**Security Concept Template**

**1. Introduction**

* **Purpose**: Briefly state the purpose of this document—to define the security posture, objectives, controls, and integration strategies for the system, ensuring security is embedded throughout its lifecycle. It aims to achieve comprehensive program and system protection within cost, schedule, and performance constraints while maintaining an acceptable level of risk.
* **Scope**: Define the boundaries of the security effort, including which system elements, data types, user groups, and lifecycle phases are covered. This concept applies from concept development through disposal, adapting to evolving threats and operational contexts.
* **Goals**: Outline the overarching goals of the security concept, such as ensuring system trustworthiness, meeting compliance standards, and enabling mission assurance.

**2. Security Objectives and Requirements**

* **Derived Objectives**: Clearly state the specific security objectives derived from stakeholder protection needs, regulatory standards (e.g., NIST, ISO), and threat analysis. These objectives often include confidentiality, integrity, availability, accountability, and assurance of information assets. Trustworthiness is a broader concept encompassing privacy, reliability, resilience, safety, and security.
* **Security Requirements**: Document explicit, testable statements specifying the security functions and properties the system must fulfill. This includes cybersecurity requirements for all DoD programs. Security scenarios, including misuse/abuse situations, are developed to aid in defining these requirements.

**3. Security Controls Architecture**

* **Control Placement and Interaction**: Describe how security controls (e.g., encryption, access control, intrusion detection) are positioned within the system architecture and how they relate to other components.
* **Defense-in-Depth Strategy**: Outline a layered approach to controls to prevent single points of failure.
* **Protection of Assets**: Explain provisions to protect intellectual property (IP) and ensure efficient access to information only for those with a need to know.

**4. Threat and Risk Analysis**

* **Threat Identification**: Identify potential attack vectors, vulnerabilities, and types of misuse or malicious behavior. This can utilize methods like STRIDE or OCTAVE. Security risks are assessed as part of the overall risk management process.
* **Risk Mapping and Mitigation**: Map identified risks to specific mitigation strategies (e.g., "Supply chain attack → Hardware root of trust"). This includes identifying critical program information (CPI) and mission-critical functions/components that need protection.

**5. Security Integration Strategy**

* **Alignment with System Architecture**: Ensure that security controls interoperate seamlessly with functional components and external systems. Architecture views and viewpoints can be used to frame specific security concerns.
* **Cross-Cutting Concerns**: Address how security is integrated across interfaces, data flows, and how it considers emergent behaviors, especially in System of Systems (SoS) environments.
* **DevSecOps Integration**: Integrate security practices into development and operations pipelines for secure code development and continuous delivery, including coding standards, analysis rules, and test coverage.

**6. Lifecycle Processes and Application**

* **Development**: Define how security-by-design principles will be applied in requirements definition, architecture, design, and testing.
* **Operations**: Outline strategies for continuous monitoring, patch management, and incident response during the system's operational phase.
* **Disposal**: Address secure decommissioning and data sanitization protocols during the retirement of the system and its information. This includes addressing hazardous materials and legal obligations.

**7. Key Outputs and Deliverables**

* **Security Controls Catalog**: A list of controls, implementation details, and their traceability to identified threats.
* **Threat Models**: Diagrams illustrating attack surfaces and potential compromise paths (e.g., Data Flow Diagrams).
* **Security Requirements Document**: Technical specifications (e.g., "Data-at-rest encrypted with AES-256").
* **Risk Register**: Documentation of security-specific risks, their mitigations, and ownership.

**8. Integration with Other Systems Engineering Artifacts**

* **Requirements Traceability**: Show how security requirements are linked to broader stakeholder needs and system functions.
* **Architectural Views**: Utilize security-specific diagrams within system architecture models (e.g., data flow diagrams with trust boundaries).
* **Verification and Validation**: Define how security testing (e.g., penetration testing, code reviews, cybersecurity test and evaluation) will be performed to validate the effectiveness of controls.
* **Concept of Operations (ConOps)**: Security aspects directly influence the ConOps, which describes how the system will be used and interact with its environment.

**9. Special Considerations and Best Practices**

* **Compliance**: Highlight adherence to regulatory standards (e.g., NIST SP 800-160, ISO 27001) critical for product acceptance and certification.
* **Supply Chain Risk Management (SCRM)**: Address risks across the supply chain, incorporating disciplines like anti-tamper and hardware/software assurance.
* **Model-Based Systems Engineering (MBSE)**: Leverage MBSE to support the integration of models and data, ensuring data security and intellectual property protection within the modeling environment itself.
* **Cybersecurity Testing**: Include specific plans for cybersecurity test and evaluation at various operational and developmental stages.