (TCSEC) was published as DoD 5200.28-STD, December 1985, to provide a basis for evaluating commercial computer systems in terms of their security features and development processes. TCSEC has been largely supplanted by the Common Criteria approach, described below, and the entire field of secure system evaluation has evolved considerably.

Before being authorized to store, process, and exchange sensitive data, i.e., to "go live," a secure system should undergo a formal, documented process to verify that an acceptable level of trust, the inverse of security risk, has been established. Traditionally, such processes tend to be variants of the Certification and Accreditation (C&A) process that has been used for many years within the Federal Government. C&A has three basic elements:

- *Evaluation* – technical assessment of the protection features of the products used in a secure system design against stated criteria, independent of any mission or operational environment, based on analysis of design documentation and on testing, which ideally includes expertise from independent external test organizations
- *Certification* – a finding by system evaluators that security policies and requirements have been satisfied
- *Accreditation* – approval by a designated official or organization for the system to operate with sensitive content

Originally, C&A was a static process under which a system received an Authority to Operate (ATO) or something equivalent for a period of time, usually with a requirement for periodic security assessments to ensure adequate security was preserved. Accordingly, a C&A process typically proceeds through phases of system development, certification testing, accreditation, and reaccreditation.

Today, the familiar C&A approach is giving way to a more rigorous strategy based on *continuous* security evaluation, sometimes called Assessment and Authorization (A&A). Government policy now requires the use of what is known as a Risk Management Framework (RMF) [28]. All Government agencies are moving to implement RMF, which is defined in a series of publications from the NIST Computer Security Division [29, 30]. This improved strategy is also appropriate for a wide range of private sector systems and enterprises. RMF is similar in many ways to earlier C&A processes, including an emphasis on managing the risks to an organization and to the individuals and systems that implement information processes. It is different, however, in that periodic discrete C&A events are replaced by a continuous process of monitoring and assessment to ensure the integrity and policy

compliance of the information protection solution. Under NIST guidance, RMF also calls for deploying more, and more effective, security controls. The RMF approach has six steps:

- Categorize the system and its information content.
- Select the baseline security controls, with tailoring and supplementation as needed.
- Implement the controls.
- Assess the controls as implemented for effectiveness and appropriateness.
- Authorize system operations based on a determination of acceptable security risk.
- Conduct ongoing monitoring and reassessment, including documentation of changes to the system, its environment, and the security impact of changes; report the security state of the system to appropriate authorities.

An effective process of continuous monitoring and assessment can entail significant effort and cost, both in the form of automated tools and in the need for human expertise, to carry out functions such as analyzing event logs, conducting frequent security testing, and conducting regular configuration audits. These must be planned for and provided if the benefits of RMF are to be realized. An approach that is growing in popularity is to create a virtual computing environment that matches the actual system but is completely separate from it, commonly labeled a "sandbox." This allows software to be hosted and thoroughly tested before it is deployed in the production environment and to be retested as necessary in response to newly discovered vulnerabilities without disrupting operations. Continuous monitoring is also helped by the availability of increasingly sophisticated tools for analyzing network traffic patterns, event logs, and other indicators. A sandbox can be used in conjunction with a honeypot or honeynet to perform forensics such as identifying an attacker.

NIST and the National Security Agency (NSA) play key roles in enhancing cybersecurity for both the public and private sectors, including guidance that covers many aspects of implementing layered defense. NIST releases a series of FIPS publications [31]; two widely used examples are FIPS Publication 191, *Guidelines for the Analysis of Local Area Network Security*, and FIPS Publication 197, *Advanced Encryption Standard (AES).* Another excellent security reference is NIST Special Publication 800-27, *Engineering Principles for Information Technology Security (A Baseline for Achieving Security)* [32]. NSA supports primarily Government system security in a wide range of areas from cryptographic technologies and devices to recommended system configurations. A complete compendium of policies, directives, guidelines, standards, and other documents dealing with cybersecurity would be very large. The documents identified in this paragraph contain additional references that can help the reader find specialized information about particular topics.

Since an important element of implementing security is the use of products that are trusted to exhibit certain secure attributes and to be free from vulnerabilities, a method of evaluating and certifying such products has been key to cybersecurity for a long time. The primary approach today is the Common Criteria for Information

Technology Security Evaluation, usually shortened to Common Criteria or simply CC, which is based on the international ISO/IEC 15408 standard [33]. The CC structure allows security requirements for an item to be specified in a way that allows a product to be tested against them. Security requirements for a class of devices are captured in a Protection Profile (PP), and one or more PPs are used in compiling a Security Target (ST) that defines the security properties of a product and a set of Security Functional Requirements (SFRs). PPs are oriented toward allowing users to define their needs, while STs allow providers to spell out the security their products provide. The CC standard provides a catalog of SFRs as well as Security Assurance Requirements (SARs) that deal with product development and testing.

The most common use of CC is to assign an Evaluation Assurance Level (EAL) from 1 to 7 to a product. Essentially, an EAL corresponds to a set of SFRs, and the higher levels represent more complete (and more expensive) testing and validation. The highest levels can consume so much time and money that products are either not economically feasible to evaluate or approaching obsolescence when the process is complete, which is one of the major criticisms of CC. Government agencies typically mandate an EAL for a component of a secure system based on the sensitivity of the information being protected and the nature of the threat [34]. For example, information whose compromise would cause some damage (roughly equivalent to a classification of Confidential or Company Private) and that faces a threat characterized as unsophisticated but willing to take risks (perhaps a basic hacker) warrants an EAL of 2. Information whose compromise would cause exceptionally severe damage (e.g., a classification of Top Secret or key intellectual property) with a sophisticated adversary who has reasonable resources and will take significant risks (perhaps a major terrorist organization) calls for an EAL of 6, which many commercial products cannot meet. Security architects working with highly sensitive information are likely to have to deal with this aspect of system design.

No product evaluation is perfect. As an example of emerging threats, Yang described a "hardware Trojan" that can be inserted in a processor chip that is undetectable by currently known methods and that can give an attacker complete access to a computer's operating system [35]. Protection against such a threat literally demands a completely secure and trusted chip fabrication process that prevents an attacker from gaining physical access to parts.

MBSE can support system security evaluation and accreditation in a number of ways, as summarized in the following paragraphs.

- A formal architecture model allows capture, maintenance, and visualization of the configuration baseline and processes or behaviors of a system or system of systems. Artifacts exported from such a model can be focused on specific aspects such as security features and functions and can effectively provide the information needed by security testers, analysts, and accreditation authorities. For example, architecture data can support planning of effective vulnerability assessments and penetration testing.

- An architecture model allows rigorous and auditable flowdown of requirements, including security requirements, to components, interfaces, and processes, to ensure all requirements have been accounted for in system design and to provide a basis for requirements verification and validation.
- As service-oriented architecture (SOA) becomes the preferred approach to implementing large information systems and enterprises, the power of architecture modeling in defining services and, especially, service interfaces is invaluable in system accreditation involving security services, both for internal services and for those a system provides or consumes as a participant in a larger enterprise.
- An architecture model provides a basis for configuration management, data management, and change management. For example, a model captures dependencies among system elements to support defining required regression testing following a design change. The model also serves as a searchable repository of system and component information, including functions such as archiving data from previous security testing to keep subsequent testing to the feasible minimum.
- Modern system architecture modeling tools allow simulation of behavioral diagrams to visualize the processes that they represent. This "executable architecture" technique is valuable in assessing the operation of security controls.

10.7 SOA, Web, and Cloud Security

Any situation in which sensitive information is transmitted outside a secure environment raises cybersecurity concerns. The two most prominent cases involve the Web, which essentially means the Internet, and the use of Cloud Computing. As discussed in Chap. 7, Service-Oriented Architecture is predicated on allowing multiple participants in an enterprise, who may come and go unpredictably and who use multiple information technologies, to interact in implementing business processes. However, proliferating access to data and resources inevitably creates security vulnerabilities. In addition to all the other cybersecurity measures discussed in this chapter, a SOA needs robust mechanisms for assuring confidentiality, integrity, and access for the services that are exposed and consumed by enterprise participants. Because Web services are the overwhelming choice today for SOA interactions, they are the focus of efforts to achieve security.

For a long time, the primary mechanism for protecting SOAP messages has been HTTPS, the secure version of the Hypertext Transmission Protocol. This encrypts an entire message and is adequately secure for a point-to-point transaction. However, Web traffic increasingly includes intermediate nodes that must read parts of a message in order to perform their functions and therefore, with HTTPS, must decode the entire message. With the sensitive content now exposed in multiple places, the level of security risk goes up significantly. The inflexibility of HTTPS and the overall need to move to higher QoP have motivated the creation of the WS-Security Framework, anchored in the WS-Security standard. Repeating the list from Chap. 7, the WS-Security Framework deals with identification, authentication, authorization,

confidentiality, and integrity and includes WS-Security, WS-Security Policy, WS-Trust, WS-Secure Conversation, WS-Federation, eXtensible Access Control Markup Language (XACML), eXtensible Rights Markup Language (XRML), XML Key Management Specification (XKMS), XML-Signature, XML-Encryption, Security Assertions Markup Language (SAML), .NET Passport, Secure Sockets Layer (SSL), and WSI-Basic Security Profile.

WS-Security defines language elements for security metadata, including header blocks, usernames, passwords, and tokens. It provides an overall means of associating security tokens with messages, incorporates security features in the header of a SOAP message, and provides integrity (anti-tamper), confidentiality (encryption), and various authentication and authorization mechanisms, including proof-of-origin certification. WS-Security allows selective encryption/decryption of message elements. In conjunction with WS-Security message encryption, a Kerberos or PKI service is commonly used to distribute session keys so that participants do not have to create, store, and protect large amounts of key material.

Working together, the standards and protocols of the WS-Security Framework provide the tools to implement cybersecurity concepts and requirements for the exchange of services in a SOA. Cybersecurity protective measures implemented at the nodes of the enterprise must be complemented by those implemented for the enterprise as a whole. When service models other than Web services are used, e.g., to meet performance requirements, equivalent mechanisms for authentication, data integrity and confidentiality, and access control must be provided. Table 10.1 lists organizations that are concerned with security for Web applications.

Additional methods used to improve Internet security include:

- *Transport Layer Security (TLS) (formerly called Secure Sockets Layer (SSL))* – a session-based protocol operating at the Transport Layer (see Chap. 9), especially for secure access to Web applications; TLS is also used for VPNs.
- *Secure Hypertext Protocol (S-HTTP, not to be confused with https)* – connectionless-oriented protocol that encapsulates data after session security has been negotiated; this approach is not widely used.
- *IPSEC* – operates at the Network Layer (see Chap. 9) to provide encryption and authentication; this is the primary VPN protocol.
- *Multiprotocol Label Switching (MPLS)* – fast packet forwarding using labels inserted between Layer 2 and 3 headers in a packet; it is protocol-independent and scalable and provides Quality of Service (QoS) with multiple Classes of Service (CoS) and Layer 3 VPN tunneling.
- *Secure Shell (SSH2)* – secure remote access using an encrypted tunnel between an SSH Client and an SSH Server; it can authenticate a Client to the Server (note: SSH1 is used but has exploitable vulnerabilities).
- *Wireless Transport Layer Security (WTLS)* – security services for the Wireless Application Protocol (WAP); classes of security include:
  - Class 1 – anonymous authentication
  - Class 2 – server authentication only

– Class 3 – client-server authentication using Service Set Identifiers (SSID) and Wired Equivalent Privacy (WEP) keys or a more secure encryption such as WAP2

Cloud security has all the challenges of other secure architectures plus the facts that data and other resources are not under the physical control of the owning organization and that those resources are shared with other cloud clients. Attackers seek to exploit vulnerabilities in the virtualization technologies used in a cloud, in access control mechanisms, and elsewhere. One example is called a Side-Channel Attack, which has several variations, all aimed at hijacking a cloud client's account or files. Another attack method is called Hyperjacking and is based on compromising the hypervisor that creates and manages virtual environments. Still another is a Virtual Machine Escape in which an attacker breaks out of a virtual machine to attack the underlying hosting environment.

The Cloud Security Alliance [36] is a nonprofit organization that seeks to promote best practices and education for improved cloud security. The following are some of the considerations involved in achieving an acceptable risk level when employing cloud services.

- It is incumbent on the Cloud Service Provider (CSP) to provide adequate assurance that personnel with access to resources and data are trustworthy, and clients should obtain explicit proof of this. Typical measures include security screening and intensive education of staff and access controls to prevent support personnel from reaching client data.
- Similarly, a CSP must provide physical security to ensure a data center is difficult or impossible to penetrate. Reliable electrical power, for example, using uninterruptible power supplies (UPSs), and environmental controls are also important to ensure against data loss.
- In any secure system, data at rest should be encrypted. This is especially important in a cloud.
- Access controls should be controlled only by the cloud client or data owner. It's highly desirable that the CSP support fine-grained user privileges so that access can be finely matched to individual users. The CSP should provide and maintain logging of all access attempts.
- The CSP must have an adequate plan for data backup and recovery and for business continuity.
- Encryption is important to protect cloud transactions over open networks, for data protection and for security of user data, logs, and other sensitive information. Advanced encryption algorithms used in cloud environments include various types of Attribute-Based Encryption (ABE) in which a set of attributes of the ciphertext and of the user's secret key must match before decryption can be performed.
- Compliance with applicable laws and regulations may be different in a cloud environment than with a system controlled by the using organization. An example is the approach used for data locating. A cloud services client must be aware of any such differences and take steps to ensure compliance.

10.8 Secure Architecture Modeling

Representing security features and functions in an architecture model is not fundamentally different from any other architecting task. There is not yet a UML/SysML profile for security analogous to SoaML, but stereotypes, design patterns, and other elements tailored to security can be used in a very similar fashion to SOA modeling as discussed in Chap. 7.

10.8.1 Security in the MBSAP Methodology

The following are important aspects of embedding security in a system architecture, following the MBSAP sequence of model development.

Requirements Security requirements, as discussed earlier in this chapter, are part of the system requirements baseline and may include FRs such as specific security controls and associated processes and NFRs such as a requirement for single sign-on or for strong password enforcement.

Operational Viewpoint System-level security features and functions are modeled first in the OV.

- Major groupings of security resources may be modeled as separate Domains. An example would be a BSS like the one shown in Fig. 10.7. Alternatively, they can be included as Subdomains of a Domain like Computing Infrastructure. Responsibilities, Values, and Operations should be captured as for any other structural element.
- A secure system should have Use Cases (and may profitably use Abuse Cases) with specifications and behavioral diagrams for security-related processes such as malware detection and user authentication. It may be useful to create an SMD showing security states and the transitions among them.
- Foundation Classes in the CDM commonly have Values and Operations associated with security; e.g., defining a Sensitivity or Classification Value ensures that all System Classes inheriting from the Foundation Class will have this key attribute.
- Any system and enterprise security services used should be documented in the services taxonomy and modeled as described in Chap. 7.
- The Contextual Perspective is a logical place to file information about the cybersecurity environment, RI/RA results, and other factors bearing on the secure solution.

Logical/Functional Viewpoint As with any other architecture aspects, the high-level structures, behaviors, data, and services from the OV are detailed to create a functional design in the LV.

- Security design details and processes are modeled in structural and behavioral diagrams just like any other. Figure 10.7 showed a structural example, and

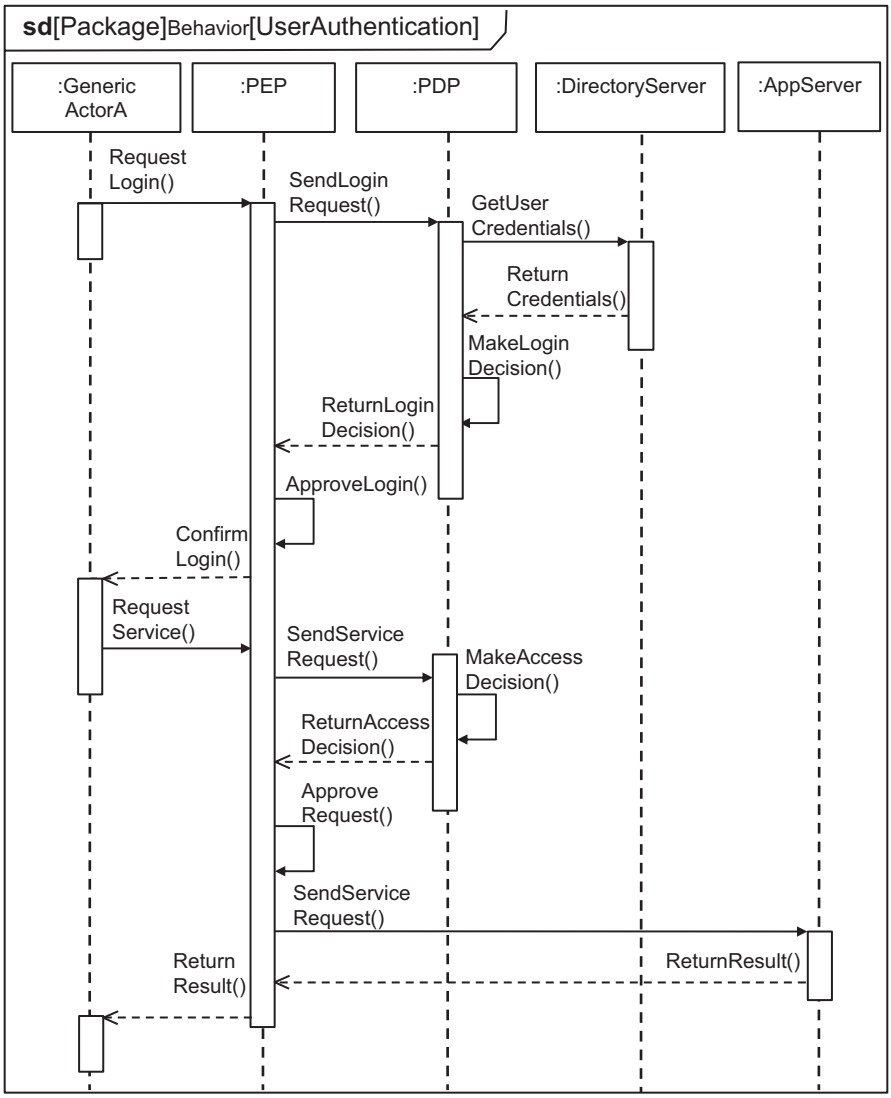


Fig. 10.10 Sequence diagram showing a user login and access to services

Fig. 10.10 shows a typical behavior, in this case a user login and access to an application. The user's login request is processed by the PEP based on current policy and the permissions and other attributes provided by a Directory Server (e.g., Active Directory), followed by a request to the application server and the return of a result. Many security controls are associated with interfaces and should be captured in interface specifications.

- Security requirements are flowed down from the OV and allocated to system elements. Security analysis and testing to demonstrate that the system achieves an acceptable risk level can be modeled using `<<VerifiedBy>>` Dependencies.
- The LDM may contain specific security data entities, and other data may have security-related Values and Operations.
- Domain-level security services from the OV should be further decomposed as appropriate and mapped to system elements such as processors, network devices, storage, and interfaces.
- Tailored artifacts such as a ports and protocols matrix are typically needed for system certification and may be generated from the model.
- The Contextual Perspective might well contain documentation of the rationale for the security design, special considerations of C&A or other authorization to operate processes, and anything else that will assist with long term-security maintenance and upgrading.

The participants in the behavior diagrammed in Fig. 10.7 all have security characteristics. Users have attributes and privileges that determine their ability to access the system and its content. The PEP and PDP are part of the BSS and have been described earlier. The Directory Server has the data and functions to securely manage user accounts. Finally, the Application Server has both Endpoint and Application Layer security controls as described under Defense in Depth.

Physical Viewpoint As with all other system elements, the PV documents the security products used and other implementation details. This could well include EALs for critical components.

Others Diagram annotations, linked files, and other documentation are often simple and effective ways to model and communicate a security design. Particular attention should be paid to information of interest to cybersecurity stakeholders, including accreditation or other approval authorities and the SYSADS/NETADs who are responsible for maintaining the system's security posture. Much of this belongs in a Security-Focused Viewpoint.

10.8.2 Security-Focused Viewpoint

A Security-Focused Viewpoint is essential for any system that deals with sensitive information, especially if a formal process such as C&A is required to obtain authority to operate. This is likely to be a large set of information, including graphics showing the placement of protective measures, a System Security Architecture document, documents defining security rules and procedures, and so forth. One approach is to take the evidence package prepared to support testing and accreditation and supplement it with any materials needed to convey the security design to program managers, other decision-makers, and system users. It is also useful to annotate the primary artifacts of the PV to show where security functions, security service interfaces, enclave boundaries and cross-domain information exchanges,

encryption, trusted software, and other design elements are implemented. The Security Architect or Security Team Lead is normally responsible for compiling and maintaining this Viewpoint.

10.9 Secure Software Development

Reported data on successful cyberattacks shows overwhelmingly the consequences of software vulnerabilities. Many of these can be traced to poor coding practices and failure to implement well-known protective features and functions. System architects and engineers dealing with secure systems have a critical responsibility to ensure that software, whether purchased, newly developed, or modified, has been built and tested using best practices to eliminate security flaws.

10.9.1 Secure Software Development Life Cycle (SSLDC)

Figure 10.11 shows schematically how the phases of a software development effort must be matched with security processes to create a Secure Software Development Life Cycle (SSDLC). The figure also lists some typical activities such as choice of security controls and other risk treatments. Both testing during software development and system testing are key elements of the SSDLC. Ultimately, the maximum feasible level of protection is achieved by a combination of:

- Deploying software that is inherently hard to attack because it is free of exploitable flaws and has any discovered vulnerabilities promptly patched
- Deploying software in a robust layered defense environment that minimizes the chances of an attacker gaining access to it

A representative list of secure software development principles would include the following. The goals are better code, less vulnerable code, an overall smaller system attack surface for improved cybersecurity, and lower total cost. Systems engineers and architects should ensure that these or their equivalents are implemented in the software process.

- *Include Abuse Cases in SW Requirements/Use Cases* – modeling the behaviors associated with potential attacks and system malfunctions helps ensure high-quality RI/RA and define an effective set of risk treatments.
- *Include security risk analysis in requirements, architecture, and design* – this is an integral part of the principle of building in security from day one of a software effort.
- *Include testable security functions and features in requirements* – this is the fundamental principle of ensuring that security requirements are part of a comprehensive and balanced security baseline.

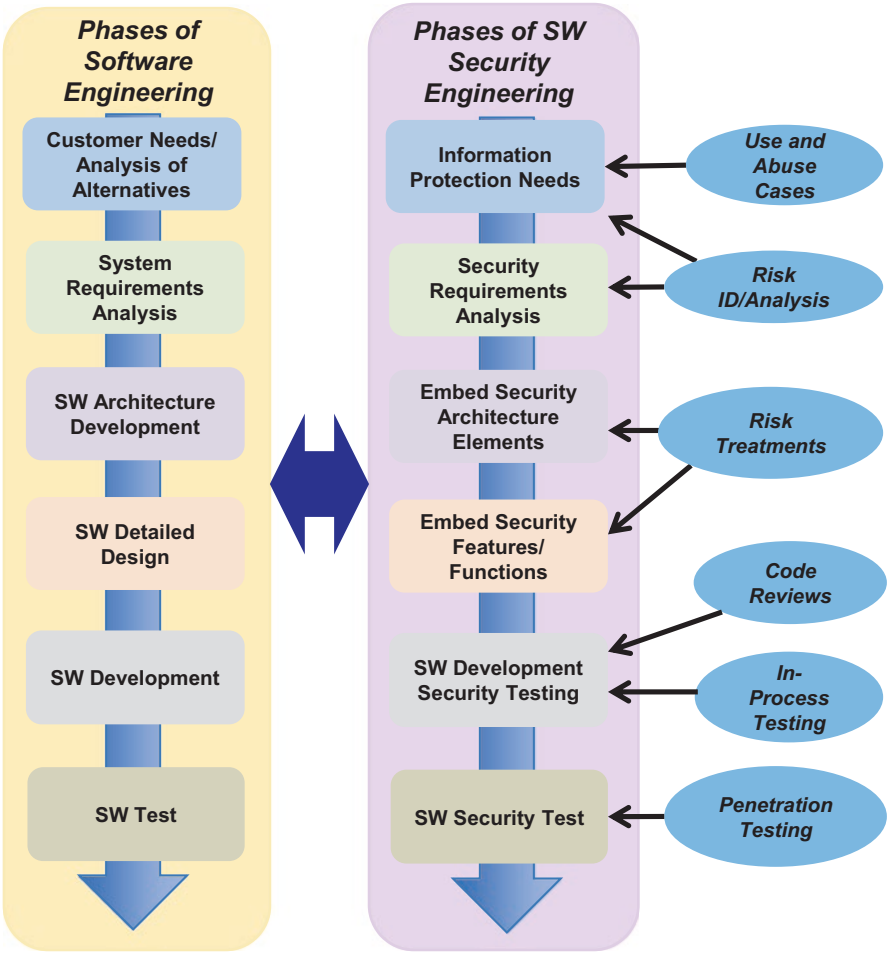


Fig. 10.11 Parallel phases of software development and secure software engineering

- *Include security functions and features in the design* – this is the remainder of the parallel secure software flow in Fig. 10.11; examples of these features are strong input checking, strong authentication, audit logs (which should be signed), code that is free of known vulnerabilities and signed to prevent tampering, privacy, and prevention of stack and buffer overflows.
- *Include vulnerability detection in design and code reviews* – the key point is to make software security evaluation an integral part of the development process.
- *Include risk-based security vulnerability testing and scanning during development and in system test* – it's important to use tools to scan for errors and vulnerabilities, and a wide variety of effective testing tools, many of them free, is available; see the next section for more details.

- *Perform penetration testing during system test and operations* – despite the best efforts to find and fix flaws during software development, it's vital that the final design be thoroughly tested to catch the vulnerabilities that are likely to remain; this continues through the system's operational life.
- *Implement software on secure, hardened servers with no access by programmers to the production environment* – an important aspect of the software development and test environment is that it be completely isolated from the operational system on which the software will be deployed.
- *Retest regularly for security and patch any vulnerabilities discovered or reported* – this has been stressed several times; the reality is that software development (or maintenance) and vulnerability testing never end.

The IEEE Computer Society Center for Secure Design (CSD) publishes a list of Top 10 Security Flaws that can serve as a useful checklist for a secure software process. These get to a level of software engineering detail that is beyond the scope of this book, but they indicate the kind of concrete coding practices that are essential to achieve trustworthy software. Appendix H gives a brief summary of the current list, which is updated periodically. Known best coding practices can eliminate common vulnerabilities to attacks based on injection of spurious user inputs, buffer overflows, and others as described in Appendix G. For example, the attack surface associated with the user access behavior in Fig. 10.10 could be quite large if secure software development is not applied to the various system components involved in it.

10.9.2 Security Testing

A variety of testing methods has been developed for assessing the effectiveness of security controls and other safeguards and for probing a system or its individual components for flaws and vulnerabilities. This is a matter of importance both to the engineers and architects responsible for a secure system solution and to the engineering team in general. In a system development or modification program, security testing must be integrated with the overall system test effort. Once a system is delivered and in operation, the need for ongoing assessment as part of governance means that some level of security testing will be a routine occurrence.

In the most common situation where system hardware is primarily purchased off the shelf, security testing is largely a vendor responsibility aimed at establishing required levels of trust through CC testing or some equivalent process. EALs are commonly published as a key element of product specifications. System developers must employ trusted components in a fashion that does not compromise their secure attributes, and testing of the complete, integrated system should confirm that no hardware flaws have been introduced.

The main focus of security testing is on software because that's where most vulnerabilities arise. Testing should be an integral part of software development,

whether in an Agile methodology such as Sprint or in a more traditional approach. This simply means that security testing is incorporated along with all the other testing that is a natural element of software development. The software development team should include, or have easy access to, security software testing expertise. Appendix I summarizes the most important categories of software security testing and lists some commonly employed tools.

10.10 Cybersecurity for the Smart Microgrid

We can illustrate the application of the concepts and techniques covered in this chapter using our Community Smart Microgrid (SMG-A) example. We will make some assumptions about the security characteristics and requirements of the SMG and consider how a tailored cybersecurity solution can be defined that maintains system performance and reliability while providing adequate protection against cyberattacks. Electric power is a core element of national critical infrastructure and is of great and growing concern as a target of malicious actors. As in other uses of the E-X and SMG examples in this book, we will not attempt a comprehensive treatment, but will illustrate some key aspects of achieving an acceptable level of security risk.

First, we need to identify the critical assets of SMG-A that require protection. Such a list, in approximate order of sensitivity, would include:

- Equipment control processes and data – this is the most critical asset category since the most serious threat involves interrupting electric service by seizing control of power resources or causing damage.
- Current operational and status data – this includes levels and directions of power flows, technical data exchanges with a Primary Grid (PG), equipment status, and similar data.
- Customer data – this includes exploitable information such as names and addresses and credit cards used to pay bills.
- Business data – this includes data exchanged with PG service providers.

The first two asset categories are typically associated with the functions of a System Control and Data Acquisition (SCADA) system. It has been widely reported that existing SCADAs, which are ubiquitous in the power industry and were designed with little or no consideration of cyber threats, are especially vulnerable to attack. A related hazard is that an attacker who gains control of the SMG may cause unsafe operating conditions that threaten the lives and safety of the system's legitimate users.

Next, we must consider possible threat agents, i.e., attackers. We will address three kinds:

- Hackers and thrill seekers who, as noted earlier, are after perverse satisfaction simply by making mischief

- Terrorists and the Advanced Persistent Threat (APT) who are attempting to cause damage to the nation and its citizens
- Insiders who may be either malicious or simply careless and poorly trained

The vulnerabilities threat agents may seek to exploit could include the following:

- Software containing unpatched vulnerabilities, which could enable unauthorized actions, injection of false commands, exposure of sensitive data, and other attacks
- Inadequate malware protection, potentially allowing an attacker to:
  - Corrupt or steal ("exfiltrate") SMG data and software.
  - Gain privileged access to the system, even to the extent of taking control.
  - Infect the Primary Grid and other external entities.
  - Commit other attacks such as ransomware.
- Inadequate access control, which allows an attacker to penetrate the system to inflict damage, steal information, and command unsafe equipment operation
- Inadequate physical security that allows an attacker to access system hardware to alter configurations and settings, replace trusted components with malicious ones, override or disable security controls, and do many other things

Given this threat environment, Table 10.4 contains some representative content of a security policy or SECOPS for the SMG along with corresponding security

Table 10.4 Examples of security policy and requirements for the Smart Microgrid

Policy statements	Requirements
The system shall implement strong access controls to ensure Confidentiality of system content	• The system shall implement Mandatory Access Control • The system shall enforce multifactor authentication • The system shall enforce a strong password policy with automated strength checking • The system shall enforce strong user account management • The system shall implement Role-Based Access Control with fine-grained access to resources (Principle of Least Privilege)
The system shall implement a layered defense to maximize Confidentiality and Integrity of system content	• The system shall implement security controls at the system periphery, Microgrid LAN, Supervisory Control, application software, database, and Microgrid Device levels • [Specific controls at each level should be specified]
The system shall protect data at rest, in transit, and in use	• The system shall encrypt all data in storage • The system shall implement VPNs for all external data communications, including with the PG • The system shall employ application software that prevents unauthorized exposure of data

requirements. Both policy and requirements should be supported by a rigorous risk analysis as well as a clear statement of security policy endorsed by the management of the SMG and coordinated with the PG. For example, the SECOPS might specify overarching security policy such as:

- No more than one data breach per year of any kind and no more than one breach of critical operational content every 10 years
- Service outages not exceeding 4.4 h/year (operational availability of 99.95%) whether due to hostile actions or natural occurrences

Specific policies and requirements like those in Table 10.4 would then be defined to support these overall security outcomes.

The SMG is a relatively small system, and careful design is essential to meet the stringent security policy with acceptable cost and operational impact. While some of the requirements cited in Table 10.4 might appear excessive, the importance of resilient cybersecurity for the SMG means that they should be satisfied to the maximum possible extent. There are two information processing enclaves: the Supervisory Control computer and the controllers and other processing embedded in SMG devices. These correspond to the Supervisory and Device Levels in Fig. 5.6. Most security controls apply to the Supervisory Control computer, but the lower level must also be accounted for. The system users are the operational, technical, and business members of the SMG staff, and selected PG personnel also need access to certain SMG information. A representative set of security controls would include the following specific safeguards, and Appendix F gives further detail.

- A Unified Threat Management (UTM) Firewall protecting the Supervisory Control computer and providing packet inspection, IDS/IPS, DLP, and possibly other perimeter defense functions. A Web Application Firewall may be required, depending on what functionality is exposed via the Web Port. Use of one or more advanced anti-malware products that use techniques such as deep learning (artificial intelligence) is mandatory.
- A whitelist of approved external IP addresses allowed to interact with the system via the Web Port.
- Monitored input/output, including traffic logging and detection of suspicious behavior.
- Secure software procedures, including:
  - Use of software products with appropriate trust certifications, including both system software (operating system, utilities, and database management) and applications
  - Continuous monitoring of published software security patches with immediate installation
  - Periodic penetration testing of the SMG network
- VPNs for all external data communications.
- Prohibition of the use of vulnerable devices such as wireless connections and USB memory.

- An encrypted SMG LAN with dedicated IP addresses for all connected devices (static LAN topology).
- Access controls, including multifactor authentication and enforced strong passwords; since both the number of users and the number of protected processes and databases are small, an effective access control solution consists of:
  - A basic system-level account manager requiring two-factor authentication (e.g., password and physical token or biometric) to log on to the system
  - Logging of all successful and unsuccessful log-on attempts
  - Automatic password strength checking plus forced password changes every 30 days
  - Secondary password-protected access to individual functions and data records
- Logging and weekly analysis of system activities; when feasible, this can include additional information such as machine identifiers, equipment and user locations derived from GPS, and many other details that can be important in detecting malicious activities and in analyzing cybersecurity incidents.
- Annual security refresher training for all personnel with system access.
- Physical security, including a perimeter fence around the central facility, locking doors, security cameras, and locking enclosures for equipment. At the device level, all sensitive items should have anti-tamper enclosures.
- A plan for incident response (both natural events and cyberattacks) with provisions to restore or maintain electrical service with minimum outage time.

10.11 Summary

This chapter has briefly surveyed an exceptionally broad and complex topic. Both private and public sector systems and the information environments that support them are under serious and growing attack from many quarters, including personnel wrongly assumed to be loyal and honest and national entities interested in stealing national secrets and intellectual property or even causing harm to a population. Every aspect of architecture practice is implicated in achieving secure, safe, and reliable systems, from initial threat assessment and security requirements to the provision of all required protective mechanisms, to product selections and system security testing. Much of the detail of security architecture and design is best entrusted to experts, but the system architect requires a working knowledge of the problem, the available solutions, the state of the technology, and the policies and best practices involved.

Exercises

1. List some typical security threats to information-intensive systems such as (a) a major bank or other financial institution, (b) an online retailer, and (c) a transportation company such as a railroad.
2. Define the following elements of a cyberattack by a foreign criminal syndicate against a domestic manufacturer: Threat Agent, Threat, Asset, Vulnerability, Security Control, and Exposure.

3. List some of the major factors in a successful system security solution for a school district that must protect a wide range of personal information, test materials, and other sensitive data.
4. List the three categories of cybersecurity safeguards or preventive measures and give examples of each.
5. How would ABAC and RBAC apply to the behavior diagrammed in Fig. 10.10?
6. What are the two fundamental elements of a resilient and affordable cybersecurity solution?
7. Describe the basic steps of risk identification (RI) and risk analysis (RA) for a national telecommunications network.
8. Describe the primary differences between traditional Certification and Accreditation and the more recent Risk Management Framework, and identify the advantages of the latter.
9. Considering the secure architecture development flow in Fig. 10.5, identify some aspects of Governance that would apply to the Secure Architecture, Implementation, and Monitoring, Maintenance, and Updating phases of the process.
10. Why would security controls such as firewalls and intrusion detection systems be applied at multiple levels of a Defense-in-Depth structure?
11. What is the basic use of an Evaluation Assurance Level (EAL)?
12. Considering the Boundary Security System in Fig. 10.7 and the Domain structure of the E-X example in Figs. 4.5 and 4.6, where would the various components of the BSS be implemented in the E-X?
13. What are the factors that would lead to the choice of a secret key or public key encryption scheme for a particular system or enterprise?
14. List some user attributes that can be used in an Attribute-Based Access Control scheme.
15. What are some factors that make securing a cloud environment more challenging?
16. Considering the phases of a Secure Software Development Life Cycle as shown in Fig. 10.11, list some specific elements of Software Security Engineering that might be applied at each phase of the Software Engineering Flow.

Student Projects

Students should add cybersecurity content to their architecture models based on the nature of each system using the concepts and techniques presented in this chapter, including risk identification and analysis, application of layered defense, and considerations of initial and recurring system testing and approval to operate.

References

1. Verizon Enterprise Services (2017) Data Breach Investigations Report (DBIR). http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/. Accessed 30 May 2017
2. Veris Community (2014) VERIS framework. http://veriscommunity.net/veris-overview.html. Accessed 31 May 2017

3. Whitmore JJ (2001) A method for designing secure solutions. IBM Syst J 40(3):747–768
4. Hershey P, Silo C (2012) Procedure for detection of and response to distributed denial of service cyber attacks on complex enterprise systems. In Proceedings of 6th Annual International Systems Conference, Vancouver, 19–22 March 2012, p 85–90
5. NIST Computer Security Division (2006) Guide for developing security plans for federal information systems, NIST SP 800-18. rev 1. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-18r1.pdf. Accessed 31 My 2017
6. NIST Computer Security Division (2006) Minimum security requirements for federal information and information systems, FIPS 200. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.200.pdf. Accessed 31 May 2017
7. Snyder D et al (2015) Improving the cybersecurity of U.S. air force military systems throughout their life cycles. Rand Corporation Research Report, Santa Monica, CA
8. SANS (2016) The CIS critical security controls for effective cyber defense ver 6.1. https://www.sans.org/critical-security-controls, http://www.tenable.com/solutions/council-on-cyber-security-critical-security-controls. Accessed 30 May 2017
9. NIST Computer Security Division (2012) Guide for conducting risk assessments, SP 800-30. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf. Accessed 31 May 2017
10. Microsoft (2014) Threat modeling tool. https://www.microsoft.com/en-us/download/details.aspx?id=42518. Accessed 31 May 2017
11. MITRE Corp (2017) Common weakness enumeration. https://cwe.mitre.org. Accessed 31 May 2017
12. Chew E et al (2008) Performance measurement guide for information security, NIST SP 800-55. National Institute for Standards and Technology, Gaithersburg
13. Bowen P et al (2006) Information security handbook: a guide for managers, NIST SP 800-100. National Institute for Standards and Technology, Gaithersburg
14. Rozanski N, Woods E (2005) Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, New York
15. Kindervag J (2013) Market overview: network segmentation gateways, Q4 2013. Forrester Research, Cambridge
16. Jones R, Horowitz B (2012) A system-aware cyber security architecture. Syst Eng 15(2):225–240
17. Trusted Computing Group (2014) TPM library specification. https://trustedcomputinggroup.org/tpm-library-specification/. Accessed 31 May 2017
18. Neuman C, et al. (2005) The Kerberos network authentication service (V5). https://tools.ietf.org/html/rfc4120. Accessed 31 May 2017
19. NIST Computer Security Division (2001) Specification for the Advanced Encryption Standard (AES), FIPS 197. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Accessed 31 May 2017
20. Rivest R, Shamir A, Adleman L (1978) a method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
21. NIST Computer Security Division (2001) Security requirements for cryptographic modules, FIPS 140-2. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf. Accessed 31 May 2017
22. Priebe T et al (2007) Supporting attribute-based access control in authorization and authentication infrastructures with ontologies. J Software 2(1):27–38
23. McGraw RW (2014) Risk Adaptable Access Control (RAdAC). http://csrc.nist.gov/news_events/privilege-management-workshop/radac-Paper0001.pdf. Accessed 31 May 2017
24. OASIS (2004) SAML 2.0 profile of XACML. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml_profile-spec-cd-02.pdf. Accessed 31 May 2017
25. Keegan W (2014) Separation kernels enable rapid development of trustworthy systems. COTS J 16(2):26–29
26. Kaplan F (2016) Dark territory: the secret history of cyber war. Simon and Schuster, New York

27. National Computer Security Center (1987) A guide to understanding audit in trusted systems. https://fas.org/irp/nsa/rainbow/tg001.htm. Accessed 31 May 2017
28. Scarfone K, et al. (2008) Technical guide to information security testing and assessment, NIST SP 800-115. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf. Accessed 31 May 2017
29. National Vulnerability Database (2017) Guide for assessing security controls in federal information systems and organizations, NIST SP 800-53, rev 4. https://nvd.nist.gov/800-53/Rev4. Accessed 31 May 2017
30. NIST Computer Security Division (2014) Guide for applying the risk management framework to federal information systems, a security life cycle approach. http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf. Accessed 31 May 2017
31. NIST Computer Security Division (2014) Federal Information Processing Standards (FIPS). http://csrc.nist.gov. Accessed 31 May 2017
32. Stoneburner G, Hayden C, Feringa A (2004) Engineering principles for information technology security (A baseline for achieving security), rev A. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-27ra.pdf. Accessed 1 June 2017
33. Common Criteria Recognition Agreement (2017) Common Criteria, ver 3.1, rel 5. http://www.commoncriteriaportal.org/cc/. Accessed 31 May 2017
34. DISA (2004) Determining the appropriate evaluation assurance level for COTS cybersecurity and cybersecurity-enable products (white paper). Defense Information Systems Agency, Ft Meade
35. Yang K (2016) A 'demonically clever' backdoor. Michigan Engineer, Fall 2016
36. Cloud Security Alliance (2017) Cloud security research reports (multiple). https://cloudsecurityalliance.org/. Accessed 31 May 2017