3.0 SECURITY PRINCIPLES

3.1 Introduction

To aid in designing a secure information system, NIST compiled a set of engineering principles for system security. These principles provide a foundation upon which a more consistent and structured approach to the design, development, and implementation of IT security capabilities can be constructed.

While the primary focus of these principles is the implementation of technical controls, these principles highlight the fact that, to be effective, a system security design should also consider non-technical issues, such as policy, operational procedures, and user education and training.

The principles described here do not apply to all systems at all times. Yet, each principle should be carefully considered throughout the life-cycle of every system. Moreover, because of the constantly changing information system security environment, the principles identified are not considered to be a static, all-inclusive list. Instead, this document is an attempt to present in a logical fashion fundamental security principles that can be used in today's operational environments. As technology improves and security techniques are refined, additions, deletions, and refinement of these security principles will be required.

Each principle has two components. The first is a table that indicates where the principle should be applied during the system life-cycle. The second is an explanatory narrative further amplifying the principle.

The five life-cycle planning phases used are defined in the *Generally Accepted Principles and Practices for Securing Information Technology Systems*, SP 800-14:

- Initiation Phase
- Development/Acquisition Phase
- Implementation Phase
- Operation/Maintenance Phase
- Disposal Phase.

In an effort to associate each principle with the relevant life-cycle planning phase(s), a table similar to the example table below, Table 3-1, has been developed for each principle. The table identifies each life-cycle phase, and "check marks" are used to indicate if the principle should be considered or applied during the specified phase. One check "✓" signifies the principle can be used to support the life-cycle phase, and two checks "✓✓" signifies the principle is key to successful completion of the life-cycle phase.

Table 3-1 Example Life-cycle Applicability Table

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Principle X	✓✓	✓		✓	

For example, the table above indicates that Principle No. X *must* be considered and is key to the successful completion of the Initiation phase. Additionally, Principle No. X, *should* be considered and applied in support of the Development/Acquisition and the Operation/Maintenance phases.

3.2 System Life-Cycle Description

The following brief descriptions of each of the five phases of the system life-cycle are taken from *Generally Accepted Principles and Practices for Securing Information Technology Systems*, SP 800-14.

- **Initiation:** During the initiation phase, the need for a system is expressed and the purpose of the system is documented. Activities include conducting an impact assessment in accordance with FIPS-199 (http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf).
- **Development/Acquisition:** During this phase, the system is designed, purchased, programmed, developed, or otherwise constructed. This phase often consists of other defined cycles, such as the system development cycle or the acquisition cycle. Activities include determining security requirements, incorporating security requirements into specifications, and obtaining the system.
- **Implementation:** During implementation, the system is tested and installed or fielded. Activities include installing/turning on controls, security testing, certification, and accreditation.
- **Operation/Maintenance:** During this phase, the system performs its work. Typically, the system is also being modified by the addition of hardware and software and by numerous other events. Activities include security operations and administration, operational assurance, and audits and monitoring.
- **Disposal:** The disposal phase of the IT system life-cycle involves the disposition of information, hardware, and software. Activities include moving, archiving, discarding or destroying information and sanitizing the media.

3.3 IT Security Principles

The 33 IT security principles are grouped into the following 6 categories: Security Foundation, Risk Based, Ease of Use, Increase Resilience, Reduce Vulnerabilities, and Design with Network in Mind. This grouping has resulted in a renumbering of the principles with respect to the original version of SP 800-27. The previous numbering is indicated to assist those familiar with the principles using the previous numbering.

3.3.1 Security Foundation

Principle 1. Establish a sound security policy as the "foundation" for design.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓	✓	✓	✓

Discussion: A security policy is an important document to develop while designing an information system. The security policy begins with the organization's basic commitment to information security formulated as a general policy statement. The policy is then applied to all aspects of the system design or security solution. The policy identifies security goals (e.g., confidentiality, integrity, availability, accountability, and assurance) the system should support, and these goals guide the procedures, standards and controls used in the IT security architecture design. The policy also should require definition of critical assets, the perceived threat, and security-related roles and responsibilities.

Principle 2. Treat security as an integral part of the overall system design.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓	✓✓	✓✓	✓

Discussion: Security must be considered in information system design. Experience has shown it to be both difficult and costly to implement security measures properly and successfully after a system has been developed, so it should be integrated fully into the system life-cycle process. This includes establishing security policies, understanding the resulting security requirements, participating in the evaluation of security products, and finally in the engineering, design, implementation, and disposal of the system.

Principle 3. Clearly delineate the physical and logical security boundaries governed by associated security policies.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓	✓	✓	

Discussion: Information technology exists in physical and logical locations, and boundaries exist between these locations. An understanding of what is to be protected from external factors can help ensure adequate protective measures are applied where they will be most effective. Sometimes a boundary is defined by people, information, and information technology associated with one physical location. But this ignores the reality that, within a single location, many different security policies may be in place, some covering publicly accessible information and some covering sensitive unclassified information. Other times a boundary is defined by a security policy that governs a specific set of information and information technology that can cross physical boundaries. Further complicating the matter is that, many times, a single machine or server may house both public-access and sensitive unclassified information. As a result,

multiple security policies may apply to a single machine or within a single system. Therefore, when developing an information system, security boundaries must be considered and communicated in relevant system documentation and security policies.

Principle 4 (formerly 33). Ensure that developers are trained in how to develop secure software.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓	✓		

Discussion: It is unwise to assume that developers know how to develop secure software. Therefore, ensure that developers are adequately trained in the development of secure software before developing the system. This includes application of engineering disciplines to design, development, configuration control, and integration and testing.

3.3.2 Risk Based

Principle 5 (formerly 4). Reduce risk to an acceptable level.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓	✓✓	✓✓	✓✓

Discussion: Risk is defined as the combination of (1) the likelihood that a particular threat source will exercise (intentionally exploit or unintentionally trigger) a particular information system vulnerability and (2) the resulting adverse impact on organizational operations, organizational assets, or individuals should this occur. Previously, risk avoidance was a common IT security goal. That changed as the nature of the risk became better understood. Today, it is recognized that elimination of all risk is not cost-effective. A cost-benefit analysis should be conducted for each proposed control. In some cases, the benefits of a more secure system may not justify the direct and indirect costs. Benefits include more than just prevention of monetary loss; for example, controls may be essential for maintaining public trust and confidence. Direct costs include the cost of purchasing and installing a given technology; indirect costs include decreased system performance and additional training. The goal is to enhance mission/business capabilities by mitigating mission/business risk to an acceptable level. (Related Principle: 6)

Principle 6 (formerly 5). Assume that external systems are insecure.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓	✓✓	✓✓	✓

Discussion: The term information domain arises from the practice of partitioning information resources according to access control, need, and levels of protection required. Organizations implement specific measures to enforce this partitioning and to provide for the deliberate flow of authorized information between information domains. The boundary of an information domain represents the security perimeter for that domain.

An external domain is one that is not under your control. In general, external systems should be considered insecure. Until an external domain has been deemed "trusted," system engineers, architects, and IT specialists should presume the security measures of an external system are different than those of a trusted internal system and design the system security features accordingly.

Principle 7 (formerly 6). *Identify potential trade-offs between reducing risk and increased costs and decrease in other aspects of operational effectiveness.*

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓		✓✓	

Discussion: To meet stated security requirements, a systems designer, architect, or security practitioner will need to identify and address all competing operational needs. It may be necessary to modify or adjust (i.e., trade-off) security goals due to other operational requirements. In modifying or adjusting security goals, an acceptance of greater risk and cost may be inevitable. By identifying and addressing these trade-offs as early as possible, decision makers will have greater latitude and be able to achieve more effective systems. (Related Principle: 4)

Principle 8. Implement tailored system security measures to meet organizational security goals.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	✓

Discussion: In general, IT security measures are tailored according to an organization's unique needs. While numerous factors, such as the overriding mission requirements, and guidance, are to be considered, the fundamental issue is the protection of the mission or business from IT security-related, negative impacts. Because IT security needs are not uniform, system designers and security practitioners should consider the level of trust when connecting to other external networks and internal sub-domains. Recognizing the uniqueness of each system allows a layered security strategy to be used – implementing lower assurance solutions with lower costs to protect less critical systems and higher assurance solutions only at the most critical areas.

Principle 9 (formerly 26). Protect information while being processed, in transit, and in storage.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	✓

Discussion: The risk of unauthorized modification or destruction of data, disclosure of information, and denial of access to data while in transit should be considered along with the risks associated with data that is in storage or being processed. Therefore, system engineers, architects, and IT specialists should implement security measures to preserve, as needed, the integrity, confidentiality, and availability of data, including application software, while the information is being processed, in transit, and in storage.

Principle 10 (formerly 29). Consider custom products to achieve adequate security.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓	

Discussion: Designers should recognize that in some instances it will not be possible to meet security goals with systems constructed entirely from COTS products. In such instances, it will be necessary to augment COTS with non-COTS mechanisms.

Principle 11 (formerly 31). Protect against all likely classes of "attacks."

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓	✓

Discussion: In designing the security controls, multiple classes of "attacks" need to be considered. Those classes that result in unacceptable risk need to be mitigated. Examples of "attack" classes are: Passive monitoring, active network attacks, exploitation by insiders, attacks requiring physical access or proximity, and the insertion of backdoors and malicious code during software development and/or distribution.

3.3.3 Ease of Use

Principle 12 (formerly 18). Where possible, base security on open standards for portability and interoperability.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓		

Discussion: Most organizations depend significantly on distributed information systems to perform their mission or business. These systems distribute information both across their own organization and to other organizations. For security capabilities to be effective in such environments, security program designers should make every effort to incorporate interoperability and portability into all security measures, including hardware and software, and implementation practices.

Principle 13 (formerly 19). Use common language in developing security requirements.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓✓	✓✓		✓✓	

Discussion: The use of a common language when developing security requirements permits organizations to evaluate and compare security products and features evaluated in a common test environment. When a "common" evaluation process is based upon common requirements or criteria, a level of confidence can be established that ensures product security functions conform to an organization's security requirements. The Common Criteria provides a source of common expressions for common needs and supports a common assessment methodology.

Principle 14 (formerly 21). Design security to allow for regular adoption of new technology, including a secure and logical technology upgrade process.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓✓	✓	✓✓	

Discussion: As mission and business processes and the threat environment change, security requirements and technical protection methods must be updated. IT-related risks to the mission/business vary over time and undergo periodic assessment. Periodic assessment should be performed to enable system designers and managers to make informed risk management decisions on whether to accept or mitigate identified risks with changes or updates to the security capability. The lack of timely identification through consistent security solution re-evaluation and correction of evolving, applicable IT vulnerabilities results in false trust and increased risk.

Each security mechanism should be able to support migration to new technology or upgrade of new features without requiring an entire system redesign. The security design should be modular so that individual parts of the security design can be upgraded without the requirement to modify the entire system.

Principle 15 (formerly 27). Strive for operational ease of use.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	

Discussion: The more difficult it is to maintain and operate a security control, the less effective that control is likely to be. Therefore, security controls should be designed to be consistent with the concept of operations and with ease-of-use as an important consideration. The experience and expertise of administrators and users should be appropriate and proportional to the operation of the security control. An organization must invest the resources necessary to ensure system administrators and users are properly trained. Moreover, administrator and user training costs along with the life-cycle operational costs should be considered when determining the cost-effectiveness of the security control.

3.3.4 Increase Resilience

Principle 16 (formerly 7). Implement layered security (Ensure no single point of vulerability).

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	✓

Discussion: Security designs should consider a layered approach to address or protect against a specific threat or to reduce vulnerability. For example, the use of a packet-filtering router in conjunction with an application gateway and an intrusion detection system combine to increase the work-factor an attacker must expend to successfully attack the system. Adding good password controls and adequate user training improves the system's security posture even more.

The need for layered protections is especially important when commercial-off-the-shelf (COTS) products are used. Practical experience has shown that the current state-of-the-art for security quality in COTS products does not provide a high degree of protection against sophisticated attacks. It is possible to help mitigate this situation by placing several controls in series, requiring additional work by attackers to accomplish their goals.

Principle 17 (formerly 10). Design and operate an IT system to limit damage and to be resilient in response.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓		✓✓	

Discussion: Information systems should be resistant to attack, should limit damage, and should recover rapidly when attacks do occur. The principle suggested here recognizes the need for adequate protection technologies at all levels to ensure that any potential cyber attack will be countered effectively. There are vulnerabilities that cannot be fixed, those that have not yet been fixed, those that are not known, and those that could be fixed but are not (e.g., risky services allowed through firewalls) to allow increased operational capabilities. In addition to achieving a secure initial state, secure systems should have a well-defined status after failure, either to a secure failure state or via a recovery procedure to a known secure state. Organizations should establish detect and respond capabilities, manage single points of failure in their systems, and implement a reporting and response strategy. (Related Principle: 14)

Principle 18 (formerly 13). Provide assurance that the system is, and continues to be, resilient in the face of expected threats.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	✓

Discussion: Assurance is the grounds for confidence that a system meets its security expectations. These expectations can typically be summarized as providing sufficient resistance to both direct penetration and attempts to circumvent security controls. Good understanding of the threat environment, evaluation of requirement sets, hardware and software engineering disciplines, and product and system evaluations are primary measures used to achieve assurance. Additionally, the documentation of the specific and evolving threats is important in making timely adjustments in applied security and strategically supporting incremental security enhancements.

Principle 19 (formerly 14). Limit or contain vulnerabilities.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓✓	✓	✓	

Discussion: Design systems to limit or contain vulnerabilities. If a vulnerability does exist, damage can be limited or contained, allowing other information system elements to function properly. Limiting and containing insecurities also helps to focus response and reconstitution efforts to information system areas most in need. (Related Principle: 10)

Principle 20 (formerly 16). Isolate public access systems from mission critical resources (e.g., data, processes, etc.).

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓	

Discussion: While the trend toward shared infrastructure has considerable merit in many cases, it is not universally applicable. In cases where the sensitivity or criticality of the information is high, organizations may want to limit the number of systems on which that data is stored and isolate them, either physically or logically. Physical isolation may include ensuring that no physical connection exists between an organization's public access information resources and an organization's critical information. When implementing logical isolation solutions, layers of security services and mechanisms should be established between public systems and secure systems responsible for protecting mission critical resources. Security layers may include using network architecture designs such as demilitarized zones and screened subnets. Finally, system designers and administrators should enforce organizational security policies and procedures regarding use of public access systems.

Principle 21 (formerly 17). Use boundary mechanisms to separate computing systems and network infrastructures.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓✓	✓	✓✓	

Discussion: To control the flow of information and access across network boundaries in computing and communications infrastructures, and to enforce the proper separation of user groups, a suite of access control devices and accompanying access control policies should be used.

Determine the following for communications across network boundaries:

- What external interfaces are required
- Whether information is pushed or pulled
- What ports, protocols, and network services are required
- What requirements exist for system information exchanges; for example, trust relationships, database replication services, and domain name resolution processes.

Principle 22 (formerly 20). Design and implement audit mechanisms to detect unauthorized use and to support incident investigations.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓✓	✓	

Discussion: Organizations should monitor, record, and periodically review audit logs to identify unauthorized use and to ensure system resources are functioning properly. In some cases, organizations may be required to disclose information obtained through auditing mechanisms to appropriate third parties, including law enforcement authorities or Freedom of Information Act (FOIA) applicants. Many organizations have implemented consent to monitor policies which state that evidence of unauthorized use (e.g., audit trails) may be used to support administrative or criminal investigations.

Principle 23 (formerly 28). Develop and exercise contingency or disaster recovery procedures to ensure appropriate availability.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓	✓	✓✓	

Discussion: Continuity of operations plans or disaster recovery procedures address continuance of an organization's operation in the event of a disaster or prolonged service interruption that affects the organization's mission. Such plans should address an emergency response phase, a recovery phase, and a return to normal operation phase. Personnel responsibilities during an incident and available resources should be identified. In reality, contingency and disaster recovery plans do not address every possible scenario or assumption. Rather, it focuses on the events most likely to occur and identifies an acceptable method of recovery. Periodically, the plans and procedures should be exercised to ensure that they are effective and well understood.

3.3.5 Reduce Vulnerabilities

Principle 24 (formerly 9). Strive for simplicity.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	

Discussion: The more complex the mechanism, the more likely it may possess exploitable flaws. Simple mechanisms tend to have fewer exploitable flaws and require less maintenance. Further, because configuration management issues are simplified, updating or replacing a simple mechanism becomes a less intensive process.

Principle 25 (formerly 11). Minimize the system elements to be trusted.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓✓	

Discussion: Security measures include people, operations, and technology. Where technology is used, hardware, firmware, and software should be designed and implemented so that a minimum number of system elements need to be trusted in order to maintain protection. Further, to ensure cost-effective and timely certification of system security features, it is important to minimize the amount of software and hardware expected to provide the most secure functions for the system.

Principle 26 (formerly 24). Implement least privilege.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓	✓	✓✓	

Discussion:
The concept of limiting access, or "least privilege," is simply to provide no more authorizations than necessary to perform required functions. This is perhaps most often applied in the administration of the system. Its goal is to reduce risk by limiting the number of people with access to critical system security controls; i.e., controlling who is allowed to enable or disable system security features or change the privileges of users or programs. Best practice suggests it is better to have several administrators with limited access to security resources rather than one person with "super user" permissions. .

Consideration should be given to implementing role-based access controls for various aspects of system use, not only administration. The system security policy can identify and define the various roles of users or processes. Each role is assigned those permissions needed to perform its functions. Each permission specifies a permitted access to a particular resource (such as "read" and "write" access to a specified file or directory, "connect" access to a given host and port, etc.). Unless a permission is granted explicitly, the user or process should not be able to access the protected resource. Additionally, identify the roles/responsibilities that, for security purposes, should remain separate, this is commonly termed “separation of duties”.

Principle 27 (formerly 25). Do not implement unnecessary security mechanisms.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓✓	✓	✓

Discussion: Every security mechanism should support a security service or set of services, and every security service should support one or more security goals. Extra measures should not be implemented if they do not support a recognized service or security goal. Such mechanisms could add unneeded complexity to the system and are potential sources of additional vulnerabilities.

An example is file encryption supporting the access control service that in turn supports the goals of confidentiality and integrity by preventing unauthorized file access. If file encryption is a necessary part of accomplishing the goals, then the mechanism is appropriate. However, if these security goals are adequately supported without inclusion of file encryption, then that mechanism would be an unneeded system complexity.

Principle 28 (formerly 30). Ensure proper security in the shutdown or disposal of a system.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓		✓	✓✓

Discussion: Although a system may be powered down, critical information still resides on the system and could be retrieved by an unauthorized user or organization. Access to critical information systems must be controlled at all times.

At the end of a system's life-cycle, system designers should develop procedures to dispose of an information system's assets in a proper and secure fashion. Procedures must be implemented to ensure system hard drives, volatile memory, and other media are purged to an acceptable level and do not retain residual information.

Principle 29 (formerly 32). Identify and prevent common errors and vulnerabilities.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓✓	✓✓	✓	

Discussion: Many errors reoccur with disturbing regularity - errors such as buffer overflows, race conditions, format string errors, failing to check input for validity, and programs being given excessive privileges. Learning from the past will improve future results.

3.3.6 Design with Network in Mind

Principle 30 (formerly 12). Implement security through a combination of measures distributed physically and logically.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability		✓✓	✓	✓	✓

Discussion: Often, a single security service is achieved by cooperating elements existing on separate machines. For example, system authentication is typically accomplished using elements ranging from the user-interface on a workstation through the networking elements to an application on an authentication server. It is important to associate all elements with the security service they provide. These components are likely to be shared across systems to achieve security as infrastructure resources come under more senior budget and operational control.

Principle 31 (formerly 15). Formulate security measures to address multiple overlapping information domains.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓✓	✓	✓	

Discussion: An information domain is a set of active entities (person, process, or devices) and their data objects. A single information domain may be subject to multiple security policies. A single security policy may span multiple information domains. An efficient and cost effective security capability should be able to enforce multiple security policies to protect multiple information domains without the need to separate physically the information and respective information systems processing the data. This principle argues for moving away from the traditional practice of creating separate LANs and infrastructures for various sensitivity levels (e.g., security classification or business function such as proposal development) and moving toward solutions that enable the use of common, shared, infrastructures with appropriate protections at the operating system, application, and workstation level.

Moreover, to accomplish missions and protect critical functions, government and private sector organizations have many types of information to safeguard. With this principle in mind, system engineers, architects, and IT specialists should develop a security capability that allows organizations with multiple levels of information sensitivity to achieve the basic security goals in an efficient manner.

Principle 32 (formerly 22). Authenticate users and processes to ensure appropriate access control decisions both within and across domains.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓	✓	✓✓	

Discussion: Authentication is the process where a system establishes the validity of a transmission, message, or a means of verifying the eligibility of an individual, process, or machine to carry out a desired action, thereby ensuring that security is not compromised by an untrusted source. It is essential that adequate authentication be achieved in order to implement security policies and achieve security goals. Additionally, level of trust is always an issue when dealing with cross-domain interactions. The solution is to establish an authentication policy and apply it to cross-domain interactions as required. Note: A user may have rights to use more than one name in multiple domains. Further, rights may differ among the domains, potentially leading to security policy violations.

Principle 33 (formerly 23). Use unique identities to ensure accountability.

	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
Applicability	✓	✓	✓	✓✓	

Discussion: An identity may represent an actual user or a process with its own identity, e.g., a program making a remote access. Unique identities are a required element in order to be able to:

- Maintain accountability and traceability of a user or process
- Assign specific rights to an individual user or process
- Provide for non-repudiation
- Enforce access control decisions
- Establish the identity of a peer in a secure communications path
- Prevent unauthorized users from masquerading as an authorized user.

3.4 Principle/Life-cycle Summary

Table 3-2 below summarizes the relationship between the 33 principles above and the life-cycles to which they apply.

Table 3-2: Principles versus Life-Cycles

	Life-Cycle Applicability				
Principle	Initiation	Devel/Acquis	Implement	Oper/Maint	Disposal
1	✓✓	✓	✓	✓	✓
2	✓✓	✓✓	✓✓	✓✓	✓
3	✓✓	✓✓	✓	✓	
4 (formerly 33)	✓✓	✓✓	✓		
5 (formerly 4)	✓✓	✓✓	✓✓	✓✓	✓✓
6 (formerly 5)	✓✓	✓✓	✓✓	✓✓	✓
7 (formerly 6)	✓✓	✓✓		✓✓	
8	✓	✓✓	✓	✓✓	✓
9 (formerly 26)	✓	✓✓	✓	✓✓	✓
10 (formerly 29)	✓	✓✓	✓	✓	
11 (formerly 31)	✓	✓✓	✓✓	✓	✓
12 (formerly 18)	✓	✓✓	✓		
13 (formerly 19)	✓✓	✓✓		✓✓	
14 (formerly 21)		✓✓	✓	✓✓	
15 (formerly 27)	✓	✓✓	✓	✓✓	
16 (formerly 7)	✓	✓✓	✓	✓✓	✓
17 (formerly 10)	✓	✓✓		✓✓	
18 (formerly 13)	✓	✓✓	✓	✓✓	✓
19 (formerly 14)		✓✓	✓	✓	
20 (formerly 16)	✓	✓✓	✓	✓	
21 (formerly 17)		✓✓	✓	✓✓	
22 (formerly 20)	✓	✓✓	✓✓	✓	
23 (formerly 28)	✓	✓	✓	✓✓	
24 (formerly 9)	✓	✓✓	✓	✓✓	
25 (formerly 11)	✓	✓✓	✓	✓✓	
26 (formerly 24)	✓	✓	✓	✓✓	
27 (formerly 25)	✓	✓	✓	✓	✓
28 (formerly 30)		✓		✓	✓✓
29 (formerly 32)		✓✓	✓✓	✓	
30 (formerly 12)		✓✓	✓	✓	✓
31 (formerly 15)	✓	✓✓	✓	✓	
32 (formerly 22)	✓	✓	✓	✓✓	
33 (formerly 23)	✓	✓	✓	✓✓	

TABLE E-6: STRATEGIC CYBER RESILIENCY DESIGN PRINCIPLES

STRATEGIC DESIGN PRINCIPLES	KEY IDEAS	RELATED DESIGN PRINCIPLES FROM OTHER DISCIPLINES
Focus on common critical assets.	Limited organizational and programmatic resources need to be applied where they can provide the greatest benefit. This results in a strategy of focusing first on assets which are both critical and common, then on those which are either critical or common.	**Security:** Inverse Modification Threshold. **Resilience Engineering:** Physical Redundancy, Layered Defense, Loose Coupling. **Survivability:** Failure Mode Reduction, Fail-Safe, Evolution.
Support agility and architect for adaptability.	Not only does the threat landscape change as adversaries evolve, so do technologies and the ways in which individuals and organizations use them. Both agility and adaptability are integral to the risk management strategy in response to the risk framing assumption that unforeseen changes will occur in the threat, technical, and operational environment through a system's lifespan.	**Security:** Secure Evolvability, Minimized Sharing, Reduced Complexity. **Resilience Engineering:** Reorganization, Human Backup, Inter-Node Interaction. **Survivability:** Mobility, Evolution.
Reduce attack surfaces.	A large attack surface is difficult to defend, requiring ongoing effort to monitor, analyze, and respond to anomalies. Reducing attack surfaces reduces ongoing protection scope costs and makes the adversary concentrate efforts on a small set of locations, resources, or environments that can be more effectively monitored and defended.	**Security:** Least Common Mechanism, Minimized Sharing, Reduced Complexity, Minimized Security Elements, Least Privilege, Predicate Permission. **Resilience Engineering:** Complexity Avoidance, Drift Correction. **Survivability:** Prevention, Failure Mode Reduction.
Assume compromised resources.	Systems and system components, ranging from chips to software modules to running services, can be compromised for extended periods without detection. In fact, some compromises may never be detected. Systems must remain capable of meeting performance and quality requirements, nonetheless.	**Security:** Trusted Components, Self-Reliant Trustworthiness, Trusted Communications Channels. *Incompatible with Security:* Hierarchical Protection. **Resilience Engineering:** Human Backup, Localized Capacity, Loose Coupling.
Expect adversaries to evolve.	Advanced cyber adversaries invest time, effort, and intelligence-gathering to improve existing and develop new TTPs. Adversaries evolve in response to opportunities offered by new technologies or uses of technology, as well as to the knowledge they gain about defender TTPs. In (increasingly short) time, the tools developed by advanced adversaries become available to less sophisticated adversaries. Therefore, systems and missions need to be resilient in the face of unexpected attacks.	**Security:** Trusted Communications Channels. **Resilience Engineering:** Reorganization, Drift Correction. **Survivability:** Evolution.

Strategic design principles are driven by an organization's risk management strategy and, in particular, by its risk framing. Risk framing includes, for example, assumptions about the threats the organization should be prepared for, the constraints on risk management decision-making (including which risk response alternatives are irrelevant), and organizational priorities and trade-offs.[109] From the standpoint of cyber resiliency, one way to express priorities is in terms of which cyber resiliency objectives are most important. Each strategic design principle supports achievement of one or more cyber resiliency objectives and relates to the design principles, concerns, or analysis processes associated with other specialty engineering disciplines. The relationships between strategic cyber resiliency design principles, risk framing, and analytic practices are indicated in Table E-7. Relationships between design principles and other cyber resiliency constructs are identified in Appendix E.6.

TABLE E-7: STRATEGIC DESIGN PRINCIPLES DRIVE ANALYSIS AND RELATE TO RISK MANAGEMENT

STRATEGIC DESIGN PRINCIPLES AND ANALYTIC PRACTICES	RISK FRAMING ELEMENTS OF RISK MANAGEMENT STRATEGY
Focus on common critical assets. ***Practices*:** Criticality Analysis, Business Impact Analysis (BIA), Mission Impact Analysis (MIA), Mission Thread Analysis	**Threat assumptions:** Conventional adversary; advanced adversary seeking path of least resistance. **Risk response constraints:** Limited programmatic resources. **Risk response priorities:** Anticipate, Withstand, Recover.
Support agility and architect for adaptability. ***Practices*:** Analysis of standards conformance, interoperability analysis, reusability analysis	**Threat assumptions**: Adaptive, agile adversary. **Risk response constraints**: Missions to be supported and mission needs can change rapidly. **Risk response priorities**: Recover, Adapt.
Reduce attack surfaces. ***Practices*:** Supply Chain Risk Management (SCRM) analysis, vulnerability and exposure analysis, Operations Security (OPSEC) analysis, Cyber-attack modeling and simulation	**Threat assumptions**: Conventional adversary; advanced adversary seeking path of least resistance. **Risk response constraints**: Limited operational resources to monitor and actively defend systems. **Risk response priorities**: Anticipate.
Assume compromised resources. ***Practices*:** Cascading failure analysis, Insider Threat analysis, Cyber-attack modeling and simulation	**Threat assumptions**: Advanced adversary. **Risk response constraints**: Ability to assure trustworthiness of system elements is limited. **Risk response priorities**: Anticipate, Withstand.
Expect adversaries to evolve. ***Practices*:** Adversary-driven Cyber Resiliency (ACR) analysis, Red Teaming	**Threat assumptions**: Advanced adversary; adversary can change TTPs and goals unpredictably. **Risk response priorities**: Anticipate, Adapt.

Sections E.5.1.1 through E.5.1.5 provide detailed descriptions of the five *strategic* cyber resiliency principles.

E.5.1.1 Focus on Common Critical Assets

A focus on critical assets (i.e., resources valued due to their importance to mission or business accomplishment)[110] is central to contingency planning, continuity of operations planning, and operational resilience, as well as to safety analysis. Critical assets can be identified using a variety of mission-oriented analysis techniques, including, for example: Mission Impact Analysis

[109] See [SP 800-39].

[110] Critical assets may also be referred to as High Value Assets (HVA) in accordance with [OMB M-19-03].

(MIA), Business Impact Analysis (BIA),[111] Functional Dependency Network Analysis (FDNA), Crown Jewels Analysis (CJA), and Mission Thread Analysis. Failure Modes, Criticality Analysis (FMECA), and Effects can, in some instances, reflect a safety-oriented approach.

Assets that are common to multiple missions or business functions are potential high value targets for adversaries either because those assets are critical or because their compromise increases the adversaries' options for lateral motion[112] or persistence [OMB M-19-03]. Once an asset is identified as critical or common, further analysis involves:

- Identifying how the asset is used in different operational contexts (e.g., normal operations, abnormal operations, crisis or emergency operations, failover). An asset that is common to multiple missions may be critical to one mission in one context but not in a second or critical to a second mission only in the second context.
- Determining which properties or attributes make the asset critical (e.g., correctness, non-observability, availability) or high value (e.g., providing access to a set of critical system elements, providing information which could be used in further malicious cyber activities) and what would constitute an acceptable (e.g., safe, secure) failure mode. Again, properties which are critical to one mission may be non-essential to another, and a failure mode which is acceptable from the standpoint of security may be unacceptable from the standpoint of safety.
- Determining which strategies to use to ensure critical properties, taking into consideration the different usage contexts and potential malicious cyber activities. Strategies for ensuring the correctness and non-observability properties include, for example, disabling noncritical functionality, restoration to default or known-good settings, and selectively isolating or disabling data flows to or from system components. Articulating trade-offs among critical properties and acceptable failure modes is central to effective risk management.

Based on the strategy or strategies that best fit a given type of asset, the most appropriate or relevant structural design principles can be determined.

This strategic design principle makes common infrastructures (e.g., networks), shared services (e.g., identity and access management services), and shared data repositories high priorities for the application of selected cyber resiliency techniques. It recognizes that the resources for risk mitigation are limited and enables systems engineers to focus resources where they will have the greatest potential impact on risk mitigation.

E.5.1.2 Support Agility and Architect for Adaptability

In Resilience Engineering, *agility* means "the effective response to opportunity and problem, within a mission" [Jackson07] [Sheard08]. In that context, resilience supports agility and counters brittleness. In the context of cyber resiliency, agility is the property of an infrastructure or a system which can be reconfigured, in which components can be reused or repurposed, and in which resources can be reallocated so that cyber defenders can define, select, and tailor cyber courses of action (CCoA) for a broad range of disruptions or malicious cyber activities. This

[111] See [SP 800-34].

[112] Lateral motion refers to an adversary's ability to move transitively from one system element to another system element or in a system-of-systems, from one constituent system to another constituent system.

strategy is consistent with the vision that the "infrastructure allows systems and missions to be reshaped nimbly to meet tactical goals or environment changes" [King12]. Agility enables the system and operational processes to incorporate new technologies and/or adapt to changing adversary capabilities.

Adaptability is the property of an architecture, a design, and/or an implementation which can accommodate changes to the threat model, mission or business functions, technologies, and systems without major programmatic impacts. A variety of strategies for agility and adaptability have been defined. These include modularity and controlled interfaces to support plug-and-play; externalization of rules and configuration data; and removal or disabling of unused components to reduce complexity. Application of this design principle early in the system life cycle can reduce sustainment costs and modernization efforts.

This design principle means that analyses of alternative architectures and designs need to search for sources of brittleness (e.g., reliance on a single operating system or communications channel; allowing single points of failure; reliance on proprietary interface standards; use of large and hard-to-analyze multi-function modules). Therefore, the analyses need to consider Redundancy, Adaptive Response, and Diversity, and the Coordinated Protection capabilities that enable cyber defenders to make effective use of these techniques. In addition, analyses need to consider where and how to use "cyber maneuver," or moving target defenses, and Deception. Finally, analyses need to consider where and how an architecture, design, or as-deployed system is bound to designated assumptions about the threat, operational, and/or technical environments.

E.5.1.3 Reduce Attack Surfaces

The term *attack surface* refers to the set of points on the boundary of a system, a system element, or an environment where an attacker can try to enter, cause an effect on, or extract data from that system, system element, or environment. The system's attack surface can be characterized as the accessible areas where weaknesses or deficiencies (including in hardware, software, and firmware system components) provide opportunities for adversaries to exploit vulnerabilities [SP 800-53], or as its exposure to reachable and exploitable vulnerabilities: any hardware, software, connection, data exchange, service, or removable media that might expose the system to potential threat access [DOD15]. Some uses of the term focus on externally exposed vulnerabilities (i.e., the attack surface of a system which connects to a network includes access control points for remote access). However, the assumption that an adversary will penetrate an organization's systems means that internal exposures (i.e., vulnerabilities which can be reached by lateral movement within a system or infrastructure) are also part of the attack surface. Conceptually, the term *attack surface* can also cover aspects of the development, operational, and maintenance environments that an adversary can reach and that could contain vulnerabilities. The supply chain for a system can also present additional attack surfaces. More broadly, an organization can be said to have an attack surface which includes its personnel and external users of organizational systems (if any) and its supply chain both for mission or business operations and for information and communications technology (ICT). To accommodate these broader interpretations of the term, the design principle refers to "attack surfaces."

This design principle is often used in conjunction with the Focus on common critical assets principle. Analysis of internal attack surfaces can reveal unplanned and unexpected paths to critical assets. It makes identification or discovery of attack surfaces a priority in system design

analyses,[113] as well as analyses of development, configuration, and maintenance environments (e.g., by considering how using free and open-source software (FOSS) or commercial off-the-shelf (COTS) products which cannot be tailored in those environments expands attack surfaces). It may be infeasible in some architectures (e.g., Internet of Things, bring-your-own-device) or procurement environments (e.g., limited supply chain), for which the Assume compromised resources principle is highly relevant.

As indicated in Table E-8, several alternative strategies for reducing an attack surface can be identified. These strategies are expressed by different controls in [SP 800-53] and apply different cyber resiliency techniques. In Table E-8, the **bolding** in the discussion of the control indicates how the control supports the strategy. These strategies can be reflected by different structural principles. For example, design decisions related to the Maximize transience and Change or disrupt the attack surface structural principles can reduce the duration of exposure; application of the Limit the need for trust principle can reduce exposure. While the controls in Table E-8 focus on attack surfaces within a system, the strategies apply more broadly to the attack surfaces of a mission or an organization. For example, Operations Security (OPSEC) can reduce exposure of the mission or organization to adversary reconnaissance. Supply chain protections can reduce the exposure of key components to tampering.

TABLE E-8: STRATEGIES FOR REDUCING ATTACK SURFACES[114]

STRATEGY	SECURITY CONTROL SUPPORTING STRATEGY	RELATED TECHNIQUES
Reduce the extent (area) of the attack surface.	Attack surface reduction includes, for example, **employing the concept of layered defenses; applying the principles of least privilege and least functionality; deprecating unsafe functions; applying secure software development practices including, for example, reducing the amount of code executing, reducing entry points available to unauthorized users, and eliminating application programming interfaces (APIs) that are vulnerable to cyber-attacks**. SA-15(5) DEVELOPMENT PROCESS, STANDARDS, AND TOOLS | ATTACK SURFACE REDUCTION [SP 800-53]	Coordinated Protection Privilege Restriction Realignment
Reduce the exposure (aperture or structural accessibility) of the attack surface.	Attack surface reduction includes, for example, **applying the principle of least privilege, employing layered defenses**, applying the principle of least functionality (i.e., restricting ports, protocols, functions, and services), deprecating unsafe functions, and eliminating application programming interfaces (APIs) that are vulnerable to cyber-attacks. SA-15(5) DEVELOPMENT PROCESS, STANDARDS, AND TOOLS | ATTACK SURFACE REDUCTION [SP 800-53]	Privilege Restriction Coordinated Protection
	Component isolation reduces the attack surface of organizational information systems. SC-7(20) BOUNDARY PROTECTION | DYNAMIC ISOLATION AND SEGREGATION [SP 800-53]	Adaptive Response Segmentation

[113] For example, [SP 800-53] control SA-11(7), Developer Security Testing | Attack Surface Reviews, calls for analysis of design and implementation changes.

[114] The security control supporting strategy includes examples and excerpts from relevant [SP 800-53] controls.

STRATEGY	SECURITY CONTROL SUPPORTING STRATEGY	RELATED TECHNIQUES
Reduce the duration (temporal accessibility) of attack surface exposure.	Mitigate risk from advanced persistent threats by significantly reducing the targeting capability of adversaries (i.e., **window of opportunity** and available attack surface) to initiate and complete cyber-attacks. SI-14 NON-PERSISTENCE [SP 800-53]	Non-Persistence

E.5.1.4 Assume Compromised Resources

A significant number of system architectures treat many, if not all, resources as non-malicious. This assumption is particularly prevalent in cyber-physical systems (CPS) and Internet of Things (IoT) architectures [Folk15]. However, systems and their components, ranging from chips to software modules to running services, can be compromised for extended periods without detection [DSB13]. In fact, some compromises may never be detected. Thus, the assumption that some system resources have been compromised is prudent. While the assumption that some resources cannot be trusted is well-established from the standpoint of security (i.e., the compromised resources cannot be trusted to follow established security policies), the concept of trustworthiness is broader. By compromising a resource, an adversary can affect its reliability, the ability to enforce other policies, or the safety of the larger system or environment of which the resource is a part [SP 1500-201, NIST16], or can use the resource in an attack on other systems.

This design principle implies the need for analysis of how the system architecture reduces the potential consequences of a successful compromise—in particular, the duration and degree of adversary-caused disruption and the speed and extent of malware propagation. An increasing number of modeling and simulation techniques support the analysis of the potential systemic consequences stemming from the compromise of a given resource or set of resources. Such analysis includes identifying different types or forms of systemic consequences (e.g., unreliable or unpredictable behavior of services, unreliable or unpredictable availability of capabilities, or data of indeterminate quality) and subsequently linking these systemic consequences to mission consequences (e.g., mission failure, safety failure) or organizational consequences (e.g., loss of trust or reputation).

E.5.1.5 Expect Adversaries to Evolve

Advanced cyber adversaries invest time, effort, and intelligence-gathering to improve existing TTPs and develop new TTPs. Adversaries evolve in response to opportunities offered by new technologies or uses of technology, as well as to the knowledge they gain about defender TTPs. In (increasingly short) time, the tools developed by advanced adversaries become available to less sophisticated adversaries. Therefore, systems and missions need to be resilient in the face of unexpected attacks. This design principle supports a risk management strategy which includes but goes beyond the common practice of searching for and seeking ways to remediate known vulnerabilities (or classes of vulnerabilities); a system which has been hardened in the sense of remediating known vulnerabilities will remain exposed to evolving adversaries.

This design principle implies the need for analyses in which the adversary perspective is explicitly represented by intelligent actors who can play the role of an adaptive or evolving

adversary. For implemented systems, such analyses are typically part of *red teaming* or *war gaming*. Analyses can use threat intelligence or repositories of attack patterns (e.g., ATT&CK [MITRE18], CAPEC [MITRE07]) to provide concrete examples, but care should be taken not to be constrained by those examples. Voice of the Adversary (VoA) is a design analysis technique in which one or more team members play the role of an adversary to critique alternatives by taking into consideration possible goals, behaviors, and cyber effects assuming varying degrees of system access or penetration. This type of design analysis can use models or taxonomies of adversary behaviors (e.g., the NTCTF [NSA18], cyber-attack life cycle or cyber kill chain models [Hutchins11], CAPEC [MITRE07] or ATT&CK [MITRE18] classes), and languages or taxonomies of cyber effects (e.g., [Temin10]).

This design principle also highlights the value of the Deception and Diversity techniques. Deception can cause adversaries to reveal their TTPs prematurely from the perspective of their cyber campaign plans, enabling defenders to develop countermeasures or defensive TTPs. Diversity can force an adversary to develop a wider range of TTPs to achieve the same objectives.

E.5.2 STRUCTURAL DESIGN PRINCIPLES

Structural cyber resiliency design principles guide and inform design and implementation decisions throughout the system life cycle. As indicated in Table E-9, many of the structural design principles are consistent with or leverage the design principles for security and/or resilience.[115] The first four design principles are closely related to protection strategies and security design principles and can be applied in mutually supportive ways. The next three design principles are closely related to design principles for resilience engineering and survivability. The next three design principles are driven by the concern for an operational environment (including cyber threats), which changes on an ongoing basis, and are closely related to design principles for evolvability. The final four principles are strongly driven by the need to manage the effects of malicious cyber activities, even when those activities are not observed. Descriptions of how structural design principles are applied, or could be applied, to a system-of-interest can help stakeholders understand how their concerns are being addressed.

TABLE E-9: STRUCTURAL CYBER RESILIENCY DESIGN PRINCIPLES

STRUCTURAL DESIGN PRINCIPLES	KEY IDEAS	RELATED DESIGN PRINCIPLES FROM OTHER DISCIPLINES
Limit the need for trust.	Limiting the number of system elements that need to be trusted (or the length of time an element needs to be trusted) reduces the level of effort needed for assurance, as well as for ongoing protection and monitoring.	**Security:** Least Common Mechanism, Trusted Components, Inverse Modification Threshold, Minimized Security Elements, Least Privilege, Predicate Permission, Self-Reliant Trustworthiness, Trusted Communications Channels. **Resilience Engineering:** Localized Capacity, Loose Coupling. **Survivability:** Prevention.

[115] The relationship between strategic and structural cyber resiliency design principles is presented in Table E-10.

STRUCTURAL DESIGN PRINCIPLES	KEY IDEAS	RELATED DESIGN PRINCIPLES FROM OTHER DISCIPLINES
Control visibility and use.	Controlling what can be discovered, observed, and used increases the effort needed by an adversary seeking to expand its foothold in or increase its impacts on systems containing cyber resources.	**Security:** Clear Abstraction, Least Common Mechanism, Least Privilege, Predicate Permission. **Resilience Engineering:** Localized Capacity, Loose Coupling. **Survivability:** Concealment, Hardness.
Contain and exclude behaviors.	Limiting what can be done and where actions can be taken reduces the possibility or extent of the spread of compromises or disruptions across components or services.	**Security:** Trusted Components, Least Privilege, Predicate Permission. **Resilience Engineering:** Localized Capacity, Loose Coupling. **Survivability:** Preemption, Hardness, Distribution.
Layer defenses and partition resources.	The combination of defense-in-depth and partitioning increases the effort required by an adversary to overcome multiple defenses.	**Security:** Modularity and Layering, Partially Ordered Dependencies, Minimized Sharing, Self-Reliant Trustworthiness, Secure Distributed Composition. **Resilience Engineering:** Layered Defense. **Survivability:** Hardness, Fail-Safe
Plan and manage diversity.	Diversity is a well-established resilience technique, removing single points of attack or failure. However, architectures and designs should take cost and manageability into consideration to avoid introducing new risks.	**Resilience Engineering:** Absorption, Repairability. **Survivability:** Heterogeneity.
Maintain redundancy.	Redundancy is key to many resilience strategies but can degrade over time as configurations are updated or connectivity changes.	**Resilience Engineering:** Absorption, Physical Redundancy, Functional Redundancy. **Survivability:** Redundancy, Margin.
Make resources location-versatile.	A resource bound to a single location (e.g., a service running only on a single hardware component, a database located in a single datacenter) can become a single point of failure and thus a high value target.	**Resilience Engineering:** Localized Capacity, Repairability. **Survivability:** Mobility, Avoidance, Distribution.
Leverage health and status data.	Health and status data can be useful in supporting situational awareness, indicating potentially suspicious behaviors, and predicting the need for adaptation to changing operational demands.	**Resilience Engineering:** Drift Correction, Inter-Node Interaction.
Maintain situational awareness.	Situational awareness, including awareness of possible performance trends and the emergence of anomalies, informs decisions about cyber courses of action to ensure mission completion.	**Resilience Engineering:** Drift Correction, Inter-Node Interaction.

STRUCTURAL DESIGN PRINCIPLES	KEY IDEAS	RELATED DESIGN PRINCIPLES FROM OTHER DISCIPLINES
Manage resources (risk-) adaptively.	Risk-adaptive management supports agility, providing supplemental risk mitigation throughout critical operations despite disruptions or outages of components.	**Security:** Trusted Components, Hierarchical Trust, Inverse Modification Threshold, Secure Distributed Composition, Trusted Communications Channels; Secure Defaults, Secure Failure and Recovery. **Resilience Engineering:** Reorganization, Repairability, Inter-Node Interaction. **Survivability:** Avoidance.
Maximize transience.	Use of transient system elements minimizes the duration of exposure to adversary activities, while periodically refreshing to a known (secure) state can expunge malware or corrupted data.	**Resilience Engineering:** Localized Capacity, Loose Coupling. **Survivability:** Avoidance.
Determine ongoing trustworthiness.	Periodic or ongoing verification and/or validation of the integrity or correctness of data or software can increase the effort needed by an adversary seeking to modify or fabricate data or functionality. Similarly, periodic or ongoing analysis of the behavior of individual users, system components, and services can increase suspicion, triggering responses such as closer monitoring, more restrictive privileges, or quarantine.	**Security:** Self-Reliant Trustworthiness, Continuous Protection, Secure Metadata Management, Self-Analysis, Accountability and Traceability. **Resilience Engineering:** Neutral State. **Survivability:** Fail-Safe.
Change or disrupt the attack surface.	Disruption of the attack surface can cause the adversary to waste resources, make incorrect assumptions about the system or the defender, or prematurely launch attacks or disclose information.	**Resilience Engineering:** Drift Correction **Survivability:** Mobility, Deterrence, Preemption, Avoidance.
Make the effects of deception and unpredictability user-transparent.	Deception and unpredictability can be highly effective techniques against an adversary, leading the adversary to reveal its presence or TTPs or to waste effort. However, when improperly applied, these techniques can also confuse users.	**Security:** Efficiently Mediated Access, Performance Security, Human Factored Security, Acceptable Security. **Survivability:** Concealment.

The selection of structural design principles is driven by strategic design principles, as shown in Table E-10.

TABLE E-10: STRATEGIC DESIGN PRINCIPLES DRIVE STRUCTURAL DESIGN PRINCIPLES

	STRATEGIC DESIGN PRINCIPLES				
STRUCTURAL DESIGN PRINCIPLES	**Focus on common critical assets**	**Support agility and architect for adaptability**	**Reduce attack surfaces**	**Assume compromised resources**	**Expect adversaries to evolve**
Limit the need for trust.			X	X	
Control visibility and use.	X		X	X	
Contain and exclude behaviors.	X			X	X
Layer defenses and partition resources.	X			X	
Plan and manage diversity.	X	X		X	
Maintain redundancy.	X	X		X	
Make resources location-versatile.	X	X			X
Leverage health and status data.	X	X		X	X
Maintain situational awareness.	X				X
Manage resources (risk-) adaptively.	X	X			X
Maximize transience.			X	X	X
Determine ongoing trustworthiness.	X			X	X
Change or disrupt the attack surface.			X	X	X
Make the effects of deception and unpredictability user-transparent.		X	X		

Structural design principles provide guidance for design decisions intended to reduce risk.[116] This guidance affects the selection and the application of cyber resiliency techniques. Table E-15 describes the relationship between structural design principles and cyber resiliency techniques. Table E-11 describes the application of structural design principles and the intended effects on risk.

TABLE E-11: STRUCTURAL DESIGN PRINCIPLES AND EFFECTS ON RISK

STRUCTURAL DESIGN PRINCIPLES	INTENDED EFFECTS ON RISK
Limit the need for trust.	Reduce likelihood of harm due to malice, error, or failure.
Control visibility and use.	Reduce likelihood of occurrence of adversarial events; reduce likelihood of harm due to malice, error, or failure.
Contain and exclude behaviors.	Reduce likelihood of occurrence of adversarial events; reduce likelihood of harm due to malice, error, or failure.
Layer defenses and partition resources.	Reduce likelihood of harm due to malice, error, or failure; reduce extent of harm.

[116] Harm to a cyber resource can take the form of degradation or disruption of functionality or performance; exfiltration or exposure of information; modification, corruption, or fabrication of information (including software, mission or business information, and configuration data); or usurpation or misuse of system resources. Unless otherwise specified, all forms of harm to systems containing cyber resources are addressed.

STRUCTURAL DESIGN PRINCIPLES	INTENDED EFFECTS ON RISK
Plan and manage diversity.	Reduce likelihood of harm due to malice, error, or failure; reduce extent of disruption.
Maintain redundancy.	Reduce likelihood of harm due to malice, error, or failure; reduce extent of disruption or degradation.
Make resources location-versatile.	Reduce likelihood of occurrence of adversarial events; reduce extent of disruption or degradation.
Leverage health and status data.	Reduce likelihood of harm due to malice, error, or failure by enabling response to changes in system state; reduce extent of harm by enabling detection of and response to indicators of damage.
Maintain situational awareness.	Reduce likelihood of harm due to malice, error, or failure by enabling response to indicators; reduce extent of harm by enabling detection of and response to indicators of damage.
Manage resources (risk-) adaptively.	Reduce likelihood of harm due to malice, error or failure by enabling response to changes in the operational environment; reduce extent of harm.
Maximize transience.	Reduce likelihood of occurrence by reducing the time during which an adverse event could occur; reduce likelihood of harm due to malice, error, or failure by reducing the time during which an event could result in harm.
Determine ongoing trustworthiness.	Reduce likelihood of harm due to corrupted, modified, or fabricated information by enabling untrustworthy information to be identified; reduce extent of harm by reducing the propagation of untrustworthy information.
Change or disrupt the attack surface.	Reduce likelihood of occurrence by removing the circumstances in which an adversarial event is feasible; reduce likelihood of harm due to adversarial events by making such events ineffective.
Make the effects of deception and unpredictability user-transparent.	Reduce the likelihood of occurrence of error; when Deception techniques are applied, reduce the likelihood of occurrence of adversarial events.

Sections E.5.2.1 through E.5.2.14 provide more detailed descriptions of the 14 structural cyber resiliency principles.

E.5.2.1 Limit the Need for Trust

Trustworthiness can be defined as an entity worthy of being trusted to fulfill whatever critical requirements may be needed for a component, subsystem, system, network, application, mission, enterprise, or other entity [Neumann04]. Trustworthiness has also been defined as the attribute of [an entity] that provides confidence to others of the qualifications, capabilities, and reliability of that entity to perform specific tasks and to fulfill assigned responsibilities [CNSSI 4009]. Assertions of trustworthiness (e.g., "this software can be relied upon to enforce the following security policies with a high level of confidence") are meaningless without some form of verification, validation, or demonstration (e.g., design analysis, testing). In the absence of some credible form of assurance (which can be costly and can be invalidated by changes in the system or the environment), assertions of trustworthiness constitute assumptions. Reducing the size of the set of trusted entities (whether individuals, software components, or hardware components) by minimizing assumptions about what is or can be trusted reduces the attack surface and lowers assurance costs.

Application of this design principle is most effective early in the system life cycle where the motivation of the Prevent/Avoid objective is clearest. When a system already exists, changes to the operational concept (consistent with the Transform objective) or to the system architecture (applying the Re-Architect objective and the Realignment technique) can increase costs. One approach to applying this design principle (using the Coordinated Protection and Privilege Restriction techniques) is through limitations on inheritance so that privileges or access rights associated with one class of system component are not automatically propagated to classes or instances created from the original one. While limitations on inheritance can increase the burden on developers or administrators initially, they can also reduce the complexity associated with multiple inheritance.

This design principle supports the strategic design principles of Reduce attack surfaces and Assume compromised resources. However, its application increases the difficulty of applying the Support agility and architect for adaptability strategic design principle. This design principle can also be used in conjunction with Determine ongoing trustworthiness; if a system element is assumed or required to have a given level of trustworthiness, some attestation mechanism is needed to verify that it has and continues to retain that trustworthiness level. Minimizing the number of elements with trustworthiness requirements reduces the level of effort involved in determining ongoing trustworthiness. Finally, this design principle can be used in conjunction with Plan and manage diversity; the managed use of multiple sources of system elements, services, or information can enable behavior or data quality to be validated by comparison.

E.5.2.2 Control Visibility and Use

Controlling visibility counters adversary attempts at reconnaissance from outside or within the system. Thus, the adversary must exert greater effort to identify potential targets, whether for exfiltration, modification, or disruption. Visibility of data can be controlled by such mechanisms as encryption, data hiding, or data obfuscation. Visibility of how some resources are used can also be controlled directly, for example, by adding chaff to network traffic. Visibility into the supply chain, development process, or system design can be limited via operations security (OPSEC), deception [Heckman15], and split or distributed design and manufacturing. Process obfuscation is an area of active research. An increasing number and variety of deception technologies, including for example, deception nets, can be applied at the system level.

Controlling use counters adversary activities and actions in the *Control*, *Execute*, and *Maintain* phases of the cyber-attack life cycle [MITRE18]. To limit visibility or to control use, access to system resources can be controlled from the perspectives of multiple security disciplines, including physical, logical (see the discussion of privileges below), and hybrid (e.g., physical locations in a geographically distributed system or in a complex, embedded system). Restrictions on access and use can be guided by information sensitivity, as in standard security practices. Restrictions can also be based on criticality (i.e., the importance to achieving mission objectives). While some resources can be determined to be mission-critical or mission-essential *a priori*, the criticality of other resources can change dynamically. For example, a resource which is vital to one phase of mission processing can become unimportant after that phase is completed.

Many systems or system components provide the capability to define and manage privileges associated with software, services, processes, hardware, communications channels, and individual users. Assignment of privileges ideally should reflect judgments of operational need (e.g., need-to-know, need-to-use) as well as trustworthiness. Restriction of privileges is well

established as a security design principle (i.e., least privilege). Privilege restrictions force adversaries to focus efforts on a restricted set of targets, which can be assured (in the case of software), validated (in the case of data), or monitored (in the case of individuals, processes, communications channels, and services). Non-Persistence and Segmentation can also limit visibility. Thus, this principle can be applied in conjunction with the Contain and exclude behaviors and Maximize transience principles.

E.5.2.3 Contain and Exclude Behaviors

The behavior of a system or system element, including what resources it uses, which systems or system elements it interacts with, or when it takes a given action, can vary based on many legitimate circumstances. However, analysis of the mission or business functions and the mission/business processes that carry out those missions and functions [SP 800-39] can identify some behaviors which are always unacceptable and others which are acceptable only under specific circumstances. Therefore, excluding behaviors prevents such behaviors from having undesirable consequences. Behaviors can be excluded *a priori* with varying degrees of assurance, from removing functionality to restricting functionality or use, with trade-offs between assurance and flexibility. For example, user activity outside of specific time windows can be precluded. In addition, behaviors can be interrupted based on ongoing monitoring when that monitoring provides a basis for suspicion.

Containing behaviors involves restricting the set of resources or system elements which can be affected by the behavior of a given system element. Such restriction can, but does not have to, involve a temporal aspect. Containment can be achieved *a priori*, via predefined privileges and segmentation. Alternately, or perhaps additionally, Adaptive Response and Dynamic Isolation can be applied. For example, a sandbox or deception environment can be dynamically created in response to suspicious behavior, and subsequent activities can be diverted there.

E.5.2.4 Layer Defenses and Partition Resources

Defense-in-depth is the integration of people, technology, and operations capabilities to establish variable barriers across multiple layers and missions [CNSSI 4009] and is a well-established security strategy. It describes security architectures constructed through the application of multiple mechanisms to create a series of barriers to prevent, delay, or deter an attack by an adversary [SP 800-160 v1]. Multiple mechanisms to achieve the same objective or to provide equivalent functionality can be used at a single layer (e.g., different COTS firewalls to separate zones in a DMZ) or at different layers (e.g., detection of suspicious behavior at the application, operating system, and network layers). To avoid inconsistencies which could result in errors or vulnerabilities, such (multiple) mechanisms should be managed consistently.

Layering of defenses restricts the adversary's movement vertically in a layered security architecture (i.e., a defense at one layer prevents a compromise at an adjacent layer from propagating). Partitioning (i.e., separating sets of resources into effectively separate systems) with controlled interfaces (e.g., cross domain solutions) between them restricts the lateral movement of the adversary. Partitioning can limit the adversary's visibility (see Control visibility and use). It can also serve to Contain and exclude behaviors. Partitioning can be based on administration and policy, as in security domains [SP 800-160 v1], or can be informed by the missions or business functions the system elements in the partition support. Partitions can be implemented physically or logically, at the network layer and within a platform (e.g., via hard or

soft partitioning). Partitioning may involve limiting resource-sharing or making fewer resources common. If resources are replicated, the Maintain redundancy principle should be applied.

E.5.2.5 Plan and Manage Diversity

Diversity (usually in conjunction with Redundancy [Sterbenz14]) is a well-established technique for improving system resilience [Sterbenz10, Höller15]. For cyber resiliency, Diversity avoids the risk of system homogeneity, in which compromise of one component can propagate to all other similar components. Diversity offers the benefit of providing alternative ways to deliver required functionality so that if a component is compromised, one or more alternative components which provide the same functionality can be used.

Multiple approaches to diversity can be identified. These include architectural diversity; design diversity; synthetic (or automated) diversity;[117] information diversity; diversity of command, control, and communications (C3) paths (including out-of-band communications); geographic diversity;[118] supply chain diversity [SP 800-160 v1, Bodeau15]; and diversity in operating procedures. In addition, some incidental architectural diversity often results from procurement over time and differing user preferences. Incidental diversity is often more apparent than real (i.e., different products can present significantly different interfaces to administrators or users, while incorporating identical components).

However, diversity can be problematic in several ways. First, it can increase the attack surface of the system. Rather than trying to compromise a single component and propagate across all such components, an adversary can attack any component in the set of alternatives, looking for a path of least resistance to establish a foothold. Second, it can increase demands on developers, system administrators, maintenance staff, and users by forcing them to deal with multiple interfaces to equivalent components. This can result in increased system life cycle costs[119] and also increase the risks that inconsistencies will be introduced, particularly if the configuration alternatives for the equivalent components are organized differently. Third, diversity can be more apparent than real (e.g., different implementations of the same mission functionality all running on the same underlying operating system, applications which reuse selected software components). Thus, analysis of the architectural approach to using diversity is critical. For embedded systems, some approaches to diversity raise a variety of research challenges. And finally, the effectiveness of diversity against adversaries is not an absolute—analysis of diversity strategies is needed to determine the best alternative in the context of adversary TTPs.

Therefore, this design principle calls for the use of Diversity in system architecture and design to take manageability into consideration. It also calls for consideration of diversity in operational processes and practices, including non-cyber alternatives such as out-of-band measures [SP 800-53] for critical capabilities. To reduce cost and other impacts, this design principle is most effective when used in conjunction with the Focus on common critical assets strategic design principle and the Maintain redundancy and Layer and partition defenses structural principles. Measurements related to this design principle can focus on the degree of diversity, the degree of manageability, or both.

[117] Synthetic diversity in conjunction with randomization, a form of Unpredictability, is a form of Moving Target Defense (MTD).

[118] Geographic diversity can be used to support the Make resources location-versatile structural design principle.

[119] These costs have historically been acceptable in some safety-critical systems.

E.5.2.6 Maintain Redundancy

Redundancy is a well-established design principle in Resilience Engineering and Survivability [Sterbenz10]. Approaches to Redundancy include surplus capacity and replication (e.g., cold spares, hot or inline spares) and can be implemented in conjunction with backup and failover procedures. It can enhance the availability of critical capabilities but requires that redundant resources be protected.

Because malware can propagate across homogeneous resources, Redundancy for cyber resiliency should be applied in conjunction with Diversity and should be considered at multiple levels or layers in a layered architecture [Sterbenz14]. However, Redundancy when used in conjunction with Diversity can increase complexity and present scalability challenges.

The extent of Redundancy should be established and maintained through analysis, looking for single points of failure and shared resources. Trends to convergence can, at times, undermine Redundancy. For example, an organization using Voice over Internet Protocol (VOIP) for its phone system cannot assert alternate communications paths for phone, email, and instant messaging.

Because maintaining surplus capacity or spare components increases system life-cycle costs, this design principle is most effective when used in conjunction with the Focus on common critical assets strategic principle—and it is also most effective in conjunction with the Plan and manage diversity and Layer and partition defenses structural principles.

E.5.2.7 Make Resources Location-Versatile

Location-versatile resources are those resources which do not require a fixed location and can be relocated or reconstituted to maximize performance, avoid disruptions, and better avoid becoming a high value target for an adversary. Different approaches can be used to provide location-versatile resources including virtualization, replication, distribution (of functionality or stored data), physical mobility, and functional relocation. Replication is a well-established approach for high-availability systems using multiple, parallel processes, and high-availability data (sometimes referred to as data resilience) using database sharding[120] (although this can present security challenges).

Replication and distribution can be across geographic locations, hardware platforms, or (in the case of services) virtual machines. While replication can take the form of redundancy, it can also involve providing ways to reconfigure system resources to provide equivalent functionality. Data virtualization (i.e., data management which enables applications to retrieve and use data without specific knowledge of the location or format) supports distribution and reduces the likelihood that local (persistent and unmaintained) data stores will proliferate. Composable services enable alternative reconstitution of mission capabilities, and diverse information sources can be used for alternative reconstitution of mission or business data.

Application of this principle involves the use of Dynamic Positioning, often in conjunction with Redundancy and/or Diversity. This principle supports the Support agility and architect for

[120] A database *shard* is a horizontal partition of data in a database. Each individual partition is referred to as a shard or database shard. Each shard is held on a separate database server instance to spread the load.

adaptability strategic principle and can be employed in conjunction with the Maximize transience and Change or disrupt the attack surface structural principles. Some approaches to the reconstitution of mission capabilities can conflict with the Control visibility and use structural principle.

E.5.2.8 Leverage Health and Status Data

In some architectures, many system components are security-unaware, incapable of enforcing a security policy (e.g., an access control policy), and therefore incapable of monitoring policy compliance (e.g., auditing or alerting on unauthorized access attempts). However, most system components provide health and status data to indicate component availability or unavailability for use. These include, for example, components of CPS (particularly components in space systems) and in the emerging IoT. In addition, system components present health and status data to providers (e.g., application or service on a virtual platform in a cloud to a cloud provider) or service-providing components (e.g., application to operating system, device to network) so that the components can allocate and scale resources effectively. Correlation of monitoring data, including health and status data, from multiple layers or types of components in the architecture can help identify potential problems early so they can be averted or contained.

As architectural convergence between information technology (IT) and operational technology (OT) or the IoT increases [SP 1500-201], application of this structural principle will support the Expect adversaries to evolve strategic principle. Given the increasing number and variety of "smart" components in the IoT, application of this principle may be driven by the Focus on common critical assets principle. In addition, components can erroneously or maliciously report health and status data by design or due to compromise. Thus, application of this principle may be more effective in conjunction with the Determine ongoing trustworthiness principle.

E.5.2.9 Maintain Situational Awareness

For security and cyber resiliency, situational awareness encompasses awareness of *system elements*, *threats*, and *mission dependencies* on system elements.[121] Awareness of system elements can rely on security status assessment, security monitoring, and performance monitoring and can be achieved in conjunction with the Leverage health and status data design principle. Awareness of threats involves ingesting and using threat intelligence, recognizing that adversaries evolve. Awareness of system elements and threats (via gathered data, correlated data, and processing capabilities) can be centralized or distributed and can be either enterprise-internal or cross-enterprise (e.g., via a managed security service provider).

Awareness of mission dependencies can be determined *a priori*, as part of system design (e.g., using CJA, MIA, or BIA). Alternately or additionally, mission dependencies can be identified during mission operations by tracking and analyzing resource use. This more dynamic approach supports agility, adaptability, and capabilities to Control visibility and use and Contain and exclude behaviors. While cyber situational awareness remains an active area of research,

[121] As a foundational capability of a Security Operations Center (SOC), situational awareness provides "regular, repeatable repackaging and redistribution of the SOC's knowledge of constituency assets, networks, threats, incidents, and vulnerabilities to constituents. This capability goes beyond cyber intel distribution, enhancing constituents' understanding of the cybersecurity posture of the constituency and portions thereof, driving effective decision-making at all levels [Zimmerman14]."

analytic capabilities are increasingly being offered, and cyber situational awareness is maturing through tailored applications in specific environments.

E.5.2.10 Manage Resources (Risk-) Adaptively

Risk-adaptive management has been developed in multiple contexts. Cybersecurity mechanisms include risk-adaptive access control (RAdAC) for systems—highly adaptive cybersecurity services (HACS) providing such functionality as penetration testing, incident response, cyber hunting, and risk and vulnerability assessment for programs—and integrated adaptive cyber defense (IACD) for the enterprise and beyond. Strategies for risk-adaptive management include:

- Changing the frequency of planned changes (e.g., resetting encryption keys, switching between operating systems or platforms, or changing the configuration of internal routers);
- Increasing security restrictions (e.g., requiring reauthentication periodically within a single session, two-factor authentication for requests from remote locations, or two-person control on specific actions, increasing privilege requirements based on changing criticality);
- Reallocating resources (e.g., reallocating processing, communications, or storage resources to enable graceful degradation, repurposing resources); and
- Discarding or isolating suspected system elements (e.g., terminating a service or locking out a user account, diverting communications to a deception environment, or quarantining processing).

Strategies for implementing this design principle can be applied in conjunction with strategies for implementing Control visibility and use (dynamically changing privileges), Contain and exclude behaviors (disabling resources and dynamic isolation), Layer defenses and partition resources (dynamic partitioning), Plan and manage diversity (switching from one resource to an equivalent resource), and Make resources location-versatile (reconstituting resources).

To be *risk*-adaptive, the selection and application of a strategy should be based on situational awareness—that is, management decisions are based on indications of changes in adversary characteristics, characteristics of system elements, or patterns of operational use which change the risk posture of the system or the mission or business function it supports. Alternately, strategies can be applied unpredictably to address unknown risks.

E.5.2.11 Maximize Transience

Non-persistence is a cyber resiliency strategy to Reduce attack surfaces in the temporal dimension. Virtualization technologies, which simulate the hardware and/or software on which other software executes [SP 800-125B], enable processes, services, and applications to be transient. At the network layer, technologies for network virtualization, network functions virtualization, software-defined networking, and just-in-time connectivity can support non-persistence. Data virtualization provides a strategy for reducing persistent local data stores. As noted above, this principle is synergistic with Make resources location-versatile. Since transient resources can be virtually isolated, this principle can also be used in conjunction with Contain and exclude behaviors.

Logical transient system elements (e.g., processes, files, connections) need to be expunged (i.e., removed in such a way that no data remains on the shared resources).[122] If an executing process or service has been compromised by malicious software which changes its behavior or corrupts the data it offers to other system elements, expunging it, either by bringing it down or by moving it and deleting the prior instance, also mitigates the compromise. This can be done in response to suspicious behavior or can be deliberately unpredictable.

In addition, system elements can be made attritable and expendable, for example, in the case of unmanned air systems. These physically transient system elements also need mechanisms for ensuring that no data is left behind.

The instantiation of a transient resource depends on being able to Determine ongoing trustworthiness of the resources from which it is constructed. Support for such verification and/or validation can include, for example, gold copies of software and configuration data, policy data for network function virtualization, and data quality validation as part of data virtualization.

E.5.2.12 Determine Ongoing Trustworthiness

In the *Control* phase of the cyber-attack life cycle [MITRE18], an adversary can modify system components (e.g., modify software, replace legitimate software with malware) and system data (e.g., modify configuration files, fabricate entries in an authorization database, fabricate or delete audit data) or mission or business data (e.g., deleting, changing, or inserting entries in a mission or business database; replacing user-created files with fabricated versions). These modifications enable the adversary to take actions in the *Execute* and *Maintain* phases of the cyber-attack life cycle. Periodic or ongoing validation can detect the effects of adversary activities before those effects become too significant or irremediable.

A variety of Substantiated Integrity mechanisms can be used to identify suspicious changes. Changes can be to properties or to behavior. Some behaviors—for example, the frequency with which a service makes requests, the latency between a request to it and its response, and the size of requests or responses it makes—can be verified or validated by other services. Other behaviors—for example, processor, memory, disk, or network use—can be verified or validated by other system components (e.g., the operating system's task manager). Note that making the behavior capable of being verified or validated can impede the use of unpredictability.

This principle is strongly synergistic with Manage resources (risk-) adaptively. Some changes can trigger the use of Privilege Restriction or Analytic Monitoring mechanisms. Other changes can trigger quarantine via Segmentation. However, such mechanisms can add storage, processing, and transmission overhead. Therefore, this structural principle is most effective in support of the Focus on common critical assets strategic principle.

Ideally, any system element which cannot be determined to be trustworthy—initially via hardware and software assurance processes and subsequently via Substantiated Integrity—should be assumed to be compromised. However, in practice, that assumption is difficult to

[122] See [SP 800-53] controls SC-4 (Information in Shared Resources) and MP-6 (Media Sanitization).

apply. This principle is consistent with the weaker assumption that some resources will be compromised and calls for mechanisms to detect and respond to evidence of compromise.

Mechanisms to determine trustworthiness need to be applied in a coordinated manner, across architectural layers, among different types of system elements, and (if applicable) with insider threat controls.

E.5.2.13 Change or Disrupt the Attack Surface

Disruption of the attack surface can also lead an adversary to reveal its presence. A growing set of moving target defenses are intended to change or disrupt the attack surface of a system. Moving Target Defense (MTD) is an active area of research and development. MTD can be categorized in terms of the *layer* or level at which the defenses are applied (e.g., software, runtime environment, data, platform, and network). However, MTD can be applied at other layers. For example, when this design principle is used in conjunction with the Make resources location-versatile principle, MTD can also be applied at the physical or geographic levels. MTD is particularly well-suited to cloud architectures [Shetty16] where implementation is at the middleware level.

MTD can also be categorized in terms of strategy: move, morph, or switch. Resources can be moved (e.g., execution of a service can be moved from one platform or virtual machine to another). This approach, which leverages the design principle of Dynamic Positioning, can be used in conjunction with the Make resources location-versatile principle. The terms "cyber maneuver" and MTD are often reserved for morphing—that is, making specific changes to the properties of the data, runtime environment, software, platform, or network [Okhravi13] or by using configuration changes in conjunction with the techniques of Diversity and Unpredictability or randomization [Jajodia11, Jajodia12] rather than including relocation or distribution. Data or software can be morphed using synthetic diversity; the behavior of system elements can be morphed via configuration or resource allocation changes. Morphing can also be part of a Deception strategy. Finally, switching can leverage diversity and distributed resources. Mission applications which rely on a supporting service can switch from one implementation of the service to another. Switching can also be used in conjunction with Deception, as when adversary interactions with the system are switched to a deception environment.

This structural design principle supports the Expect adversaries to evolve strategic principle. It can also support the Reduce attack surfaces strategic principle. Alternately, the principle can support the Assume compromised resources principle. When Unpredictability is part of the way this principle is applied, it should be used in conjunction with the Make the effects of deception and unpredictability user-transparent structural principle.

E.5.2.14 Make Deception and Unpredictability Effects User-Transparent

Deception and unpredictability are intended to increase the adversaries' uncertainty about the system's structure and behavior, what effects an adversary might be able to achieve, and what actions cyber defenders might take in response to suspected malicious cyber-related activities. [Heckman15] provides a detailed discussion of deception and its role in active cyber defense. Deception includes obfuscation, which increases the effort needed by the adversary and can hide mission activities long enough for the mission to complete without adversary disruption.

Active deception can divert adversary activities, causing the adversary to waste resources and reveal TTPs, intent, and targeting.

Unpredictability can apply to structure, characteristics, or behavior. Unpredictable structure (e.g., dynamically changing partitions or isolating components) undermines the adversary's reconnaissance efforts. Unpredictable characteristics (e.g., configurations, selection of an equivalent element from a diverse set) force the adversary to develop a broader range of TTPs. Unpredictable behavior (e.g., response latency) increases uncertainty about effects and about whether system behavior indicates defender awareness of malicious cyber activities.

Unpredictability and deception can be applied separately, as well as synergistically. These two techniques can be highly effective against advanced adversaries. However, deception and unpredictability, if implemented poorly, can also increase the uncertainty of end-users and administrators about how the system will behave. Such user and administrator confusion can reduce overall resilience, reliability, and security. This uncertainty can, in turn, make detection of unauthorized or suspicious behavior more difficult. This design principle calls for a sound implementation, which makes system behaviors directed at the adversary transparent to end-users and system administrators.

E.6 RELATIONSHIPS AMONG CYBER RESILIENCY CONSTRUCTS

Sections E.1 through E.5 presented and described the cyber resiliency constructs of goals, objectives, techniques, approaches, and design principles. Table E-12 and Table E-13 illustrate that the mapping between the goals and objectives is many-to-many, as are the mappings between techniques (including the approaches to implementing or applying techniques) and objectives.

TABLE E-12: CYBER RESILIENCY OBJECTIVES SUPPORTING CYBER RESILIENCY GOALS

Goals / Objectives	ANTICIPATE	WITHSTAND	RECOVER	ADAPT
Prevent/Avoid	X	X		
Prepare	X	X	X	X
Continue		X	X	
Constrain		X	X	
Reconstitute			X	
Understand	X	X	X	X
Transform			X	X
Re-Architect			X	X