Defense Acquisition Guidebook

Chapter 13 - Program Protection

Production Date: 15 May 2013

DEFENSE ACQUISITION GUIDEBOOK
Chapter 13 - Program Protection

13.0. Overview

13.1. The Program Protection Process

13.2. The Program Protection Plan (PPP)

13.3. Critical Program Information (CPI) and Mission-Critical Functions and Components

13.4. Intelligence and Counterintelligence (CI) Support

13.5. Vulnerability Assessment

13.6. Risk Assessment

13.7. Countermeasures

13.8. Horizontal Protection

13.9. Foreign Involvement

13.10. Managing and Implementing PPPs

13.11. Compromises

13.12. Costs

13.13. Contracting

13.14. Detailed System Security Engineering

13.0. Overview

13.0.1. Purpose

13.0.2. Contents

13.1. The Program Protection Process

13.2. The Program Protection Plan (PPP)

13.0. Overview

Program Protection is the integrating process for mitigating and managing risks to advanced technology and mission-critical system functionality from foreign collection, design vulnerability, or supply chain exploitation/insertion, battlefield loss, and unauthorized or inadvertent disclosure throughout the acquisition lifecycle.

At its core, Program Protection protects technology, components, and information from compromise through the cost-effective application of countermeasures to mitigate risks posed by threats and vulnerabilities. In a simple sense, Program Protection seeks to defend warfighting capability by keeping secret things from getting out and keeping malicious things from getting in. Where the capability is derived from advanced or leading-edge technology, Program Protection mitigates the risk that the technology will be lost to an adversary; where the capability is derived from integration of commercially available or developed components, Program Protection mitigates the risk that design vulnerabilities or supply chains will be exploited to degrade system performance. The Program Protection Plan (PPP) is the milestone acquisition document that describes the plan, responsibilities, and decisions for all Program Protection activities.

13.0.1. Purpose

This chapter provides guidance and expectations for the major activities associated with Program Protection.

13.0.2. Contents

Chapter 13 addresses the following topics:

The Program Protection Process

The Program Protection Plan (PPP)

Critical Program Information (CPI) and Mission-Critical Functions and Components

Intelligence and Counterintelligence (CI) Support

Vulnerability Assessment

Risk Assessment

Countermeasures

Horizontal Protection

Foreign Involvement

Managing and Implementing Program Protection Plans (PPP)

Compromises

Costs

Contracting

Detailed Systems Security Engineering (SSE)

Program Protection Plan (PPP) Review/Approval

Program Protection Plan (PPP) Classification Guidance

13.1. The Program Protection Process

Program Protection is an iterative risk management process within system design and acquisition, composed of the following activities:

- Critical Program Information (CPI) Identification and Criticality Analysis (Section 13.3)
- Threat Analysis (Section 13.4)
- Vulnerability Assessment (Section 13.5)
- Risk Assessment (Section 13.6)
- Countermeasure Implementation (Section 13.7)
- Horizontal Protection (Section 13.8)
- Foreign Involvement (Section 13.9)

Additional considerations (Defense Exportability Features, Program Protection Plan (PPP) Approval, etc.) are covered in subsequent sections.

Commanders, Program Executive Officers, S&T Project Site Directors, Program Managers (PMs) (used throughout this chapter to include program/project leaders prior to official PM designation), systems engineering, system security, information assurance, Test and Evaluation (T&E), and acquisition personnel should be aware of the Program Protection process and should be engaged in supporting it. Program Managers are responsible with complying with this process holistically such that protection decisions are made in the context and trade space of other cost, schedule, and performance considerations. It is important to implement this process across the full acquisition lifecycle in order to build security into the system. The process is repeated at each of the following points in the lifecycle, building on the growing system maturity:

- Systems Engineering Technical Reviews (SETR) (see Section 13.10.2 for further elaboration on specific Systems Engineering Technical Reviews event expectations), starting Pre-Milestone A with the Alternative Systems Review (ASR)
- Systems Engineering (SE) analyses that support preparation for each Acquisition Milestone (see Sections 13.7.6 and 13.14 for further elaboration on how this process is tied to lifecycle phase-related Systems Security Engineering (SSE)

- Development and release of each Request for Proposal (RFP) (see Section 13.13.1 for further details on what should be incorporated in the Request for Proposal (RFP) package)

At each of these points, the process is iterated several times to achieve comprehensive results that are integrated into the system design and acquisition. This process applies to all programs and projects regardless of acquisition category (ACAT) or status (i.e., all acquisition categories (ACATs), Quick Reaction Capability (QRC), Request for Information (RFI), Joint Capability Technology Demonstration (JCTD), Science and Technology (S&T) or Authority to Operate (ATO)), or whether the technology is meant for Government and or military use.

13.2. The Program Protection Plan (PPP)

Program Protection is the Department's holistic approach for delivering trusted systems and ensures that programs adequately protect their technology, components, and information. The purpose of the Program Protection Plan (PPP) is to ensure that programs adequately protect their technology, components, and information throughout the acquisition process during design, development, delivery and sustainment. The scope of information includes information that alone might not be damaging and might be unclassified, but that in combination with other information could allow an adversary to clone, counter, compromise or defeat warfighting capability.

The process of preparing a PPP is intended to help program offices consciously think through what needs to be protected and to develop a plan to provide that protection. Once a PPP is in place, it should guide program office security measures and be updated as threats and vulnerabilities change or are better understood.

It is important that an end-to-end system view be taken when developing and executing the PPP. External, interdependent, or government furnished components that may be outside a program managers' control must be considered.

The PPP is the focal point for documentation of the program protection analysis, plans and implementation within the program for understanding and managing the full spectrum of the program throughout the acquisition lifecycle. The PPP is a plan, not a treatise; it should contain the information someone working on the program needs to carry out his or her Program Protection responsibilities and it should be generated as part of the program planning process.

The Program Protection Plan Outline and Guidance , established as expected business practice through a July 18, 2011 Principal Deputy Under Secretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) policy memo, can be found at: http://www.acq.osd.mil/se/docs/PDUSD-ATLMemo-Expected-Bus-Practice-PPP-18Jul11.pdf .

13.3. Critical Program Information (CPI) and Mission-Critical Functions and Components

13.3.1. Critical Program Information (CPI)

13.3.2. Mission-Critical Functions and Components

13.3.2.1. Criticality Analysis (CA)

13.3. Critical Program Information (CPI) and Mission-Critical Functions and Components

Critical Program Information (CPI) and mission-critical functions and components are the foundations of Program Protection. They are the technology, components, and information that provide mission-essential capability to our defense acquisition programs, and Program Protection is the process of managing the risks that they will be compromised.

13.3.1. Critical Program Information (CPI)

DoDI 5200.39 defines Critical Program Information (CPI) as Elements or components of a research, development, and acquisition (RDA) program that, if compromised, could cause significant degradation in mission effectiveness; shorten the expected combat-effective life of the system; reduce technological advantage; significantly alter program direction; or enable an adversary to defeat, counter, copy, or reverse engineer the technology or capability. Includes information about applications, capabilities, processes, and end-items. Includes elements or components critical to a military system or network mission effectiveness. Includes technology that would reduce the US technological advantage if it came under foreign control.

Metaphorically, CPI should be thought of as the technological crown jewels of the program. The United States gains military advantages from maintaining technology leads in key areas, so we must protect them from compromise in the development environment and on fielded systems.

CPI may include classified military information that is considered a national security asset that will be protected and shared with foreign governments only when there is a clearly defined benefit to the United States (see DoD Instruction 5200.39). It may also include Controlled Unclassified Information (CUI), which is official unclassified information that has been determined by designated officials to be exempt from public disclosure, and to which access or distribution limitations have been applied in accordance with national laws and regulations such as the International Traffic in Arms Regulations for U.S. Munitions List items and the Export Administration Regulations for commerce controlled dual-use items. In some cases (dependent on the PM's determination) a commercial-off-the shelf (COTS) technology can be designated CPI if the COTS element is determined to fulfill a critical function within the system and the risk of manipulation needs mitigation.

CPI requires protection to prevent unauthorized or inadvertent disclosure, destruction, transfer, alteration, reverse engineering, or loss (often referred to as "compromise").

CPI identified during research and development or Science and Technology should be safeguarded to sustain or advance the DoD technological lead in the warfighter's battlespace or joint operational arena.

The CPI, if compromised, will significantly alter program direction; result in unauthorized or

inadvertent disclosure of the program or system capabilities; shorten the combat effective life of the system; or require additional research, development, test, and evaluation resources to counter the impact of its loss.

The theft or misappropriation of U.S. proprietary information or trade secrets, especially to foreign governments and their agents, directly threatens the economic competitiveness of the U.S. economy. Increasingly, foreign governments, through a variety of means, actively target U.S. businesses, academic centers, and scientific developments to obtain critical technologies and thereby provide their own economies with an advantage. Industrial espionage, by both traditionally friendly nations and recognized adversaries, proliferated in the 1990s and has intensified with computer network attacks today.

Information that may be restricted and protected is identified, marked, and controlled in accordance with DoD Directives 5230.24 and 5230.25 or applicable national-level policy and is limited to the following:

- Information that is classified in accordance with Executive Order 13526, and
- Unclassified information that has restrictions placed on its distribution by:
- U.S. Statutes (e.g., Arms Export Control Act , Export Administration Act);
- Statute-driven national regulations (e.g., Export Administration Regulations (EAR), International Traffic in Arms Regulations (ITAR)); and
- Related national policy (e.g., Executive Order 13526, National Security Decision Directive 189).
- 13.3.1.1 Critical Program Information (CPI) Identification

CPI determination is done with decision aids and Subject Matter Experts (SMEs). As general guidance, PMs should identify an element or component as CPI if:

- Critical technology components will endure over its lifecycle
- A critical component which supports the warfighter is difficult to replace
- A capability depends on technology that was adjusted/adapted/calibrated during testing and there is no other way to extrapolate usage/function/application
- The component / element was identified as CPI previously and the technology has been improved or has been adapted for a new application
- The component / element contains a unique attribute that provides a clear warfighting advantage (i.e. automation, decreased response time, a force multiplier)
- The component / element involves a unique method, technique, application that cannot be achieved using alternate methods and techniques
- The component / elements performance depends on a specific production process or procedure
- The component / element affords significant operational savings and/or lower operational risks over prior doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF) methods
- The Technology Protection and/or Systems Engineering (SE) Team recommends that the component/element is identified as CPI
- The component / element will be exported through Foreign Military Sales (FMS)/Direct

Commercial Sales (DCS) or International Cooperation

PMs should contact their Component research and development acquisition protection community for assistance in identifying CPI.

13.3.2. Mission-Critical Functions and Components

Mission-critical functions are those functions of the system being acquired that, if corrupted or disabled, would likely lead to mission failure or degradation. Mission-critical components are primarily the elements of the system (hardware, software, and firmware) that implement critical functions. In addition, the system components which implement protections of those inherently critical components, and other components with unmediated access to those inherently critical components, may themselves be mission-critical.

Mission-critical functions and components are equal in importance to Critical Program Information (CPI) with respect to their inclusion in comprehensive program protection, it's planning (documented in the Program Protection Plan (PPP)), and it's execution, including:

- Trade-space considerations (including cost/benefit analyses)
- Resource allocations (staffing and budget)
- Countermeasures planning and implementation
- Adjustment of countermeasures, as appropriate, for variations in the planned use or environment of inherited critical components
- Summary of consequences if compromised
- Residual risk identification after countermeasures are implemented, including follow-up mitigation plans and actions

Efforts to identify mission-critical functions and components and their protection must begin early in the lifecycle and be revised as system designs evolve and mature.

13.3.2.1. Criticality Analysis (CA)

What is a Criticality Analysis (CA)? CA is the primary method by which mission-critical functions and components are identified and prioritized. It is an end-to-end functional decomposition of the system which involves:

- Identifying and prioritizing system mission threads;
- Decomposing the mission threads into their mission-critical functions; and
- Identifying the system components (hardware, software, and firmware) that implement those functions; i.e., components that are critical to the mission effectiveness of the system or an interfaced network.

Also included are components that defend or have unmediated access to mission-critical components.

The identified functions and components are assigned levels of criticality commensurate with the

consequence of their failure on the system's ability to perform its mission, as shown in Table 13.3.2.1.T1 .

Table 13.3.2.1.T1. Protection Failure Criticality Levels

Level I Total Mission Failure	Program protection failure that results in total compromise of mission capability
Level II Significant/Unacceptable Degradation	Program protection failure that results in unacceptable compromise of mission capability or significant mission degradation
Level III Partial/Acceptable	Program protection failure that results in partial compromise of mission capability or partial mission degradation
Level IV Negligible	Program protection failure that results in little or no compromise of mission capability

When to perform a CA? The CA is an iterative process. To be effective, many CAs must be executed across the acquisition lifecycle, building on the growing system maturity, knowledge gained from prior CAs, updated risk assessment information, and updated threat and vulnerability data.

At each key decision point, system design changes may result in adding or removing specific items from the list of critical system functions and components. Guidance for the iterative performance of CAs includes:

- Prior to Milestone A: Evaluate mission-threads, identify system functions, and analyze notional system architectures to identify mission-critical functions.
- Prior to Milestone B: Refine the critical function list and identify critical system components and candidate subcomponents (hardware, software, and firmware).
- Prior to CDR: Analyze the detailed design/architecture and update the list to identify all critical system components and subcomponents.

Who performs the CAs? The Government program office should perform an initial CA early in the lifecycle (pre-Milestone A). When contracts are awarded, the DoD contracting office should develop Requests for Proposals (RFPs) that require contractors to perform updated CAs periodically, based on earlier CAs (see Section 13.13.1.2 for guidance on what to include in the Statement of Work (SOW) requirements).

The CA should be led by systems engineers and mission/operator representatives; however, it is a team effort and mission-critical functions and components must be identified by a multi-disciplined group.

How is a CA performed? What is the process? While the Government should perform an initial CA during the Materiel Solution Analysis (MSA) phase, realize that it may only be possible to execute some of the steps in the CA process given below and/or to execute them at a high level. As noted previously, to be effective, Criticality Analyses (CAs) must be executed iteratively

across the acquisition lifecycle, building on the growing system maturity, knowledge gained from prior CAs, updated risk assessment information, and updated threat and vulnerability data.

For example, the first pass through the CA process, together with assessments of vulnerabilities, threats, risks, and countermeasures, might take just a few days and provide a 30% solution. This CA might only involve Subject Matter Expert (SME) input during several work sessions (to address system and architecture), as opposed to detailed information collected from numerous program documents. For an early iteration, precision is not possible, as it takes several iterations to complete the initial Criticality Analysis (CA).

The detailed procedural steps in performing a CA are:

Identify Missions and Mission-Essential Functions	**Sources of Information**
1. Identify mission threads and principle system functions. • Derived first during pre-Milestone A and revised as needed for successive development milestones.	Joint Capabilities Integration Development System (JCIDS) Documents: Initial Capabilities Documents (ICD), Capability Development Documents (CDD), Capability Production Documents (CPD) Concept of Operations
2. If possible or necessary, group the mission capabilities by relative importance. Training or reporting functions may not be as important as core mission capabilities.	Operational Representative Subject Matter Expertise (Integration Experts, Chief Engineers)
3. Identify the systems mission-critical functions based on mission threads and the likelihood of mission failure if the function is corrupted or disabled. (Mission-critical functions may include navigating, targeting, fire control, etc.).	Activity Diagrams Use Cases Functional Decomposition Potential Department of Defense Architecture Framework (DODAF) Sources • OV-5 (Operational Activity Model) • SV-4 (System Functionality Description) Subject Matter Expertise
Identify Critical Subsystems, Configuration Items, and Components	

4. Map the mission threads and functions to the system architecture and identify critical subsystems, Configuration Items, and sub-Cis (components). Note : Focus on Configuration Items and components containing Information and Communications Technologies (ICT). Logic-bearing components have been singled out as often implementing critical functions and as susceptible to lifecycle corruption.	System/Segment Design Document Architecture Description Document Requirements Traceability/Verify. Matrix Potential Department of Defense Architecture Framework (DODAF) Sources • SV-5a (Operational Activity to System Function Traceability Matrix)
5. Assign levels of criticality (I, II, III, IV) to the identified Configuration Items or components. Factors or criteria may include: • Frequency of component use across mission threads • Presence of redundancy triple-redundant designs can indicate critical functions. • Subject matter expertise	Subject Matter Expertise • Systems Engineer • Operators Representative • Program Office
6. Identify any Configuration Items or components that do not directly implement critical functions, but either have unmediated communications access (i.e., an open access channel) to one or more critical functions or protect a critical function. • Which components give or receive information to/from this the critical components? Note : a non-critical component may communicate with a critical function in a way that exposes the critical function to attack. In some cases, the architecture may need to include defensive functions or other countermeasures to protect the critical functions.	Architecture Diagrams Subject Matter Expertise Data Flow Diagram
Initial Start Conditions	

7. Identify critical conditions/information required to initialize the system to complete mission-essential functions. a. What information is needed to successfully execute capabilities? How is this information obtained, provided, or accessed by the system? b. How quickly must information be received to be useful? c. Does the sequence in which the system initializes itself (power, software load, etc.) have an impact on performance?	Data Flow Diagram Information Support Plan
8. Based on the answers to the questions above, identify these functions or components to be included in Program Protection risk management.	
Operating Environment	
9. Identify the system functions or components required to support operations in the intended environment. This may include propulsion (the system has to roll, float, fly, etc.), thermal regulation (keep warm in space, keep cool in other places, etc.) or other environmentally relevant subsystems that must be operational before the system can perform it's missions.	Architecture Diagrams
10. Identify the Information and Communications Technologies (ICT) implementing those system functions and any associated vulnerabilities with the design and implementation of that Information and Communications Technologies (ICT).	
Critical Suppliers	
11. Identify suppliers of critical configuration items or Information and Communications Technologies (ICT) components.	Manufacturing Lead
Note: Repeat this process as the system architecture is refined or modified, such as at SETRs and major acquisition milestone decision points • Design changes may result in adding or removing specific Configuration Items and sub-Configuration Items from the list of critical functions and components	

Important considerations in carrying out the CA process described above include:

- Document the results of each step
  - Include rationale

- Use SE tools to support the analysis; for example:
    - Fault-tree analysis can be useful in determining critical components (see Section 13.7.6 for further details)
    - What information is needed to successfully execute capabilities?
    - How is this information obtained, provided, or accessed by the system?
    - How quickly must information be received to be useful?
    - Does the sequence in which the system initializes itself (power, software load, etc.) have an impact on performance?
    - Example: These may include propulsion (the system has to roll, float, fly, etc.), thermal regulation (keep warm in space, keep cool in other places, etc.), or other environmentally relevant subsystems that must be operational before the system can perform it's missions.
- Use available artifacts to inform the CA; for example:
    - SE artifacts such as architectures/designs and requirements traceability matrices
    - Available threat and vulnerability information
    - Residual vulnerability risk assessments to inform follow-up CAs

- In isolating critical functions/components, identify critical conditions/information required to initialize the system to complete mission-critical functions

- Identify the subsystems or components required to support operations in the intended environment

What is the CA output? The expected output of an effective CA process is:

- A complete list of mission-critical functions and components
- Criticality Level assignments for all items in the list
- Rationale for inclusion or exclusion from the list
- Supplier information for each critical component
- Identification of critical elements for inclusion in a Defense Intelligence Agency (DIA) Threat Assessment Center (TAC) Request (See Section 13.4)

The identification of critical functions and components and the assessment of system impact if compromised is documented in the Program Protection Plan (PPP) as discussed in Appendix C (Table C-1) of the PPP Outline.

The prioritization of Level I and Level II components for expending resources and attention will be documented in the PPP as discussed in Appendix C (Table C-2) of the PPP Outline.

Why is the CA performed? The level I and selected level II components from the CA are used as inputs to the threat assessment, vulnerability assessment, risk assessment, and countermeasure selection. The following sections describe these activities.

13.4. Intelligence and Counterintelligence (CI) Support

13.4.1. Defense Intelligence Agency Supply Chain Risk Management Threat Assessment

Center (DIA SCRM TAC)

13.4.1.1. Threat Assessments

13.4.1.2. Criticality Analysis to Inform TAC Requests

13.4.2. Counterintelligence Support

13.4.2.1. Requesting Counterintelligence (CI) Analytical Support

13.4.2.2. Preliminary Counterintelligence (CI) Analytical Product

13.4.2.3. Final Counterintelligence (CI) Analytical Product

13.5. Vulnerability Assessment

13.5.1. Approaches to Identifying Vulnerabilities

13.5.2. Rating Vulnerability Severity

13.5.3. Identifying Vulnerability Mitigations or Countermeasures

13.5.4. Interactions with Other Program Protection Processes

13.4. Intelligence and Counterintelligence (CI) Support

13.4.1. Defense Intelligence Agency Supply Chain Risk Management Threat Assessment Center (DIA SCRM TAC)

DoD has designated the Defense Intelligence Agency (DIA) to be the DoD enterprise focal point for threat assessments needed by the DoD acquisition community to assess supplier risks. DIA established the Threat Assessment Center (TAC) for this purpose. The Threat Assessment Center (TAC) provides the enterprise management and interface to resources within the National Counterintelligence Executive (NCIX), and coordinates with the Defense Intelligence and Defense Counterintelligence Components to provide standardized all-source intelligence assessments to support acquisition risk management efforts. This enterprise integration role of the DoD Threat Assessment Center (TAC) was designed and organized to achieve comprehensive and consistent engagements with the United States Government (USG) across all of the Military Departments (MILDEPs) and Defense Agencies needs for supplier threat assessments and to ensure the efficiency and coherent use of the results provided to the acquisition community.

13.4.1.1. Threat Assessments

Defense Intelligence Agency (DIA) Threat Assessments provide specific and timely threat characterization of the identified suppliers to inform program management. Threat Assessment

Center (TAC) reports are used by the Program Manager and the engineering team to assist in selecting supplier and/or architecture alternatives and developing appropriate mitigations for supply chain risks. For the policy and procedures regarding the request, receipts, and handling of Threat Assessment Center (TAC) reports, refer to DoD Instruction O-5240.24.

Supplier threat assessment requests are developed based on the criticality analysis. An annotated work breakdown structure (WBS) or system breakdown structure (SBS) that identifies the suppliers of the critical functions components may be used to assist with the creation of the Threat Assessment Center (TAC) requests. Supplier threat assessment requests may be submitted as soon as sources of critical capability are identifiable. Near the end of the Materiel Solution Analysis (MSA) Phase, as some threat information is available from the capstone threat assessment (CTA) and technologies and potential suppliers are identified, Supply Chain Risk Management (SCRM) Threat Assessments may be used to assist in defining lowest risk architectures, based on suppliers for particular architecture alternatives. Note that early in the system lifecycle the threat requests may be more focused on suppliers in general technology areas to inform architecture choices, while later in the system lifecycle they may be more focused on critical components defined in the criticality analysis.

13.4.1.2. Criticality Analysis to Inform TAC Requests

Engineering activities related to Supply Chain Risk Management (SCRM) begin as architecture alternatives are considered and continue throughout the acquisition lifecycle. As the systems engineering team develops the initial view of system requirements and system design concepts, a criticality analysis is performed to define critical technology elements. Criticality analysis produces a list of critical components and suppliers that are used to generate Threat Assessment Center (TAC) requests and supplier risk mitigation.

The criticality analysis begins early in the system acquisition lifecycle and continues to be updated and enhanced through Milestone C, becoming more specific as architecture decisions are made and the system boundaries are fully defined. The engineering team may at any point, beginning prior to Milestone A, identify technology elements and potential manufacturers and request supplier threat assessments. It is expected that the number of supplier threat assessment requests will grow as the criticality analysis becomes more specific and the system architecture and boundaries are fully specified, i.e., the greatest number of Threat Assessment Center (TAC) requests will typically occur between Milestones B and C (i.e., Preliminary Design Review (PDR) and Critical Design Review (CDR)). See Section 13.3.2 for more information.

13.4.2. Counterintelligence Support

[This section will be updated to reflect implementation guidance for DoD Instruction O-5240.24, but content was not ready by the submission deadline for this major update.]

When an acquisition program containing Critical Program Information (CPI) is initiated, the Program Manager (PM) should request a counterintelligence (CI) analysis of CPI from the servicing CI organization. The CI analysis focuses on how the opposition sees the program and on how to counter the opposition's collection efforts. The CI analyst, in addition to having an in-

depth understanding and expertise on foreign intelligence collection capabilities, must have a good working knowledge of the U.S. program. Therefore, CI organizations need information that describes the CPI and its projected use to determine the foreign collection threat to an acquisition program.

The CI analytical product that results from the analysis will provide the PM with an evaluation of foreign collection threats to specific program or project technologies, the impact if that technology is compromised, and the identification of related foreign technologies that could impact program or project success. The CI analytical product is updated as necessary (usually prior to each major milestone decision) throughout the acquisition process. Changes are briefed to the Program or PM within 60 days.

13.4.2.1. Requesting Counterintelligence (CI) Analytical Support

The PM's request to the counterintelligence organization for an analytical product normally contains the following information and is classified according to content:

- Program office, designator, and address;
- PM's name and telephone number;
- Point of contact's (POCs) name, address, and telephone number;
- Supporting or supported programs' or projects' names and locations;
- Operational employment role, if any;
- List of CPI;
- Relationship to key technologies or other controlled technology lists of the Departments of Defense, Commerce, and/or State;
- CPI technical description, including distinguishing characteristics (e.g., emissions; sight or sensor sensitivities) and methods of CPI transmittal, usage, storage, and testing;
- Use of foreign equipment or technology during testing (if known);
- Anticipated foreign involvement in the development, testing, or production of the U.S. system;
- Contractor names, locations, Points of Contact (POCs), and telephone numbers, as well as the identification of each CPI used at each location; and
- Reports of known or suspected compromise of CPI.

13.4.2.2. Preliminary Counterintelligence (CI) Analytical Product

After the request is submitted, the DoD Component CI organization provides a preliminary CI analytical product to the program manager within 90 days. A preliminary analytical product is more generic and less detailed than the final product. It is limited in use since it only provides an indication of which countries have the capability to collect intelligence on the U.S. system or technology as well as the possible interest and/or intention to collect it. The preliminary CI analytical product may serve as the basis for the draft Program Protection Plan.

13.4.2.3. Final Counterintelligence (CI) Analytical Product

The program manager approves the Program Protection Plan only after the final CI analysis of

Critical Program Information (CPI) has been received from the applicable DoD Component CI and/or intelligence support activity. Normally, the CI analysis of CPI is returned to the requesting program office within 180 days of the CI and/or intelligence organization receiving the request.

The CI analysis of CPI answers the following questions about CPI:

- Which foreign interests might be targeting the CPI and why?
- What capabilities does each foreign interest have to collect information on the CPI at each location identified by the program office?
- Does evidence exist to indicate that a program CPI has been targeted?
- Has any CPI been compromised?

13.5. Vulnerability Assessment

This section briefly describes a process for identifying vulnerabilities in systems. A vulnerability is any weakness in system design, development, production, or operation that can be exploited by a threat to defeat a systems mission objectives or significantly degrade it's performance. Decisions about which vulnerabilities need to be addressed and which countermeasures or mitigation approaches should be applied will be based on an overall understanding of threats, risks, and program priorities. Vulnerability assessment is a step in the overall risk assessment process, as described in Section 13.5 .

Vulnerability assessments should focus first on the mission-critical functions and components identified by a Criticality Analysis (see Section 13.3.2) and the Critical Program Information (CPI) identified (see Section 13.3.1). The search for vulnerabilities should begin with these critical functions, associated components and CPI.

13.5.1. Approaches to Identifying Vulnerabilities

Potential malicious activities that could interfere with a systems operation should be considered throughout a systems design, development testing, production, and maintenance. Vulnerabilities identified early in a systems design can often be eliminated with simple design changes at lower cost. Vulnerabilities found later may require add-on protection measures or operating constraints that may be less effective and more expensive.

The principal vulnerabilities to watch for in an overall review of system engineering processes are:

- Access paths within the supply chain that would allow threats to introduce components that could cause the system to fail at some later time (components here include hardware, software, and firmware); and
- Access paths that would allow threats to trigger a component malfunction or failure at a time of their choosing.

Supply chain here means any point in a systems design, engineering and manufacturing

development, production, configuration in the field, updates, and maintenance. Access opportunities may be for extended or brief periods (but potentially exploitable).

Two design processes that have proven effective in identifying vulnerabilities are Fault Tree Analysis (FTA) and Failure Modes, Effects, and Criticality Analysis (FMECA). An important twist in applying these techniques is that the potential sources of failures are malicious actors, not random device failures. Malicious actors invalidate many assumptions made about randomness and event independence in reliability analysis. Both FTA and FMECA assume hypothetical system or mission failures have occurred, and trace back through the system to determine contributing component malfunctions or failures. For a vulnerability assessment, the possible access paths and opportunities a threat would have to exercise to introduce the vulnerability or trigger the failure must also be considered.

For software, a number of software tools are available that will identify common vulnerabilities. These tools apply different criteria and often find different flaws. It is therefore beneficial to run code through multiple tools.

Controls on access to software during development and in the field are critical to limiting opportunities for exploitation. One approach to testing access controls and software vulnerabilities in general is Red Teaming. Red teams typically subject a system under test to a series of attacks, simulating the tactics of an actual threat. (See further discussion of software tools and access controls in Section 13.7.3, Software Assurance .)

13.5.2. Rating Vulnerability Severity

The consequences of exploiting a vulnerability should be levels on the same scale as criticality (catastrophic, critical, marginal, and negligible). Vulnerability levels however, may not be the same as the criticality levels. For example, a vulnerability may expose a critical function in a way that has only a marginal consequence. At the same time, another vulnerability may expose several critical functions that taken together could lead to a catastrophic system failure.

Additional factors that should be rated include the ease or difficulty of exploiting a vulnerability, the developers or maintainers ability to detect access used to introduce or trigger a vulnerability, and any other deterrents to threats such as the consequences of being caught. A summary table of the vulnerability assessment is illustrated in Table 13.5.2.T1 .

Table 13.5.2.T1. Sample summary vulnerability assessment table

Critical Components (Hardware, Software, Firmware)	Identified Vulnerabilities	Exploitability	System Impact (I, II, III, IV)	Exposure
Processor X	Vulnerability 1 Vulnerability 4	Low Medium	II	Low

SW Module Y	Vulnerability 1 Vulnerability 2 Vulnerability 3 Vulnerability 6	High Low Medium High	I	High
SW Algorithm A	None	Very Low	II	Very Low
FPGA 123	Vulnerability 1 Vulnerability 23	Low	I	Low

13.5.3. Identifying Vulnerability Mitigations or Countermeasures

There are multiple countermeasures available to mitigate a wide range of possible vulnerability risks. Design changes may either 1) eliminate an exploitation, 2) reduce the consequences of exploitation, or 3) block the access necessary for introduction or exploitation. Add-on protection mechanisms may block the access required to trigger an exploitation. An effective update process, particularly for software, can correct or counteract vulnerabilities discovered after fielding.

As a result of globalization, commercial off-the-shelf (COTS) components are designed and manufactured anywhere in the world, and it may be difficult or impossible to trace all opportunities for malicious access. If the source of a particular component might be compromised, it may be possible to substitute a comparable component from another, more dependable source. Anonymous purchases (Blind Buys) may prevent an untrustworthy supplier from knowing where the component is being used. More extensive testing may be required for critical components from unverified or less dependable sources. A variety of different countermeasures should be identified to inform and provide options for the program managers risk-mitigation decisions.

13.5.4. Interactions with Other Program Protection Processes

Investigation of vulnerabilities may indicate the need to raise or at least reconsider the criticality levels of functions and components identified in earlier criticality analyses. Investigation of vulnerabilities may also identify additional threats, or opportunities for threats, that were not considered risks in earlier vulnerability assessments. Vulnerabilities inform the risk assessment and the countermeasure cost-risk-benefit trade-off.

Discovery of a potentially malicious source from the threat assessment may warrant additional checks for vulnerabilities in other (less-critical) products procured from that source. Therefore, threat assessments can inform vulnerability assessments.

In the Program Protection Plan (PPP) the vulnerability process should be documented at a high level along with the person responsible for the process. The date of the vulnerability assessment,

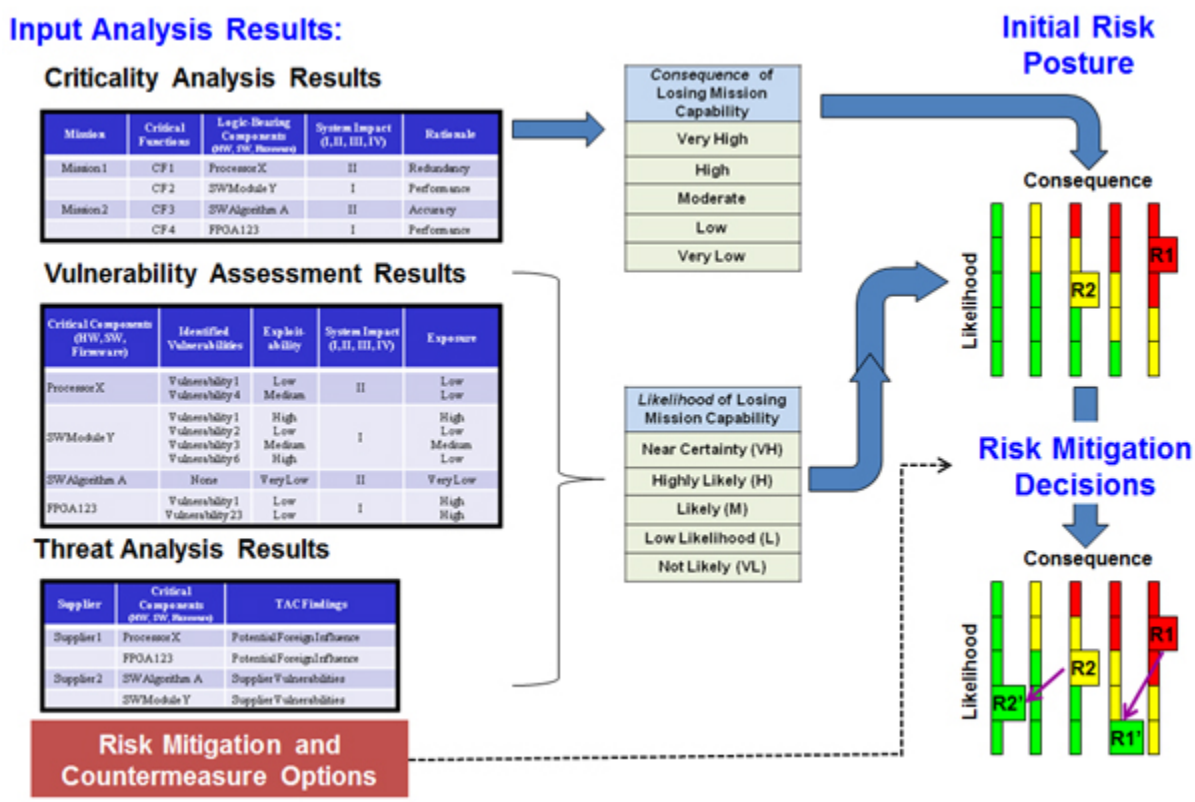
the results of the vulnerability assessment and the planned dates or period of future vulnerability assessments is also recorded in the PPP.

13.6. Risk Assessment

13.6. Risk Assessment

For each Level I and Level II critical function or component the program performs a risk assessment. Figure 13.6.F1 shows the overall risk assessment methodology.

Figure 13.6.F1. Risk Assessment Methodology



The system impact level from the criticality analysis is used to determine the risk consequence. The risk likelihood is based upon the vulnerability assessment and the knowledge or suspicion of threats within the supply chain and potential vulnerabilities within supplied hardware, software, and firmware products. Each Service and program may have specific guidance on how to use the threat assessment and vulnerability assessment to develop the risk likelihood. A basic method which may be used in the absence of program or service specific guidance is described in this section.

One way to translate the threat assessments and vulnerability assessments into risk likelihood or probability is to develop specific questions for supply chain and software assurance. The following paragraphs list two sets of sample Yes/No vulnerability questions that a program can

use to establish the risk likelihood. The first set of vulnerability questions applies to supply chain considerations.

- **Does the Contractor:**
    - Have visibility into lower-level suppliers that provide sub-components used in constructing or assembling critical components?
    - Vet suppliers of critical function components (hardware / software / firmware) based upon the security of their processes?
    - Have processes to verify critical function components received from suppliers to ensure that components are free from malicious insertion (e.g. seals, inspection, secure shipping, testing, etc.)?
    - Have controls in place to ensure technical manuals are printed by a trusted supplier who limit's access to the technical material?
    - Have a process to establish trusted suppliers of critical components?
    - Require suppliers to have similar processes for the above questions?
    - Have processes to limit access to critical components? Can the contractor identify everyone that has access to critical components?
- **Are Blind Buys Used to Contract for Critical Function Components?**
- **Are Specific Test Requirements Established for Critical Components?**
- **Does the Developer Require Secure Design and Fabrication or Manufacturing Standards for Critical Components?**
- **Are Critical Program Information (CPI) and Critical Functions stored, maintained, transported, or transmitted (e.g., electronic media, blueprints, training materials, facsimile, modem) securely?**

The second set of sample Yes / No questions apply to software/ firmware assurance considerations.

- **Does the Developer have:**
    - A design and code inspection process that requires specific secure design and coding standards as part of the inspection criteria?
    - Secure design and coding standards that consider Common Weakness Enumeration (CWE), Software Engineering Institute (SEI) *Top 10* secure coding practices, and other sources when defining the standards?
    - From Common Weakness Enumeration (CWE)
    - Common Vulnerabilities and Exposures (Common Vulnerabilities and Exposures (CVE)
    - Common Attack Pattern Enumeration and Classification (CAPEC)
- **Have software vulnerabilities derived from these three sources been mitigated?**
    - From Common Weakness Enumeration (CWE)
    - Common Vulnerabilities and Exposures (Common Vulnerabilities and Exposures (CVE))
    - Common Attack Pattern Enumeration and Classification (CAPEC)
- **Are static analysis tools used to identify and mitigate vulnerabilities?**
- **Does the software contain Fault Detection/Fault Isolation (FDFI) and tracking or logging of faults?**

- **Do the software interfaces contain input checking and validation?**
- **Is access to the development environment controlled with limited authorities and does it enable tracing all code changes to specific individuals?**
- **Are specific code test-coverage metrics used to ensure adequate testing?**
- **Are regression tests routinely run following changes to code?**

No responses to the questions provide points where a countermeasure may be considered for risk mitigation. A simple way of translating the No responses into a risk likelihood is to map the percentage of No responses to a risk likelihood, such as is shown in Table 13.6.T1 .

Table 13.6.T1 Sample Risk Likelihood Mapping

Number of No Responses	Risk Likelihood
All NO	Near Certainty (VH)
>=75% NO	High Likelihood (H)
>= 25% No	Likely (M)
<= 25% No	Low Likelihood (L)
<= 10% No	Not Likely (NL)

Table 13.6.T2 provides an example of a table that summarizes the vulnerability and threat assessment results used to develop the risk likelihood. A table similar to this is beneficial to the program in understanding the rationale and should be documented in the Risk section of the Program Protection Plan (PPP). The overall likelihood is derived from the supply chain risk likelihood, the software assurance risk likelihood and the threat assessment. The Overall Risk Likelihood may be derived by using a weighted average of the three inputs or using the highest risk. In the example shown in Table 13.6.T2 , the overall risk likelihood of High was derived by applying equal weights for the Supply Chain and Software Assurance Risk Likelihood and the Threat Assessment Risk. The program or service may develop their own specific weightings based upon their program and domain specific knowledge.

Table 13.6.T2 Risk Likelihood Derived From Vulnerability and Threat Assessments

Critical Function Component	Mission Impact	Supply Chain Risk Likelihood	Software Assurance Risk Likelihood	Threat Assessment Risk	Overall Risk Likelihood
Component 1	I	High - No blind buys - No Supply Chain visibility - No supplier qualification process - No receiving verification - No trusted suppliers	Very High - No fault logging - No secure design standard - No static analysis - No Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC) - No input validation - No dev envir ctrl - No Regression test - Low test coverage	Medium	High
Component 2	II	Low - No Supply Chain visibility - No supplier qualification	Not Likely	Medium	Low

The No responses to the questions help to determine the possible countermeasures to be considered for risk mitigation. A similar table may be created which records the countermeasures planned and the new risk probability as a result of the planned mitigations. Table 13.6.T3

provides an example worksheet for planning the countermeasures and the resulting Risk Likelihood.

Table 13.6.T3. Risk Likelihood After Mitigations

Critical Function Component	Mission Impact	Supply chain mitigations	Software assurance mitigations	Threat assessment risk	Overall Risk Likelihood
Component 1	I	- Blind buys - Supply Chain (SC) visibility included in Statement of Work (SOW) - Supplier verification and test of Commercial off-the-shelf (COTS) - Requirement to flow down Statement of Work (SOW) requirements to sub-tier suppliers	- Secure design and coding std included in SOW - Fault logging added - Static analysis added - Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC) used to establish and update secure design standards - Input validation added to interfaces - Development environment control added to limit access and record all access - Regression testing added - Test coverage increased to 60% - Penetration testing added	Medium	Low to Medium

The risk is then incorporated into the program technical risks. The risk entry may look similar to the following example:

Software Assurance Technical Risks		Mitigation Activities
R1. Field-programmable gate array (FPGA) 123 has high exposure to software vulnerabilities with potential foreign influence		Establishing a wrapper to implement secure design standards and fault logging, static analysis, increased test coverage, and penetration testing
Technical Issues		
1. May impact performance, cost, and schedule		
Opportunities		
O1. Low investment, great benefit for program and overall for Missile Programs		Low cost, benefit for program and command

Ensure that the top program protection risks (very high and high) have a risk cube and mitigation plans.

13.7. Countermeasures

13.7.1. Anti-Tamper

13.7.1.1. Critical Technologies (CT)

13.7.1.2. Anti-Tamper Considerations

13.7.1.3. Anti-Tamper Execution

13.7.1.3.1. Process

13.7.1.3.2. Sustainment

13.7.1.3.3. Packaging

13.7.1.4. Anti-Tamper Disclosure Guidelines

13.7.1.5. DoD Anti-Tamper Executive Agent (ATEA) Office

13.7.1.6. Anti-Tamper Verification and Validation (V&V)

13.7.1.7. Anti-Tamper and Verification and Validation (V&V) Plan Approval

13.7.2. Information Assurance (IA)

13.7.2.1. Critical Program Information (CPI) in DoD Information Systems

13.7.2.2. Critical Program Information (CPI) in Other Than DoD Information Systems

13.7.2.3. Indicators of Achieving Baseline Information Assurance (IA) Protection of Critical Program Information (CPI)

13.7.3. Software Assurance

13.7.3.1. Development Process

13.7.3.1.1 Static Analysis

13.7.3.1.2 Design Inspection

13.7.3.1.3 Code Inspection

13.7.3.1.4. Common Vulnerabilities and Exposures (CVE)

13.7.3.1.5. Common Attack Pattern Enumeration and Classification (CAPEC)

13.7.3.1.6. Common Weakness Enumeration information (CWE)

13.7.3.1.7. Penetration Test

13.7.3.1.8 Test Coverage

13.7.3.2. Operational System

13.7.3.2.1. Failover Multiple Supplier Redundancy

13.7.3.2.2. Fault Isolation

13.7.3.2.3. Least Privilege

13.7.3.2.4. System Element Isolation

13.7.3.2.5. Input Checking/Validation

13.7.3.2.6. Software Encryption and Anti-Tamper Techniques (SW load key)

13.7.3.3. Development Environment

13.7.3.3.1 Source Code Availability

13.7.3.3.2. Release Testing

13.7.3.3.3. Generated Code Inspection

13.7.3.3.3. Additional Countermeasures

13.7.4. Supply Chain Risk Management (SCRM)

13.7.4.1. Scope of Supply Chain Risk Management (SCRM)

13.7.4.2. Supply Chain Risk Management (SCRM) Throughout the System Lifecycle

13.7.4.2.1. Criticality Analysis

13.7.4.2.2. Supplier Annotated Work Breakdown Structure (WBS) or System Breakdown Structure (SBS)

13.7.4.2.3. Securing the Supply Chain Through Maintaining Control Over the Information and Information Flow

13.7.4.2.4. Design and Engineering Protections

13.7.4.2.5. Supply Alternatives

13.7.4.2.6. Procurement Authorities

13.7.4.2.7. Enhanced Vulnerability Detection

13.7.5. Trusted Suppliers

13.7.6. System Security Engineering

13.7.6.1. Definitions

13.7.6.2. Context of Systems Security Engineering (SSE) Within SE

13.7.6.3. Systems Security Engineering (SSE) Across the Lifecycle

13.7.7. Security

13.7. Countermeasures

This section describes the guidance and expectations for Program Protection countermeasures. Countermeasures are cost-effective activities and attributes to manage risks to Critical Program Information (CPI) and critical functions and components. They vary from process activities (e.g., using a blind buying strategy to obscure end use of a critical component) to design attributes (e.g., Anti-Tamper design to defeat Critical Program Information (CPI) reverse engineering attempts) and should be selected to mitigate a particular threat. For each countermeasure being

implemented, the program should identify someone responsible for its execution and a time- or event-phased plan for implementation.

Many countermeasures may have to be partially or completely implemented by prime and subcontractors on the program. See Section 13.13 for guidance on contracting for the implementation of Program Protection.

13.7.1. Anti-Tamper

Anti-Tamper (AT) is the Systems Engineering activities intended to deter and/or delay exploitation of critical technologies in U.S. defense systems in order to impede countermeasure development, unintended technology transfer, or alteration of a system. (DoDI 5200.39) Properly fielded Anti-Tamper (AT) should:

- Deter countermeasure development that would decrease U.S. warfighting capabilities
- Prevent the unauthorized or out of baseline augmentation or change of capabilities in Direct Commercial Sales (DCS), Foreign Military Sales (FMS), and International Cooperative Development Programs
- Prevent unauthorized technology changes on Critical Program Information (CPI) or any released capability
- Protect U.S. critical design knowledge and manufacturing processes from unauthorized transfer
- Protect U.S. investment in weapon system capabilities, and avoid additional unplanned investment to regain the U.S. advantage

13.7.1.1. Critical Technologies (CT)

A subset of Critical Program Information (CPI) that specifically resides within a weapon system, training or its support equipment, must be considered for protection by Anti-Tamper (AT) techniques to delay or prevent Reverse Engineering (RE). Critical Technologies can be found in: System hardware, embedded software, application software, and data. Critical Technologies should not be confused or associated with Critical Technology Elements (CTE), in other words, Critical Technologies (CTs) as it apply to Anti-Tamper (AT) is not a matter of maturity or integration level.

13.7.1.2. Anti-Tamper Considerations

Anti-Tamper (AT) is more cost effective when implemented at program onset. Therefore, Anti-Tamper (AT) considerations and techniques should be initiated prior to MS A, during program development, preferably in the program material solution analysis phases:

- The PM should include both Anti-Tamper (AT) requirements and costs in capability development, acquisition and Planning, Programming, Budgeting and Execution (PPBE) process planning cycles.
- Anti-Tamper (AT) requirements may affect other aspects of a program, such as associated maintenance and training devices, and should include end item assessment of

cost, schedule and performance if not considered at program onset.

- Anti-Tamper (AT) requirements should be included (but not limited to) in the following documents; Request for Proposal (RFP), SOO, Statement of Work (SOW), Initial Capabilities Document (ICD), Capability Production Documents (CPD), Capability Development Documents (CPD), Acquisition Strategy (AS), Work Breakdown Structure (WBS), Test and Evaluation Master Plan (TEMP) , Information Assurance Strategy (IAS), Systems Engineering Plan (SEP) , and Systems Engineering Management Plan (SEMP) should be included in the DoD Acquisition systems review process (i.e., Systems Engineering Technical Reviews (SETRs)). Refer to DoD Anti-Tamper (AT) Desk Reference for sample language and further guidance.
- Anti-Tamper (AT) is also applicable to DoD systems during Pre-Planned Product Improvement (P^3I) upgrades as new Critical Technologies (CT) may be added to the system. Additionally, Anti-Tamper (AT) should be specifically addressed in export sales (direct commercial sales, foreign military sales) and international cooperative programs if those systems have Critical Technologies (CT) to protect.
- Anti-Tamper (AT) also involves risk management. The level of Anti-Tamper (AT) should be based on the risk of the loss of U.S. control on the asset containing Critical Program Information (CPI) (level of exposure) and the operational impact (criticality and consequence) if the Critical Program Information (CPI) is lost or compromised. Refer to DoD Anti-Tamper (AT) Guidelines for further guidance.

13.7.1.3. Anti-Tamper Execution

The DoD Anti-Tamper Executive Agent (ATEA) provides support to the PM by helping to determine whether or not to implement Anti-Tamper (AT), per DODI 5200.39. The decision to use or not to use Anti-Tamper (AT) will be documented in a classified annex to the Program Protection Plan (PPP), referred to as the Anti-Tamper (AT) Plan. The Anti-Tamper (AT) Plan includes, but is not limited to, the following information:

- The Program Manager (PM) recommendation and the Milestone Decision Authority (MDA) decision on Anti-Tamper (AT);
- Identification of the Critical Technology (CT) being protected and a description of its criticality to system performance;
- Foreign Teaming and foreign countries / companies participating;
- Threat assessment and countermeasure attack tree;
- Anti-Tamper (AT) system level techniques and subsystem Anti-Tamper (AT) techniques investigated;
- System maintenance plan with respect to Anti-Tamper (AT);
- Recommended Anti-Tamper (AT) solution set to include system, subsystem and component level;
- Determination of how long Anti-Tamper (AT) is intended to delay hostile, or foreign exploitation or reverse-engineering efforts;
- The effect that compromise would have on the acquisition program if Anti-Tamper (AT) were not implemented;
- The estimated time and cost required for system or component redesign if a compromise occurs and;

- Key Management Plan.

13.7.1.3.1. Process

The Anti-Tamper (AT) Process consists of fifteen steps:

1. Identify critical program information
2. Refine results from Step 1 to determine Critical Technologies (CT)
3. Evaluate Critical Technologies (Critical Technologies (CT)) exposure level
4. Evaluate criticality and consequence of compromise of Critical Technologies (CT)
5. Identify Anti-Tamper (AT) protection level requirements
6. Identify potential Anti-Tamper (AT) solution sets considered
7. Describe Critical Program Information (CPI)/Critical Technologies (CT) engineering solution analysis
8. Select initial architecture technique(s) and solution set(s)
9. Identify Anti-Tamper (AT) requirements in the System Functional Baseline
10. Develop Anti-Tamper (AT) architecture
11. Identify Anti-Tamper (AT) implementations in allocated baseline
12. Finalize Anti-Tamper (AT) architecture
13. Implement Anti-Tamper (AT) Architecture and identify residual vulnerabilities
14. Fabricate system and Test (Verification &Validation)
15. Verification and Validation (V&V) results published 60 days prior to deployment

(Consult the DoD Anti-Tamper (AT) Desk reference for further guidance.)

Note: It is highly recommended that the program contact the Component Anti-Tamper (AT) Office of Primary Responsibility (OPR) to obtain a name of a Verification and Validation (V&V) lead. This Verification and Validation (V&V) lead and his team will follow the progress of the Anti-Tamper (AT) Plan implementation and provide consultation. This Verification and Validation (V&V) lead will also determine if the Anti-Tamper (AT) Plan meets the protection level and provide whether the Component Anti-Tamper (AT) Office of Primary Responsibility (OPR) should concur/non-concur with the Anti-Tamper (AT) Plan. This Verification and Validation (V&V) lead will also be the witness to the actual testing of the Anti-Tamper (AT) Plan and provide a memo back to the program as to whether it did complete the Anti-Tamper (AT) testing. The Verification and Validation (V&V) lead and team will be provided to the program at no cost but only as consultants. They will not develop the Anti-Tamper (AT) Plan. That is for the program office/contractor.

13.7.1.3.2. Sustainment

Anti-Tamper (AT) is not limited to development and fielding of a system. It is equally important during life-cycle management of the system, particularly during maintenance. Maintenance instructions and technical orders should clearly indicate the level at which maintenance is authorized; and include warnings that damage may occur if improper or unauthorized maintenance is attempted.

To protect Critical Technologies (CT) during maintenance, it may be necessary, as prescribed by the Delegation of Disclosure Authority Letter, to limit the level and extent of maintenance an export customer may perform. This may mean that maintenance involving the Anti-Tamper (AT) measures will be accomplished only at the contractor or U.S. Government facility in the U.S. or overseas. Such maintenance restrictions may be no different than those imposed on U.S. Government users of Anti-Tamper (AT) protected systems. Contracts, purchase agreements, memoranda of understanding, memoranda of agreement, letters of agreement, or other similar documents should state such maintenance and logistics restrictions. The contract terms and conditions should establish that unauthorized maintenance or other unauthorized activities:

- Should be regarded as hostile attempts to exploit or reverse engineer the weapon system or the Anti-Tamper (AT) measure itself; and
- Should void the warranty or performance guarantee.

Note: The U.S. Government and U.S. industry should be protected against warranty and performance claims in the event Anti-Tamper (AT) measures are activated by unauthorized maintenance or other intrusion. Such unauthorized activities are regarded as hostile attempts to exploit or reverse engineer the system or the Anti-Tamper (AT) measures.

Note: Programs should also plan and budget for Anti-Tamper (AT) maintenance to include government/contractor investigations of tamper events.

13.7.1.3.3. Packaging

Anti-Tamper (AT) affected equipment may need specially designed and approved shipping containers ready upon delivery. The containers should provide the same level of protection from exploitation as the protected Critical Technologies (CT) within the container while in the supply chain or have the Anti-Tamper (AT) equipment active while shipping.

13.7.1.4. Anti-Tamper Disclosure Guidelines

Anti-Tamper (AT) processes and techniques cannot be discussed or revealed to non-U.S. or unauthorized U.S. persons:

- The fact that Anti-Tamper (AT) has been implemented on a specific system is classified as Unclassified/FOUO (For Official Use Only) unless otherwise directed (e.g. specific direction requiring system Anti-Tamper (AT) be handled at a higher classification level, system security classifying system Anti-Tamper (AT) higher)
- The fact that Anti-Tamper (AT) has been implemented on a specific sub-system or even a component of a sub-system is classified SECRET. Refer to the DoD Anti-Tamper (AT) Security Classification Guide (SCG) for further clarification.

13.7.1.5. DoD Anti-Tamper Executive Agent (ATEA) Office

The DoD Anti-Tamper Executive Agent (ATEA) is responsible for all Anti-Tamper (AT) policy consistent with the DoDI 5000.02 and DoDI 5200.39 . The office has established a network of

DoD Component Anti-Tamper (AT) points of contacts (POCs) to assist program managers in responding to Anti-Tamper (AT) technology and/or implementation questions. Additionally, the Acquisition Security Database (ASDB) has been developed as a common shared database of Anti-Tamper (AT) related information.

13.7.1.6. Anti-Tamper Verification and Validation (V&V)

Anti-Tamper (AT) implementation is tested and verified during developmental test and evaluation and operational test and evaluation.

The PM develops the validation plan and provides the necessary funding for the Anti-Tamper (AT) Verification and Validation (V&V) on actual or representative system components. The Verification and Validation (V&V) plan, which is developed to support Milestone C, is reviewed and approved by the DoD Anti-Tamper (AT) Executive Agent , or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR), prior to milestone decision. The program office conducts the Verification and Validation (V&V) of the implemented Anti-Tamper (AT) plan. The Anti-Tamper (AT) Verification and Validation (V&V) team witnesses these activities and verifies that the Anti-Tamper (AT) techniques described in the Anti-Tamper (AT) Plan are implemented into the system and performs according to the Anti-Tamper (AT) plan. The validation results are reported to the Milestone Decision Authority.

13.7.1.7. Anti-Tamper and Verification and Validation (V&V) Plan Approval

The DoD Anti-Tamper Executive Agent (ATEA) has published a DoD Anti-Tamper (AT) and Verification and Validation (V&V) Plan Templates to assist program managers and contractors with the required content for approval. The latest Templates can be downloaded by registered users at https://www.at.dod.mil/ .

There is a two-step approval process for all Anti-Tamper (AT) plans:

Domestic cases:

The Anti-Tamper (AT) Plans (Initial and Final) are to be created by the government or government contractor and approved first by the program manager. Then, they are submitted to the Component Anti-Tamper (AT) Office of Primary Responsibility (OPR) 60 days prior to PDR for initial plans and 60 days prior to Critical Design Review (CDR) for final Anti-Tamper (AT) Plans. The same approval timeline holds true for the verification and validation plans (Initial and Final) if separated from the Anti-Tamper (AT) Plan.

After the program manager has approved the Anti-Tamper (AT) Plan, the Anti-Tamper (AT) Executive Agent or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR), provides an evaluation of the Anti-Tamper (AT) Plan and a letter of concurrence to the program office and Milestone Decision Authority.

Export cases:

1. An Initial Anti-Tamper (AT) Plan MUST be submitted to the DoD Anti-Tamper Executive Agent (ATEA) (or designee) NLT 60 days prior to submission of a Letter of Offer and Acceptance (LOA) or contract signature, whichever comes first. Written DoD Anti-Tamper Executive Agent (ATEA) (or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR)) approval of the Initial Anti-Tamper (AT) Plan must be obtained prior to release of the Letter of Agreement (LOA) or contract signature. As a minimum, the plan should include:
    a. A description of the architecture
    b. The Critical Program Information (CPI) and proviso limit's requiring protection
    c. Level of Anti-Tamper (AT) required for each Critical Program Information (CPI) (1-5)
    d. Top-level solution w/core Anti-Tamper (AT) technology being implemented
    e. Penalty/Response initial thoughts
    f. Support and tamper event reporting concept
    g. Cost and Schedule Rough Order of Magnitude (ROM) for Anti-Tamper (AT) implementation
    h. Risk to implementation
2. An update to the Initial Anti-Tamper (AT) Plan must be provided to the DoD Anti-Tamper Executive Agent (ATEA) (or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR)) within 60 days after contract award.
3. A Final Anti-Tamper (AT) Plan must be submitted to the DoD Anti-Tamper Executive Agent (ATEA) (or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR)) NLT 60 days prior to Critical Design Review (CDR). Written DoD Anti-Tamper Executive Agent (ATEA) (or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR)) approval of the Final Anti-Tamper (AT) Plan must be obtained prior to Critical Design Review (CDR) closure.
4. Verification and Validation (V&V) testing must be completed NLT 60 days prior to system export. Written DoD Anti-Tamper Executive Agent (ATEA) (or Component Anti-Tamper (AT) Office of Primary Responsibility (OPR)) concurrence of satisfactory Verification and Validation (V&V) completion must be obtained prior to system export.

13.7.2. Information Assurance (IA)

Information Assurance (IA) is defined as measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation.

All mission critical functions and components, and information systems storing, processing, or transmitting Critical Program Information (CPI) must be appropriately protected, regardless of whether the information systems are owned and controlled by the Department of Defense or by external entities. Programs with identified Critical Program Information (CPI) need to ensure that the Critical Program Information (CPI) is protected in every computing environment that hosts it, or over which it is transmitted. With the requirement to identify system critical functions and associated components, Information Assurance (IA) needs to determine the Information

Assurance (IA) controls needed for their protection. (See Chapter 7 for further details on Information Assurance (IA) Implementation.)

13.7.2.1. Critical Program Information (CPI) in DoD Information Systems

DoDD 8500.01E and DoDI 8500.2 detail the policy, process, and procedures for implementing appropriate Information Assurance (IA) into DoD information systems. They mandate a controls-based approach, which considers a systems assigned Mission Assurance Category (MAC) and Confidentiality Level (CL) in determining the required robustness of Information Assurance (IA) controls to be implemented. DoD information systems with Critical Program Information (CPI) must be accredited in accordance with DoDI 8510.01 (DIACAP). The DoD Information Assurance Certification and Accreditation Process (DIACAP) establishes a standard process, set of activities, general task descriptions, and a management structure to certify and accredit information systems throughout the system lifecycle. The DoD Information Assurance Certification and Accreditation Process (DIACAP) provides an independent validation process that verifies that appropriate protection measures have been implemented, tested, and maintained, and that any residual risk is at an acceptable level for system operation.

It is important to differentiate between the implementation of Information Assurance (IA) in program support information systems (U.S. Government or contractor) for the protection of Critical Program Information (CPI) as opposed to the implementation of Information Assurance (IA) in the system being acquired. For example, an acquisition program office acquiring a new weapons system may utilize a DoD information system that hosts Critical Program Information (CPI). Similarly, that same program may have Critical Program Information (CPI) being processed or transmitted on the prime contractor or systems integrators design, development, or support systems. The Information Assurance (IA) requirements and certification and accreditation process for each of these support systems are totally separate and distinct from those of the weapons system being acquired, which may also contain Critical Program Information (CPI).

In practice, the implementation of Information Assurance (IA) to protect Critical Program Information (CPI) is no different from the implementation to protect any other information type. DoD information systems with Critical Program Information (CPI) must have both a Mission Assurance Category (MAC) and Confidentiality Level (CL) designated in accordance with DoDD 8500.01E. The presence of Critical Program Information (CPI) may be a factor in the Confidentiality Level (CL) assigned (public, sensitive, or classified), if the Critical Program Information (CPI) sensitivity drives the assignment to a higher level.

13.7.2.2. Critical Program Information (CPI) in Other Than DoD Information Systems

As previously noted, adequate security must be provided to all Critical Program Information (CPI) released to or developed by and in the possession of offeror's, DoD contractors, grantees, or other sharing partners, to include when it is stored or processed on information systems and networks that are not owned by or operated on behalf of the Department. This may be a very diverse group, and may include prime and subcontractors, system integrators, program management support contractors, Federally Funded Research and Development Centers

(FFRDC), and independent test organizations. Critical Program Information (CPI) that is classified must be protected in contractor facilities with Defense Security Service (DSS) accredited information systems in accordance with the National Industrial Security Program Operating Manual (NISPOM). Unclassified Critical Program Information (CPI) that resides on unclassified non-DoD systems must be protected in accordance with Directive-Type Memorandum (DTM) 08-027, Security of Unclassified DoD Information on Non-DoD Information Systems. These requirements should be incorporated as clauses in contracts or change orders, or as appropriate language in sharing agreements and grants.

13.7.2.3. Indicators of Achieving Baseline Information Assurance (IA) Protection of Critical Program Information (CPI)

For DoD information systems containing Critical Program Information (CPI):

- Confidentiality Level (CL) appropriate to the Critical Program Information (CPI) sensitivity and/or classification, and applicable baseline Information Assurance (IA) controls is implemented.
- Authorization to Operate (ATO) or Interim Authorization to Operate (IATO) issued by the hosting systems designated accrediting authority (DAA) is current.
- Information Technology (IT) security plan of action and milestones (POA&M) does not identify any security weaknesses impacting Critical Program Information (CPI) that are not sufficiently mitigated; Information Technology (IT) security plan of action and milestones (POA&M) indicates appropriate level of follow-up actions.
- Inventory of Critical Program Information (CPI) (item, site, system hosting, and Information Assurance (IA) Point of Contact (POC)) is complete.
- Any supplemental Information Assurance (IA) controls specified to protect Critical Program Information (CPI) are incorporated into the DoD Information Assurance Certification and Accreditation Process (DIACAP) implementation plan or equivalent security requirements traceability matrix.

For non-DoD information systems containing Critical Program Information (CPI):

- Required Information Assurance (IA) protection measures are negotiated and agreed to in contract or sharing agreement. Protection requirements flow down through prime to subcontractors, as appropriate.
- For DoD contractor systems accredited under the National Industrial Security Program Operating Manual (NISPOM), accreditation decision issued by the Defense Security Service (DSS) Designated Approving Authority (DAA) is current.
- Reports of Information Assurance (IA) protection self-assessments are submitted to the program office periodically. Reports include appropriate levels of follow-up activity to clear discrepancies. Defense Security Service (DSS) will notify the Government Contracting Agency (GCA) of security compromises or serious security violations involving such systems and of a marginal or unsatisfactory security review rating for the facility.
- Inventory of Critical Program Information (CPI) (item, site, system hosting, and

Information Assurance (IA) Point of Contact (POC)) is complete.

The details of the programs Information Assurance (IA) approach to protecting all Critical Program Information (CPI) and system critical functions and components should be documented in the Countermeasures subsections of the Program Protection Plan (PPP), and should address the content of Sections 13.7.2.1 and 13.7.2.2 , as applicable.

13.7.3. Software Assurance

The extensive use of Commercial off-the-shelf (COTS), Government off-the-shelf (GOTS), open source, and other off the shelf software as well as developmental software in DoD systems necessitates early planning for and design of software security protections that address the threats to those systems and the types of attacks those threats can orchestrate against the systems. Systems must be securely supplied, designed and tested to assure mission success as well as the protection of critical functions, associated components, and Critical Program Information (CPI). Of particular interest are the protection and assurance activities that are undertaken during the COTS integration and development processes, those aimed at mitigating attacks against the operational system (the fielded system), and those that address threats to the development environment. The purpose of this section is to develop a plan and statement of requirements for software assurance early in the acquisition lifecycle and incorporate the requirements into the request for proposal (RFP). Then use that plan to track software assurance protections throughout the acquisition. The progress toward achieving the plan is measured by actual accomplishments/results that are reported at each of the Systems Engineering Technical Reviews (SETRs) and recorded as part of the Program Protection Plan.

The Program Protection Plan (PPP) Outline and Guidance requires acquisition programs to address software assurance responsibilities for the planning and implementation of program protection countermeasures. Such countermeasures address the anticipated attacks a system may experience from the threats it will face by eliminating or reducing vulnerabilities. The countermeasures are selected with an understanding of which parts of the software are the most critical to the success of the mission. The plan includes a sample Software Assurance Countermeasures Table, which summarizes the planned and current state of a programs software assurance activities. The table is also used as part of a vulnerability assessment to identify operational, developmental, design, COTS and software tool vulnerabilities that that can be addressed by planning and implementing software assurance countermeasures.

The table in the PPP is divided into 3 sections that provide different vulnerability and countermeasure perspectives on software assurance plans and implementation:

- Development Process assurance activities conducted during the development process to mitigate and minimize attacks (e.g., threat assessment and modeling, attack surface analysis, architecture and design reviews, application of static and dynamic code assessment tools and services, penetration testing, and red teaming) that the developed system is likely to face when deployed into operation
- Operational System attack countermeasures and other assurance activities applied within the operational environment (e.g., failover, fault isolation, encryption, application

firewalls, least privilege, and secure exception handling) to mitigate attacks against the delivered system and software interfaces, which may include COTS, GOTS, open source, and other off the shelf software
- Development Environment assurance activities and controls (e.g., access controls, configuration management, and release testing) applied to tools and activities (e.g., compilers, linkers, integrated development environments, run-time libraries, and test harnesses) used to develop and sustain software to mitigate attacks

Given the constraints of cost, schedule, and performance, fully comprehensive assessment and testing is often not feasible. Thus SwA planning should reflect priorities chosen to mitigate risk and deliver mission capability with acceptable levels of assurance. The coding language, source of code (i.e. custom, COTS, GOTS, open source), platform (i.e. web based, mobile, embedded, etc.) as well as the results of criticality analysis (see 13.3.2.1) will be used to prioritize software assurance activities when planning for SwA.

13.7.3.1. Development Process

The purpose of this section of the table is to measure and explicitly capture the assurance activities conducted during software development and the integration of off-the-shelf components. As appropriate to the risk of compromise and criticality of the software in question, PMs are to analyze the development activities for:

- potential introduction of vulnerabilities and risks based on the anticipated threat and the attacks the threats are capable of making against the system;
- development of a plan for the assurance process as well as the technical disciplines and knowledge needed for Integrated Project Teams (IPTs);
- how IPTs address the architecture, design, code, and implementation choices to include the appropriate mitigations necessary to address the anticipated attacks and assure the critical function software components; and
- review points to track/assess the progress at the milestones in the Program Protection Plan.

Not all software will require the same level of software assurance activities and mitigation planning and implementation in programs with millions of lines of code, there may be some functions (perhaps a monthly reporting feature) that are less mission-critical than other (perhaps a satellite station-keeping module). It may also be difficult to perform some types of assessment and mitigation activities on COTS software for which the source code is not available. Note that in such cases software related risks still exists and may be unmitigated. The software assurance table in the PPP recognizes these varying types of software and allows for differing plans/implementation of assurance as needed.

13.7.3.1.1 Static Analysis

Programs should investigate the applicability of automated static analysis tools to review source and/or binary copies of their software and, where advantageous, apply both static source code and static binary analysis to assist in identifying latent weaknesses that would manifest as

operational system vulnerabilities and allow attackers to interfere, manipulate, or otherwise suborn the systems mission capabilities. The use of these types of tools within the development activity (i.e., as an add-on to the developers Integrated Development Environment (IDE)) as well as in the Independent Test and Evaluation (IT&E) activities are both valuable and useful. Approaches that integrate such forms of continuous assessment into the developers activities should be emphasized and encouraged.

13.7.3.1.2 Design Inspection

The establishment and update of secure design and code standards by the program should address the potential types of attacks the system would face and draw upon DoD, Government, Federally Funded Research and Development Centers (FFRDC), academia, commercial web sites and industry sources for mitigation approaches and methods to address those that could impact the systems mission capabilities. The list of attack patterns captured in the Common Attack Pattern Enumeration and Classification (CAPEC) collection can be used to help consistently analyze a system for potential types of attacks they may face and to bring consistency into the validation activities when the program is verifying that the design and coding standards are being followed.

13.7.3.1.3 Code Inspection

Due to the subtle nature of most weaknesses in code that lead to unreliable, insecure, and brittle applications that are easily influenced by attackers it is important that code inspections utilizing tools be part of the approach used to minimize these weaknesses. There are over 700 documented types of weaknesses in code, design, architecture, and implementation captured in the Common Weakness Enumeration (CWE) catalog but not all of them are equal threats to any specific application or system. Programs may wish to draw upon secure design and coding approaches defined on websites such as top 10 secure coding practices (https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices) and the Common Weakness Enumeration (CWE)/ SysAdmin, Audit, Network, Security (SANS) top 25 most dangerous software errors (http://cwe.mitre.org/top25/index.html) to establish and update their secure design and coding standards. As a minimum the code inspection is used to inspect for conformance to the secure design and coding standards established for the program.

An important part of the code inspection is to identify the subset of the overall CWE collection to focus on initially. Alternate approaches to focusing in on a subset of the weaknesses are described in the CWE paragraph below (13.7.3.1.6.) and the CAPEC paragraph (13.7.3.1.5.) . These approaches can be used independently or in combination if desired.

Because of the dynamic nature of the threat environment and information about how systems can be compromised through software weaknesses, the program should have a methodology to periodically update their secure design and coding standards so that reviews using them address new types of attacks and types of weaknesses.

The next three sections of this document describe the middle three columns of the PPP Software

Assurance Table, which are meant to capture how the established vulnerability (CVE), weakness (CWE), and attack pattern (CAPEC) collections are being used by the project team to identify and mitigate the most dangerous types of vulnerabilities in the software. These columns are further defined below but the most critical part of completing these three columns is the analysis of which CVEs, CWEs, and CAPECs should be used as the denominator of these percentage calculations and the documentation within the project team of the rationale and methodology followed in determining those lists and keeping them current throughout the project as the system design, development and testing progresses and the threat environment and other factors change.

13.7.3.1.4. Common Vulnerabilities and Exposures (CVE)

Common Vulnerabilities and Exposures (CVE) information is used to identify, track, and coordinate mitigation activities of the publicly known vulnerabilities in commercial (COTS) and open source software which are often used by threats actors/agents to attack systems. Programs that incorporate COTS software into their systems should perform regular searches of the CVE lists before purchase and throughout the software lifecycle to understand vulnerabilities in those COTS software components and assess potential threat to mission success.

The CVE list is a compilation of publicly known information about security vulnerabilities and exposures. The list is international in scope, free for public use, and referenced in most commercial tools that scan operational systems and networks for vulnerabilities. The CVE list can be used to identify publicly known software vulnerabilities that could:

- Allow an attacker to execute unauthorized code or commands;
- Allow an attacker to gain privileges or assume identities;
- Allow an attacker to access and/or manipulate data that is contrary to the specified access restrictions for that data;
- Bypass protection mechanisms:
- Allow an attacker to hide their activities; and
- Allow an attacker to conduct denial of service attacks.

CVE is intended for use by security experts, so it assumes a certain level of knowledge. Programs should use a tool during incremental software testing of their commercial and open source packages that scans those operational components and matches the results with the CVE dictionary. Alternately, a scan of the affected software packages on the CVE list can be used to review the list for any publicly known vulnerabilities for the software packages being used by a DoD program. A list of CVE compatible tools is available at http://cve.mitre.org/compatible/product.html .

The CVE column in the Program Protection Plan Software Assurance table reports the planned and actual percentages of software components that incorporate COTS or open source that have been analyzed and acceptably remediated against the CVEs from the CVE list that apply to those COTS and open source packages.

Supportive analysis by the project team must record the CVEs found, the remediation applied,

and the residual risk to the mission of any unresolved CVEs. To identify which CVEs should be included in the analysis the list of CVEs for each COTS product and open source should be tracked and those that were remediated documented as such. For each COTS and open source package utilized as part of the system, the project staff should determine whether an explicit vulnerability advisory/alert activity is provided/offered by the provider/developer of those packages.

For those that do not provide publicly available advisories/alerts about security issues that need to be resolved the project staff should carefully consider the risk they are inheriting from that developer by not providing patch information in a manner that CVE identifiers can be assigned. Without CVE identifiers it is much harder to track and manage the state of deployed software within the DoD's vulnerability management practice and the automation tooling deployed within the DoD. 100% of developmental Critical Program Information (CPI) software and developmental critical-function software packages, whether COTS or open source, must be evaluated using CVE, to surface exposures inherited by incorporating open source or COTS libraries or products.

Guidance on searching the CVE is located at http://cve.mitre.org/about/faqs.html#c . An important aspect of applying CVE tools and reviews to a collection of COTS and open source is to apply the Common Vulnerability Scoring System (CVSS) to the determination of which CVEs to mitigate first and to understand the severity of the remaining CVEs.

If the selected tool outputs any CVE with a CVSS score above medium (4), programs should mitigate the vulnerability with highest priority first and then work through the next highest priority issue until the residual risk represented by the remaining vulnerabilities is acceptable to the mission owner. CVEs that are included in any DoD Information Assurance Vulnerability Management (IAVM) alerts and advisories should be addressed in accordance to the priorities and timeframe included in the IAVM from DISA.

The CVE web site is at http://cve.mitre.org

13.7.3.1.5. Common Attack Pattern Enumeration and Classification (CAPEC)

Common Attack Pattern Enumeration and Classification (CAPEC) is meant to be used for the analysis of common patterns of attacks against systems, whether for understanding how attacks are done, scoping of relevant threats, templates for malicious testing, or as a foil for thinking about the susceptibility of systems architecture, design, and technical implementation to specific attacks.

CAPEC is international in scope, free for public use, catalog of attack patterns outlining information such as a comprehensive description of the phases and steps in attacks, the weaknesses they are effective against (using CWEs), and a classification taxonomy that can be used for the analysis of common attack patterns. CAPEC attack patterns cover a wide variety of families of attacks including: data leakage attacks, resource depletion attacks, injection attacks, spoofing attacks, time and state attacks, abuse of functionality attacks, attacks using probabilistic techniques, attacks exploiting authentication, attacks exploiting privilege/trust, attacks exploiting

data structure, resource manipulation attacks, network reconnaissance, social engineering attacks, as well as some physical security attacks and supply chain attacks.

The attack patterns in CAPEC can be a powerful mechanism to capture and communicate the attacker's perspective, organize the analysis of a system with respect to attacks, and prioritize weaknesses (CWEs) based on the anticipated attack patterns. They are descriptions of common methods for exploiting software. Identified attack patterns may influence the selection of the COTS and open source software products, programming languages, and design alternatives. By understanding the attackers perspective and how a programs software is likely to be attacked, programs can directly consider these exploit attempt methods and mitigate them with design, architecture, coding and deployment choices that will lead to more secure software.

Programs should identify the set of attack patterns that pose the most significant risk and leverage them at each stage of the Software Development Lifecycle (SDLC). A discussion of how to use CAPEC in this manner is available on the Engineering for Attack page on the CWE site (http://cwe.mitre.org/community/swa/attacks.html). This is the same basic methodology described in the new *ISO/IEC Technical Report 20004, "Refining software vulnerability analysis under ISO/IEC 15408 and ISO/IEC 18045"* , which describes an alternate approach for doing a vulnerability analysis of a software-based system under the Common Criteria regime. ISO/IEC 15408 and ISO/IEC 18045 are the two standards that guide and describe the Common Criteria evaluation methodology.

Basically that page describes how an analysis using attack patterns to represent the expected threat and identify the subset of weaknesses that are of most concern, can be used to identify which weaknesses those attacks would be effective at exploiting and that list can be used to influence the choices about design and architecture, considering the planned operational use, the creation of security policies, requirements, and thinking through the risks related to the systems intended use. This list of the weaknesses, the ones that are exploitable by the attack patterns the systems adversary are capable of using against the system can be used to identify the subset of relevant CWE weaknesses to avoid and to vet for during implementation. The lists associated CAPECs can be used to guide the software testing by identifying high priority test cases that should be created for risk-based security testing, penetration testing, and red teaming. [1]

The CAPEC column in the Program Protection Plan Software Assurance table reports the planned and actual percentages of developed software components that have been evaluated utilizing the attack patterns from the CAPEC list to identify the appropriate sub-set of CWEs, to consider alternate design and architectures or implementations, or to drive the creation of appropriate misuse and abuse test cases.

Supportive analysis by the project team must record the CAPECs identified as germane to the system, the CWEs identified as being susceptible to those CAPECs and the remediation applied along with an understanding of the residual risk to the mission of any CWEs that weren't tested by simulating CAPECs against the system. To identify which CWEs should be included in the testing analysis based on CAPEC inspired test cases the list of CWEs reviewed for the static analysis tools/services should be tracked and those that were identified, covered by the analysis tool/service and appropriately remediated should be documented as such. For each CWE that

was not covered by a static analysis tool/service, the project staff should determine whether an appropriate CAPEC inspired test case or Red Team activity was conducted without finding an exploitable CWE.

For those CWEs that were not covered by static analysis or testing, the project staff should carefully consider the risk to the mission from the potential of those weaknesses remaining in the system. Without demonstrable evidence that the CWEs that an attacker could exploit are mitigated there will always be some level of risk but it is incumbent on the project staff to document this residual risk for the end user so they can manage that risk when the system is deployed within the DoD. 100% of developmental Critical Program Information (CPI) software and developmental critical-function software should be evaluated against the CAPEC list.

The CAPEC web site is http://capec.mitre.org. A description of the CAPEC schema is located in the Documentation portion of the CAPEC Documents page at http://capec.mitre.org/about/documents.html .

13.7.3.1.6. Common Weakness Enumeration information (CWE)

The Common Weakness Enumeration (CWE) is international in scope and free for public use. CWE provides a unified, measurable set of software weaknesses to enable more effective discussion, description, selection, and use of software security tools and services to find weaknesses in source code and operational systems components as well as to better understand and manage software weaknesses related to architecture and design.

CWE is targeted to developers and security practitioners. Programs should use CWE-compatible tools to scan software for CWE. A list of CWE-compatible products is available at http://cwe.mitre.org/compatible/product.html .

The CWE column in the table reports the planned and actual percentages of developed software components that have been evaluated utilizing the weaknesses from the CWE list to identify the appropriate sub-set of CWEs, to consider alternate design and architectures or alternate coding constructs.

Supportive analysis by the project team must record the subset of CWEs identified as being most germane to the secure operation of the system. The subset of CWEs can be taken from the CWE/SANS Top 25 Most Dangerous Software Errors list or by utilizing the Common Weakness Risk Analysis Framework (CWRAF) to identify the subset of CWEs that are the most dangerous to the systems mission given what the software is doing for the mission. CWRAF allows a project team to create their own list of the most dangerous CWEs based on the specifics of their system and which failure modes are the most important to mitigate/prevent.

The CWE/SANS Top 25 Most Dangerous Software Errors list on the CWE and SANS Web sites provides detailed descriptions of the top 25 programming errors along with authoritative guidance for mitigating and avoiding them.

The CWRAF methodology is described on the CWE web site and numerous examples are

provided to help a project team learn how to apply the methodology to their system in combination with the Common Weakness Scoring System (CWSS).

By using the Common Weakness Scoring System (CWSS) a program can also reflect their specific list of dangerous CWEs into their tools so the risk to the mission of the weaknesses found during static and dynamic analysis or penetration testing reflects the relative importance of those impacts.

The CWE web site is at http://cwe.mitre.org and the CWSS web page is at http://cwe.mitre.org/cwss/ .

Additionally, the project team should have a documented understanding of the residual risk to the mission of any CWEs that weren't reviewed for by static analysis tools/services or tested by simulating the CAPECs that would be effective against those CWEs. For CWEs deemed to be dangerous but not covered by a static analysis tool/service, the project staff should determine whether an appropriate CAPEC inspired test case or Red Team activity was conducted without finding an exploitable CWE.

For those CWEs that were not covered by static analysis or testing, the project staff should carefully consider the risk to the mission from the potential of those weaknesses remaining in the system. Without demonstrable evidence that the CWEs that an attacker could exploit are mitigated there will always be some level of risk but it is incumbent on the project staff to document this residual risk for the end user so they can manage that risk when the system is deployed within the DoD. 100% of developmental Critical Program Information (CPI) software and developmental critical-function software should be evaluated against the identified subset of the CWE list.

In addition to the above listed MITRE websites, PMs should consider best practices identified at http://www.safecode.org/index.php .

13.7.3.1.7. Penetration Test

Programs should report what portion of the system will undergo penetration testing. The purpose of penetration testing is to subject the system to an attack exercise to raise awareness of exploitable vulnerabilities in the system and accelerate their remediation. Also the knowledge that a system will undergo penetration testing increases the vigilance of the software engineers responsible for architecting, designing, implementing, and fielding the systems.

The text should support the number with brief an explanation of the penetration testing performed and a reference to any supporting reports generated by that testing.

The unit's used for planned/actual percentages for this metric are at the discretion of the program. They should be explained in the text and be meaningful and provide insight into the completeness of the testing. For example a network that exposes a certain number of protocols may measure the percentages in the space of protocol states. A system with an API may measure

the number of interface functions probed.

13.7.3.1.8 Test Coverage

Programs should report on their planned and actual test coverage. Unit's and metrics for test coverage are at the discretion of the program, but should be meaningful and yield insight into the completeness of the testing regimen.

Possible measure for test coverage include percentage of statements exercises, percentages of API calls and exception conditions exercised, number of function points tested.

13.7.3.2. Operational System

This section refers to the software and firmware on the fielded system. Software assurance countermeasures is a rapidly evolving area. Successful assessments, techniques, applications, and example outcomes are frequently published in papers that can be found at DoD, Government, Funded Research and Development Centers (FFRDC), and commercial web sites. The FFRDC Carnegie Mellon Software Engineering Institute (SEI) and MITRE both have searchable libraries containing information about the approaches to Software Assurance indicated in the *Program Protection Plan Outline &* Guidance , Table 5.3.3-1 Application of Software Assurance Countermeasures.

13.7.3.2.1. Failover Multiple Supplier Redundancy

Identical code for a failed function will most likely suffer the same failure as the original. For redundancy in software, therefore, a completely separate implementation of the function is needed. This independence reduces the probability that the failover code will be susceptible to the same problem.

13.7.3.2.2. Fault Isolation

Design principles applied to software to isolate faults, include functions to trap, log, and otherwise protect element failures from affecting other elements and the larger system. Logs help trace the sources of operational faults. Logs can also be examined to determine whether there was a malicious attack.

Programs reporting a Yes in the table should be prepared elaborate with technical detail on how the fault isolation mechanisms were employed in the architecture and design for the particular component or sub-system. Fail over or fault isolation is also where the logging of the failure event and the capture of relevant data needed to determine root cause of the failover event is best included.

13.7.3.2.3. Least Privilege

Design principle applied to software that limits the number, size, and privileges of system elements. Least privilege includes separate user roles, authentication, and limited access to

enable all necessary functions but minimize adverse consequences of inappropriate actions.

Programs reporting a Yes in the table should be prepared elaborate with technical detail on how least privilege principles were employed in the architecture and design for the particular component or sub-system.

13.7.3.2.4. System Element Isolation

Design principles applied to software to allow system element functions to operate without interference from other elements.

Programs reporting a Yes in the table should be prepared elaborate with technical detail on how system element isolation principles were employed in the architecture and design for the particular component or sub-system.

13.7.3.2.5. Input Checking/Validation

The degree to which software element inputs are checked and validated according to defined criteria and functionality. Input checking and validation should ensure that out-of-bounds values are handled without causing failures and the invalid input events are logged

Programs reporting a Yes in the table should be prepared to elaborate on the architectural and design criteria governing the extent of input checking/validation employed.

13.7.3.2.6. Software Encryption and Anti-Tamper Techniques (SW load key)

The degree to which executable software code is encrypted or otherwise protected (e.g., by checksums or cyclic redundancy checks) from corruption, between factory delivery and use in a military mission. Defense Acquisition University (DAU) currently teaches an anti-tamper course, which provides some Anti-Tamper (AT) techniques that can be used for software encryption.

Programs reporting a Yes in the table should be prepared to elaborate on specific anti-tamper techniques are included in the architecture, design, and implementation of the software component or sub-system and what risks they are intended to mitigate.

13.7.3.3. Development Environment

Software tools used in the development environment (as opposed to the actual fielded software) are another source of risk to warfighting capability and should be considered in the Program Protection Plan (PPP). In particular a compromised development environment could be leveraged by an attacker to insert malicious code, exploitable vulnerabilities, and/or software backdoors into the operational software before it is fielded.

Examples of software development tools include:

- Compilers, assemblers, pre-compilers, and other code generating tools such as design

templates
- Structured code editors and pretty printers
- Code static analysis tools
- Debugging and timing analysis tools
- Code configuration management tools
- Accounts and access controls on development computers and networks
- Test management tools, test data generators, test harnesses, automated regression testing tools

Examples of compromising tools to achieve malicious insertion include

- Modify compiler to generate or insert additional functionality into the operational code
- Modifying a math library of routines with malware that then gets incorporated into the operational code.

Programs should tailor the list contents of the SW Product column in this section of the table to enumerate the software tools pertinent to the programs development environment(s). For each SW product listed table entries should address the items enumerated in the following columns.

13.7.3.3.1 Source Code Availability

When source code is available, it becomes easier to answer some questions about the behavior of the tool and detect potential compromise.

Is source code available for the tool? A simple yes or no should suffice. If further information (e.g. coding language, code size, licensing cost constraints) would provide useful insight annotate the entry with a note.

13.7.3.3.2. Release Testing

Software tools are often updated. These updates are a potential path for an attacker to compromise the development environment and thus the operational software.

Indicate whether testing for indications of malicious insertion or tool compromise are performed on each update of the tool before that update is incorporated into the development environment.

13.7.3.3.3. Generated Code Inspection

Indicate whether/how any generated code for the system is examined for malicious code or exploitable vulnerability potentially inserted by the software tool in question.

In general, the problem of how to effectively inspect generated code for malicious insertion remains an open area of research. From the practical standpoint, it is better to perform some inspection than to ignore the problem entirely. That at least raises the bar for what an attacker needs to do compromise the system undetected.

Potential code inspection countermeasures include:

- Manual inspection of a representative sample of the generated code
- Analysis of the code with reverse engineering tools
- Identification of the libraries compiled into an executable
- A sanity check of components output by the tools against components expected
- Comparison to baselines generated by previous versions of the tool.
- Manual inspection of tool outputs against a known/analyzable test corpus.
- Advanced/experimental techniques such as automated function extraction.

Note that in many instances simple sanity checks can be effective in detecting some injected malware. For example: extracting, comparing and sorting strings might point to a trigger string used to open a backdoor. Decompiling an executable may reveal the presence of OP codes not normally generated by the compiler.

Where generated code inspection is deemed of benefit programs should tailor the form of inspection to the unique aspects of the program and report planned and actual percentages appropriately.

13.7.3.3.4. Additional Countermeasures

Programs should consider adding additional columns to this area of the software assurance table with the rationale for the additions if programs judge them to significantly reduce the risk of malicious insertion.

Additional countermeasures may include:

- Access controls and other controls detect malicious behavior or suspicious artifacts in the development environment?
- Information assurance controls to safeguard technical data in the development environment (networks, computers, test equipment, and configuration systems)?

Controlling and accounting for printing of technical manuals and other documentation.

---

[1] http://capec.mitre.org/documents/An_Introduction_to_Attack_Patterns_as_a_Software_Assurance_Knowledge_Resource.pdf

13.7.4. Supply Chain Risk Management (SCRM)

This section describes how a program can manage supply chain risks to critical program information and critical functions and components with a variety of risk mitigation activities throughout the entire system lifecycle. The Supply Chain Risk Management (SCRM) guidance in this section identifies references that establish a sample of practices for managing supply chain risks. As Supply Chain Risk Management (SCRM) techniques and practices continue to evolve, additional guidance will be published to refine the Departments understanding and

implementation of Supply Chain Risk Management (SCRM). There are a variety of existing resources available to aid in the understanding and implementation Supply Chain Risk Management (SCRM). The following is a list that includes, but is not limited to the following foundational documents:

- DTM 09-016 Supply Chain Risk Management (SCRM) to Improve the Integrity of Components Used in DoD Systems - Establishes authority for implementing Supply Chain Risk Management (SCRM) throughout DoD and for developing initial operating capabilities.
- DoD Supply Chain Risk Management (SCRM) Key Practices and Implementation Guide Provides a set of practices that organizations acquiring goods and services can implement in order to proactively protect the supply chain against exploitation, subversion, or sabotage throughout the acquisition lifecycle.
- National Defense Industrial Association (NDIA) System Assurance Guidebook Provides guidance on how to build assurance into a system throughout its lifecycle, organized around the Undersecretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) Life Cycle Management Framework.
- DoD Instruction O-5240.24, Counterintelligence (CI) Activities Supporting Research, Development, and Acquisition (RDA)

13.7.4.1. Scope of Supply Chain Risk Management (SCRM)

Currently, Program Protection Planning Supply Chain Risk Management pertains to Information and Communications Technology (ICT). In the digital age where supply chains are distributed globally, and design, manufacturing and production often occur internationally, supply chains have a greater exposure to threats and exploitation.

Supply chain risk management provides programs with a framework for analyzing all the risks associated with the supply chain, which enables the determination of what risks may be mitigated, and what risks may be accepted. This determination will vary based on the purpose and mission being performed. Applying Supply Chain Risk Management (SCRM) early in the production lifecycle will allow for earlier mitigations and a more strategic approach for managing risk.

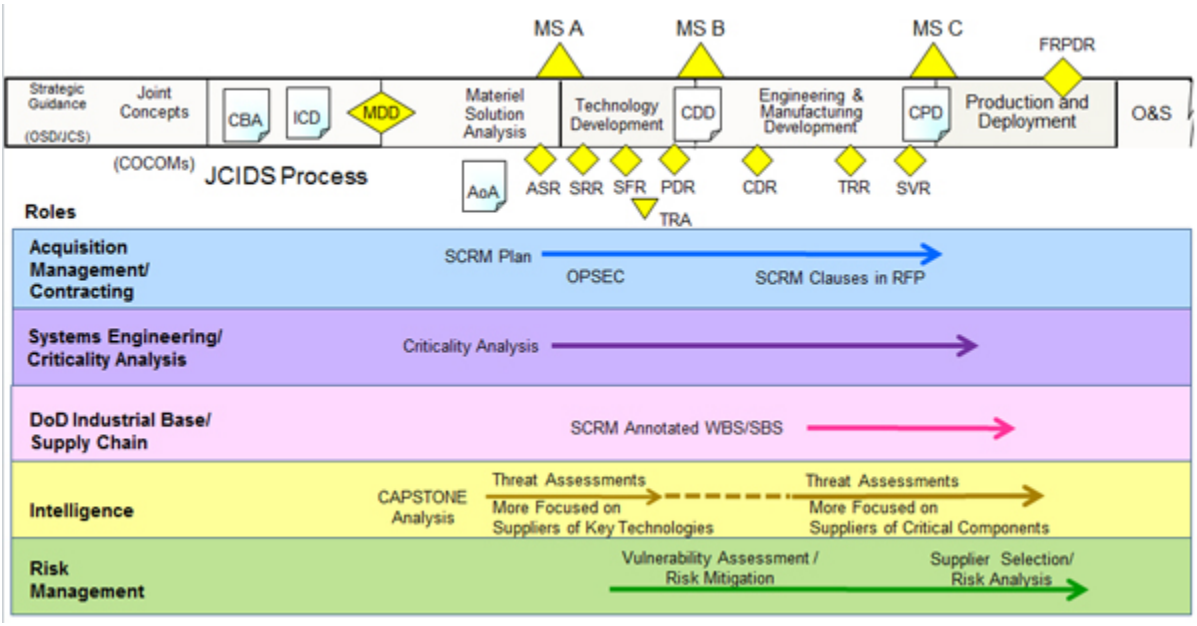
13.7.4.2. Supply Chain Risk Management (SCRM) Throughout the System Lifecycle

The protection of mission-critical systems (including the information technology that compose those systems) must be a priority throughout the entire system development life cycle (i.e., during design and development, manufacturing, packaging, assembly, distribution, system integration, operations, maintenance, and disposal). This is accomplished through threat awareness and by the identification, management, and potential elimination of vulnerabilities at each phase of the life cycle using complementary, mutually reinforcing strategies to mitigate risk.

Figure 13.7.4.2.F1 illustrates how key Supply Chain Risk Management (SCRM) activities align with the steps in the DoD Acquisition Lifecycle. Activities are organized by the various roles

that should perform the indicated functions/procedures. Due to the multidisciplinary nature of Supply Chain Risk Management (SCRM), Program Protection requires careful planning and coordination across multiple stakeholders.

Figure 13.7.4.2.F1. Supply Chain Risk Management (SCRM) Activities throughout the System Lifecycle



Mitigation of supply chain risks is most effective when identification and implementation occur early in a programs acquisition planning and contracting. Generally, mitigation choices narrow and become more expensive the further into the lifecycle they occur. Given the amount of information and supply choices that are present in the marketplace, Operations Security (OPSEC) related to the acquisition process for programs is vital.

13.7.4.2.1. Criticality Analysis

Information on Criticality Analysis can be found in Section 13.3.2.1 .

13.7.4.2.2. Supplier Annotated Work Breakdown Structure (WBS) or System Breakdown Structure (SBS)

A cornerstone for the identification of supply chain risks and the development of supply chain risk management strategies and mitigations for critical components is the criticality analysis. The Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) may be used to annotate the suppliers of the critical components identified by the criticality analysis. A Supplier-Annotated Supply Chain Risk Management (SCRM) Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) is a helpful tool to assist with tracking and managing the supply chain risks. The Supply Chain Risk Management (SCRM) Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) is a detailed breakdown identifying all system

assemblies, subassemblies and components and their suppliers for, at a minimum, all critical components identified through criticality analysis. The Supply Chain Risk Management (SCRM) Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) may also include alternative suppliers for all critical components down to the Commercial off-the-shelf (COTS)-item level, with the cost, schedule, and performance impact data for each alternative. Although the Supply Chain Risk Management (SCRM) Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) is not a current requirement, it may be an effective way to record, track and manage the potential suppliers of critical functions as the trade-offs analysis between security, performance, and cost is examined.

The Supply Chain Risk Management (SCRM) System Breakdown Structure (SBS) may provide insight into any teaming arrangements based on an understanding of the defense industrial base and subsequent supply chain. Prior to Milestone B, manufacturers typically develop their supplier lists and enter into teaming agreements. Because of that, programs may consider requiring oversight and input into any supplier teaming arrangements. The program could put controls in place so that supplier lists provide alternatives/alternative suppliers for critical components. Between Preliminary Design Review (PDR) and Critical Design Review (CDR), the Supply Chain Risk Management (SCRM) Work Breakdown Structure (WBS) or System Breakdown Structure (SBS) should be provided by suppliers to the government for review and vulnerability/risk assessment. It is essential that the DoD work with potential Prime Contractors to develop supplier lists and gain insight to potential teaming arrangements. This input is supported by contract clauses such as Consent to Subcontract.

13.7.4.2.3. Securing the Supply Chain Through Maintaining Control Over the Information and Information Flow

OPSEC

Sensitive information must be protected from suppliers and potential suppliers. Operations Security (OPSEC) and appropriate classification guides should be employed to protect system supply chain, design, test and other information from potential adversaries. This includes limiting the sharing of information with suppliers and potential suppliers at multiple tiers sufficient to manage risk. Confidentiality of key information (such as user identities, element uses, suppliers, and their processes, requirements, design, testing, etc.) must be protected when critical to mission success.

Provenance

It is important to establish and maintain the origin, development, delivery path, and mechanisms to protect the integrity of critical components, tools, and processes, as well as their associated changes, throughout the lifecycle. This enables accurate supply chain (SC) risk assessment and mitigation, which requires accurate information on the origin of components, how they are developed, how they are delivered throughout the supply chain. This includes strong system and component configuration management to ensure traceability against unauthorized changes. Selecting suppliers who maintain provenance is the first step to reducing supply chain (SC) risks.

13.7.4.2.4. Design and Engineering Protections

Once critical functions and components have been identified, design and engineering protections can be employed to reduce the attack surface and reduce what is considered critical. These protections should further protect intrinsically critical functions and reduce existing unmediated access to them.

System elements may still have unintentional or intentional vulnerabilities (whether in isolation or when combined) even if they all come from trustworthy suppliers. Defensive Design helps reduce the attack surface and limit the exposure of vulnerabilities. Defensive approaches reduce opportunities to expose or access an element, process, system, or information and to minimize adverse consequences of such exposure or access. In particular, defensive design should be used to protect critical elements and functions by reducing unnecessary or unmediated access within system design.

13.7.4.2.5. Supply Alternatives

Application-specific integrated circuits (ASICs) should be acquired from a trusted supplier because if they are compromised, then unique DoD designs could be exposed and critical system information could become available to adversaries. For information on trusted suppliers of microelectronics, please refer to Sect ion 13.7.5 .

13.7.4.2.6. Procurement Authorities

Procurement language has been developed and is available for use to help mitigate supply chain risk through contractual requirements in the Statement of Work (SOW). Refer to Section 13.13.1.2 below for suggested language.

Supplier Security Practices

Organizations can help mitigate supply chain risk down the contract stack by requiring and encouraging suppliers and sub-suppliers to use sound security practices and allow transparency into processes and security practices. It is recommended that contract vehicles should require, encourage, or provide incentives for suppliers to deliver up-to-date information on changes that affect supply chain (SC) risk, such as changes in their suppliers, locations, process, and technology.

Use of the acquisition and procurement process early in the system lifecycle is a key way to protect the supply chain by defining and creating supplier requirements and incentives; using procurement carve-outs and controlled delivery path processes; and using all-source intelligence in procurement decisions. Source selection criteria and procedures should be developed in order to encourage suppliers to provide detailed visibility into the organization, elements, services, and processes. Other procurement tools may be available to manage the criticality of components and address risk in acquisition planning and strategy development.

13.7.4.2.7. Enhanced Vulnerability Detection

Due diligence analysis for items of supply is performed to counter unintentional vulnerabilities, intentional vulnerabilities (e.g., malicious wares and processes), and counterfeit's. It may include software static analysis, dynamic analysis (including the use of simulation, white and black box testing), penetration testing, and ensuring that the component or service is genuine (e.g., tag, digital signature, or cryptographic hash verification). Tools can include development, testing and operational tools.

Refer to Section 13.7.3 for more information on Software Assurance.

13.7.5. Trusted Suppliers

In the context of Program Protection, trusted suppliers are specific to microelectronic components and services. The Department is currently developing new policy on the criteria for using trusted suppliers. This content will be updated when that policy is published.

Defense Microelectronics Activity (DMEA) maintains a list of accredited suppliers on its website at http://www.dmea.osd.mil/trustedic.html .

13.7.6. System Security Engineering

13.7.6.1. Definitions

System Security Engineering (SSE) : An element of system engineering that applies scientific and engineering principles to identify security vulnerabilities and minimize or contain risks associated with these vulnerabilities. It uses mathematical, physical, and related scientific disciplines, and the principles and methods of engineering design and analysis to specify, predict, and evaluate the vulnerability of the system to security threats. (MIL-HDBK-1785)

System Assurance (SA) : The justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle. (National Defense Industrial Association (NDIA) Guidebook, *Engineering for System Assurance* , Ver. 1.0)

Therefore, Systems Security Engineering (SSE) comprises the security-related Systems Engineering (SE) processes, activities, products, and artifacts for Systems Analysis (SA). Chapter 4 discusses the need to apply Systems Engineering (SE) early in the acquisition lifecycle. Accordingly, Systems Security Engineering (SSE) must also be considered early (and often).

13.7.6.2. Context of Systems Security Engineering (SSE) Within SE

In order to be cost-efficient and technically effective, Systems Security Engineering (SSE) must be integrated into Systems Engineering (SE) as a key sub-discipline. In fact, Section 5.3.5 of the Program Protection Plan (PPP) Outline indicates that the Program Protection Plan (PPP) should

Describe the linkage between system security engineering and the Systems Engineering Plan and answer the question, How will system security design considerations be addressed?

DAG Chapter 4 provides comprehensive guidance for Systems Engineering (SE). In this chapter, Section 13.7.6.3 provides an overview of Systems Security Engineering (SSE) as a key countermeasure and Section 13.14 provides elaboration on how to include Systems Security Engineering (SSE) within Systems Engineering (SE). As a contextual starting point, the evolution of specifications and associated baselines across the acquisition is shown in Table 13.7.6.2.T1 .

Table 13.7.6.2.T1 Evolution of Specifications/Baselines

The	is developed by	and forms the
System Requirements Document (SRD)	the Government	Requirements Baseline
System Specification	the Contractor	Functional Baseline
Lower-level Subsystem Spec	the Contractor	Allocated Baseline
Fully-decomposed Component Spec	the Contractor	Product Baseline

Each of these specifications should baseline the developing system security requirements to be included in system design by applying the methods and tools of good Systems Engineering (SE). For example, as described in Section 13.3.2.1, repeated application of the Criticality Analysis (CA) methodology, reinforced by Systems Engineering (SE) tools such as fault isolation trees and system response analysis, will yield incremental refinements in the determination of what to protect and how to protect it.

Table 13.7.6.2.T2 shows the expected maturity of the baselines across the system lifecycle, according to the Systems Engineering Technical Reviews (SETR) events at which they should be assessed. It is noteworthy that even as early as the Alternative Systems Review (ASR), the preliminary system requirements should be identified. For further details concerning the appropriate system security content of the maturing baselines as they relate to the Systems Engineering (SE) review timeline, see Section 13.10.2 (*Systems Engineering Technical Reviews*).

Table 13.7.6.2.T2. Expected Maturity of Baselines at Each Systems Engineering Technical Reviews (SETR) Event

SETR or Audit	Typical Required Maturity of the Baselines				
	Requirements	Functional	Allocated	Design Release	Product
ASR	Preliminary	--	--	--	--
SRR	Draft	Preliminary	--	--	--
SFR	Approved	Entrance: Draft Exit: Established	Preliminary	Initial Preliminary	--

PDR	Maintained	Approved and Maintained	Entrance: Draft Exit: Established	Preliminary Draft	--
CDR	Maintained	Maintained	Approved and Maintained	Approved	Exit: Initial Baseline Established
FCA	Maintained	Maintained	Maintained	--	Controlled
SVR	Maintained	Maintained	Maintained	--	Controlled
PCA	Maintained	Maintained	Maintained		Finalized; Approved

13.7.6.3. Systems Security Engineering (SSE) Across the Lifecycle

While Systems Security Engineering (SSE) is categorized in this chapter as a countermeasure, it is important to realize that Systems Security Engineering (SSE) is actually an overarching Systems Engineering (SE) sub-discipline, within the context of which a broad array of countermeasures is appropriate. Some of these Systems Security Engineering (SSE)-related countermeasures represent an overlap with other countermeasure categories, such as Software Assurance (see Section 13.7.3) and Supply Chain Risk Management (SCRM) (see Section 13.7.4).

As a countermeasure in its own right, key Systems Security Engineering (SSE) activities are highlighted as follows:

- **Integrate Security into Requirements, Systems Security Engineering (SSE) Processes, and Constructs**
  - Integrate security requirements into the evolving system designs and baselines
  - Use secure design considerations to inform lifecycle trade space decisions
- **Activity Considerations by Phase**
  - Pre-Milestone A: Evaluate mission threads, identify system functions, and analyze notional system architectures to identify mission critical functions
  - Pre-Milestone B: Refine critical function list and identify critical system components and candidate subcomponents (hardware, software, and firmware)
  - Pre-Milestone C: Refine list of critical system components and subcomponents
  - **Note:** Consider rationale for inclusion or exclusion in the list and for priority assignments
- **Identify and implement countermeasures and sub-countermeasures**
  - Assess risks and determine mitigation approaches to minimize process vulnerabilities and design weaknesses
  - Perform cost/benefit trade-offs where necessary

Key Systems Security Engineering (SSE) criteria can be specified for each of the phases leading up to a major program Milestone, and it is important to establish these criteria across the full lifecycle in order to build security into the system. Further details are provided in Section 13.14 .

13.7.7. Security

This section will be updated in the next Defense Acquisition Guidebook (DAG) update.

[1] http://capec.mitre.org/documents/An_Introduction_to_Attack_Patterns_as_a_Software_Assurance_Knowledge_Resource.pdf

13.8. Horizontal Protection

13.8.1. Acquisition Security Database (ASDB)

13.8.2. Horizontal Protection Process

13.9. Foreign Involvement

13.9.1. Programs with Foreign Participation

13.9.2. Defense Exportability Features (DEF)

13.8. Horizontal Protection

Horizontal protection analysis is the process that determines if critical defense technologies, to include Critical Program Information (CPI), associated with more than one Research, Development, and Acquisition (RDA) program, are protected to the same degree by all involved DoD activities (DoDI 5200.39). DoD horizontal protection requires that all those who develop, process or store the same or similar Critical Program Information (CPI) use the same or equally effective:

- Classification standards
- Export Control guidelines
- Foreign disclosure arrangements
- Anti-tamper protections
- Information Assurance standards
- Physical Security Standards

Horizontal protection is necessary to ensure that an investment made by one program to protect Critical Program Information (CPI) is not diminished due to another program exposing the same Critical Program Information (CPI) or a similar technology with great risk. The goal of the adjudication process is an agreement on a common risk mitigation level among affected programs for the same Critical Program Information (CPI) or similar technologies, not a common protection requirement. The adjudication step in the horizontal protection process will only be necessary in rare cases.

13.8.1. Acquisition Security Database (ASDB)

The Acquisition Security Database (ASDB) is designed to support PMs, Research and Technology Protection (RTP), Anti Tamper (AT), Counterintelligence, Operations Security

(OPSEC), and Security personnel supporting Acquisition Programs. It provides automated tools and functionality to enable efficient and cost-effective identification and protection of Critical Technologies (CT) and Critical Program Information (CPI). It offers a federated search capability and facilitates a standardized identification, implementation, and tracking of Critical Program Information (CPI) countermeasures.

The Acquisition Security Database (ASDB) supports program protection and specifically the horizontal protection process by providing a repository for Critical Program Information (CPI) and Countermeasures and offering a collaboration environment for programs, counterintelligence, security Subject Matter Experts (SMEs), and enterprise individuals supporting the program protection planning effort. The use of the Acquisition Security Database (ASDB) by DoD Components was directed by an Acquisition, Technology, and Logistics (AT&L) memo on July 22, 2010 which directed DoDI 5200.39 establishes policy to conduct comparative analysis of defense systems' technologies and align Critical Program Information (CPI) protection activities horizontally throughout the DoD.

The Acquisition Security Database (ASDB) supports the horizontal protection process in three ways. First, the Acquisition Security Database (ASDB) features a federated search that will allow enterprise individuals to search for similar Critical Program Information (CPI) based on name, Military Critical Technology List (MCTL), or technology type. The search results include the names of Program Protection Plans (PPPs) that match the search criteria. The Critical Program Information (CPI) description within the Program Protection Plan (PPP) can be reviewed and the assessment made as to whether or not the two Critical Program Information (CPI) are similar enough to require similar risk mitigation levels. Second, after the same or similar Critical Program Information (CPI) have been identified and determined to require equivalent risk mitigation, the protections lists in the Program Protection Plans (PPPs) can be analyzed for applicability. Third, the Acquisition Security Database (ASDB) provides the collaboration environment for discussions about the comparison of Critical Program Information (CPI), counterintelligence threats, and planned protections.

The program Acquisition Security Database (ASDB) record should be created as soon as Critical Program Information (CPI) is identified and updated periodically, as changes occur, and at each subsequent milestone. Critical Functions/Components are not identified in the Acquisition Security Database (ASDB).

To request access to the Acquisition Security Database (ASDB), please do the following from a SIPRNET Terminal:

1. Navigate to the Acquisition Security Database (ASDB) Public Site: https://asdb.strikenet.navy.smil.mil/default.aspx .
2. Select "Register" (on left menu).
3. Fill out User Information.
4. Click "Submit Information".

Your access request will be placed into the workflow process for approval. Once approved you will receive a SIPR and NIPR email from the Acquisition Security Database (ASDB) Technical

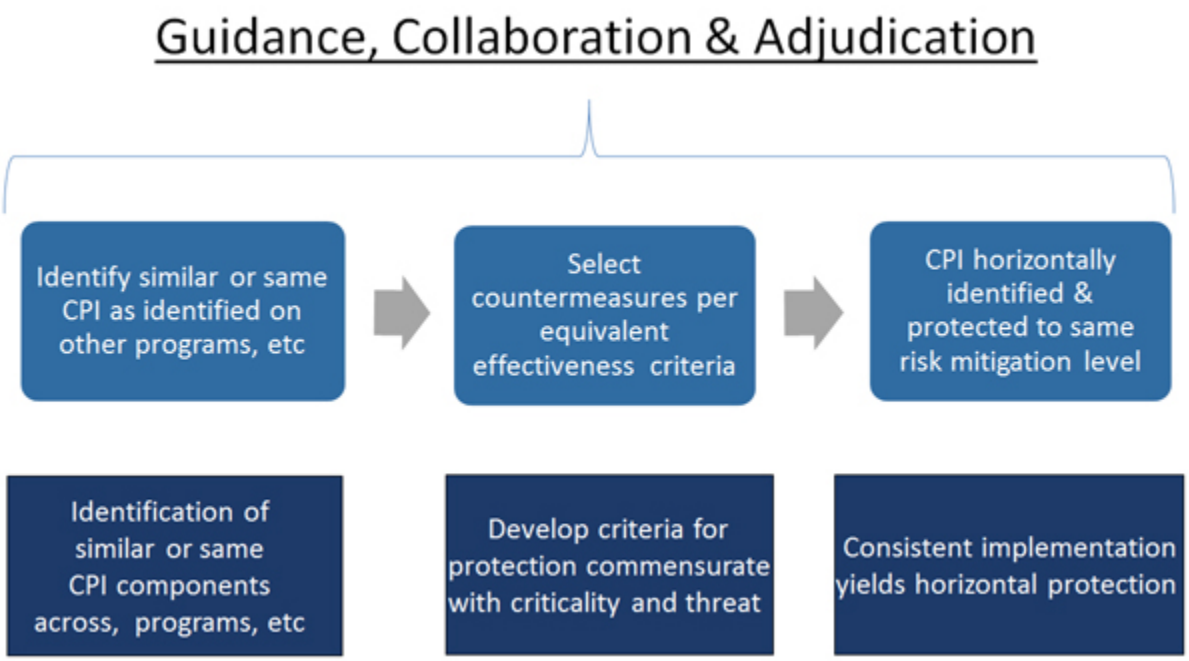
Team.

If you need assistance, please contact the Acquisition Security Database (ASDB) Technical Team at (252) 464-5914 or DSN 451-5914.

13.8.2. Horizontal Protection Process

Programs should be continuously doing horizontal protection as soon as Critical Program Information (CPI) is identified. Below is the high-level horizontal protection process that shows how to protect Critical Program Information (CPI) to the same degree across the Services and programs.

Figure 13.8.2.F1. Horizontal Protection Process



The process for doing horizontal protection during Program Protection Plan (PPP) creation, update, or review is outlined below:

1. Request Acquisition Security Database (ASDB) access at https://asdb.strikenet.navy.smil.mil/default.aspx .
2. Create a record and fill out appropriate fields.
3. Use the search capabilities in the Acquisition Security Database (ASDB) to identify other programs with potentially similar Critical Program Information (CPI). Consider threat and vulnerability differences between programs.
4. Compare planned countermeasure protection against the similar Critical Program Information (CPI) and consider threat and vulnerability differences between programs.
5. If there are perceived discrepancies or concerns, adjudicate the differences at the lowest organizational level.

6. If the discrepancies are not resolved, escalate to an executive decision making organization as determined by Acquisition, Technology, and Logistics (AT&L).
7. Incorporate the results of adjudication into the Program Protection Plan (PPP).

If you have any questions or need assistance, please contact the Acquisition Security Database (ASDB) Technical Team.

Phone: (252) 464-5914 or DSN 451-5914

NIPR E-Mail: W_SPAWAR_CHRL_SSCLANT_ChPt_TEAM_US@navy.mil

SIPR E-Mail: webmaster@strikenet.navy.smil.mil

13.9. Foreign Involvement

13.9.1. Programs with Foreign Participation

The Department of Defense, by law, is to consider cooperative opportunities with the North Atlantic Treaty Organization (NATO), NATO member nations, NATO organizations, non-NATO major allies, and other friendly countries for major defense acquisition programs. In general, DoD policy encourages foreign participation in DoD acquisition programs. Foreign participation ranges for cooperative research and development through production (i.e. foreign industrial participation), acquisition of the system through direct commercial sales (DCS) or foreign military sales (FMS), and other follow-on support.

When it is likely that there will be foreign involvement or access to the resulting system or related information, the Program Manager must plan for this foreign involvement to assist in identifying vulnerabilities to foreign exposure and developing technology security and foreign disclosure guidance to ensure that foreign access is limited to the intended purpose for that involvement while protecting critical program information and mission-critical functions and components and other sensitive information.

Some considerations to forecast potential foreign involvement includes:

- Cooperative research and development with allied or friendly foreign countries
- An allied system may be adopted
- System will replace/upgrade a system that had been previously sold to allies and friends
- System interoperability will be needed for use in future coalition operations

The Program Manager should consult with their international programs organization and technology security and foreign disclosure offices (i.e. National Disclosure Policy, Low Observable/Counter Low Observable (LO/CLO), Defensive Security Systems (DSS), Anti-Tamper (AT), Communication Security (COMSEC), Intelligence, etc.) to obtain assistance in addressing this matter. An integrated product team might be established for this purpose. International considerations to be addressed include the following, many of which should be available from the analysis used in developing the International Cooperation section of the

Technology Development Strategy:

- Summarize any plans for cooperative development with foreign governments or cognizant organizations.
- Summarize plans to increase the opportunity for coalition interoperability as part of the developing DoD program.
- Assess the feasibility from a foreign disclosure and technology security perspective.
- Prepare guidance for negotiating the transfer of classified information and critical technologies involved in international agreements.
- Identify security arrangements for international programs.
- Coordinate with the disclosure authority on development of Delegation of Disclosure Authority Letter (DDL).
- Assist with development of Program Security Instruction (PSI) for a cooperative development program in support of program.
- Support decisions on the extent and timing of foreign involvement in the program,
- Foreign sales, and access to program information by foreign interests.
- Specify the potential or plans for Foreign Military Sales (FMS) and/or Direct Commercial Sales (DCS) and the impact upon program cost due to program protection and exportability features.
- Identify/nominate for consideration as a Defense Exportability Features (DEF) candidate.

13.9.2. Defense Exportability Features (DEF)

Defense Exportability Features (DEF) was established in the fiscal year 2011 National Defense Authorization Act to develop and incorporate technology protection features into a system or subsystem during its research and development phase. By doing this, exportable versions of a system or subsystem could be sold earlier in the Production and Development phase, thereby (1) enabling capability to be available to allies and friendly companies more rapidly and (2) lowering the unit cost of DoD procurements. Prior to the Engineering and Manufacturing Development Phase, programs should investigate the necessity and feasibility (from cost, engineering, and exportability perspectives) of the design and development of differential capability and enhanced protection of exportable versions of the system or subsystem. See Chapter 11.2.1.2. International Considerations within the Acquisition Management Framework for summary of Defense Exportability Features (DEF) nomination and feasibility assessment.

13.10. Managing and Implementing PPPs

13.10.1. Audit's

13.10.2. Systems Engineering Technical Reviews

13.10.2.1. Initial Technical Review (ITR)

13.10.2.2. Alternative Systems Review (ASR)

13.10.2.3. System Requirements review (SRR)

13.10.2.4. System Functional Review (SFR)

13.10.2.5. Preliminary Design Review (PDR)

13.10.2.6. Critical Design Review (CDR)

13.10.2.7. Test Readiness Review (TRR)

13.10.2.8. System Verification Review (SVR) / Functional Configuration Audit (FCA)

13.10.3. Verification and Validation (V&V)

13.10.4. Sustainment

13.10. Managing and Implementing PPPs

The Program Protection Plan (PPP) is a living document, required at Milestones A, B, C, and the Full-Rate Production decision through a July 18, 2011 Principal Deputy Undersecretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) policy memo. It is a best practice to also update the Program Protection Plan (PPP) prior to export decisions, and in order to report progress at each Systems Engineering Technical Review (SETR) event. The Program Protection Plan (PPP) is the single focal point for all program protection and system security activities on the program. It:

- Can serve as a focal point for capturing critical System Security Engineering (SSE) analysis and assessment results as it gathers and matures.
- Will provide previously-missing coverage of Systems Security Engineering (SSE) activities and associated analyses.
- Should contain either references to where the information can be found or the actual information.

The Program Protection Plan (PPP) is a plan a forward-looking document according to which the execution of protection will be performed. But it is also a report, which gathers the analysis and assessment results for effective program protection and system security as the plan is executed.

In this manner (plan + report), the Program Protection Plan (PPP) serves to provide the Security Assurance Case (reference International Organization for Standardization (ISO) Standard 15026 - Part 2 for a discussion on Assurance Cases), in which the preferred system concept represents the assurance claims, the system design represents the assurance arguments and the test plan results and Requirements Traceability Matrix (RTM) is the assurance evidence that can be traced to the system requirements.

All the pieces of an Assurance Case (Claims/Arguments/Evidence) are represented automatically by the inclusion of system security in System Engineering artifacts, such as the System

Requirements Document (SRD), the system/subsystem specs, the preliminary and detailed design documents, the Requirements Traceability Matrix (RTM), and the Test Plans and Reports. The Assurance Case is thus built by the traceability of all the Systems Security Engineering (SSE) artifacts.

The Program Protection Plan (PPP) should tie all these things together. For example:

- Table 3.3-1 of the Program Protection Plan (PPP) Outline is used to provide traceability of critical components from mission-level documents (JCIDS (Joint Capabilities Integration Development System) Key Performance Parameters, Key System Attributes, etc.) and Critical Technology Elements (CTE) to the system architecture.
- Section 5.3 of the Program Protection Plan (PPP) Outline discusses the requirement to indicate the Request for Proposal (RFP) Contract Line Item Number (CLIN) or Data Item Description (DID) that will be used to ensure that Critical Program Information (CPI) and critical functions/components are protected in the development environment and on the system.
- Section 9.3 of the Program Protection Plan (PPP) Outline directs the implementation of Verification and Validation (V&V) to provide evidence that system security has been achieved, including a link to relevant discussion in the Test and Evaluation (T&E) documents.

The last bullet above indicates a method of checking Program Protection Plan (PPP) implementation (i.e., Verification and Validation (V&V)). Audit's/inspections are also used; namely, to ensure compliance with applicable laws, regulations, and policies. Engineering reviews are used to ensure that system security requirements are identified, traceable, and met throughout the acquisition lifecycle. These methods of checking Program Protection Plan (PPP) implementation are described in the following subparagraphs.

13.10.1. Audit's

Program Protection Surveys and other security audit's and inspections check for compliance with protection requirements. These processes check that statutory and regulatory security activities are being performed. Program Managers (PMs) should check with their Component research and development acquisition protection resources for guidance on performing these audit's.

13.10.2. Systems Engineering Technical Reviews

Section 9.2 of the Program Protection Plan (PPP) Outline requires that the Program Protection Plan (PPP) answer these questions:

- How will system security requirements be addressed in Systems Engineering Technical Reviews (SETR), functional/physical configuration audit's, etc.?
- What Program Protection entry/exit criteria will be used for these reviews?

The Systems Engineering Technical Reviews (SETR) process provides a key Systems

Engineering health and risk assessment tool that is discussed in detail in Chapter 4 .

The following subparagraphs provide key Systems Security Engineering (SSE) and Supply Chain Risk Management (SCRM) criteria, recommended as Systems Engineering Technical Reviews (SETR) entry/exit criteria, in order to assess and ensure that an appropriate level and discipline of program protection activities and analyses are conducted across the full system context.

13.10.2.1. Initial Technical Review (ITR)

System security objectives and criteria are in the process of being defined and will be included in the next update.

13.10.2.2. Alternative Systems Review (ASR)

Relevant objectives include:

- System security and Supply Chain Risk Management (SCRM) are addressed as part of the alternative systems analysis and the development of the preferred system concept.
- The preferred system concept is based on an initial criticality analysis, using current threat data, informed by supply chain risk identification.
- Potential countermeasures for candidate Critical Program Information (CPI) and critical functions are identified.
- Plans are defined to protect critical functions/components, processes, tools, and data.
- The Statement of Work (SOW) for the Technology Development (TD) phase Request for Proposal (RFP) includes appropriate tasks for Systems Security Engineering (SSE) and Supply Chain Risk Management (SCRM).

Recommended Criteria:

- Where system security and Supply Chain Risk Management (SCRM) addressed as part of the alternative systems analysis and the development of the preferred system concept?
    - Was an initial criticality analysis performed and documented for the Program Protection Plan (PPP)?
    - Did it use relevant, current threat data and potential vulnerability information?
    - Was the preferred system concept critical function/component alternatives evaluated for supply chain risks through the threat assessment center and used to add constraints to the system requirements?
    - Are the preferred concept engineering analysis and Request for Proposal (RFP) requirements being informed by supply chain risk considerations such as limited potential suppliers and defensive design?
    - Were criticality analysis results used to determine and evaluate candidate Critical Program Information (CPI) and critical functions, with rationale?
    - Have candidate countermeasures and possible sub-countermeasures been identified, with an emphasis on logic bearing components and supply chain risks?
    - Did the analysis include the full system context, including the multiple systems

that support end-to-end mission threads?

- Was Systems Security Engineering (SSE) an integral part of the Milestone A phase systems engineering analysis?
    - Did all of the Systems Security Engineering (SSE) and Supply Chain Risk Management (SCRM) considerations and analyses inform the identification of requirements in the preliminary system requirements document (SRD)?
    - Have potential subsystem and component alternatives for critical functions been evaluated for potential suppliers, software assurance, and system assurance risks?
    - Has the assessment of security risks resulted in system security requirements in the System Requirements Document (SRD)?
    - Have residual Systems Security Engineering (SSE) based program protection risks and supply chain risks been identified for mitigation?
- Are plans in place to protect critical components, processes, tools, and data?
    - Do they promote awareness and provide personnel training on supply chain risks?
    - Are plans to define and protect critical processes, including the identity of users and system uses, included?
    - What tools are being used, how are they being protected (physically and operationally), and how are tools and data managed (including hardware development tools, software development tools, developer collaboration tools, and configuration management tools)?
- Have appropriate tasks been included in the Statement of Work (SOW) for the Technology Development (TD) phase Request for Proposal (RFP)?
    - Are specific responsibilities for Supply Chain Risk Management (SCRM) and for updated criticality analyses to assess critical functions and refine the identification of critical components included?
    - Is competitive prototyping and design included, as appropriate, for candidate Critical Program Information (CPI) and critical functions?
    - Are tasks to develop associated protection countermeasures included, based on the previously identified potential protection countermeasures and the system security requirements in the System Requirements Document (SRD)?
    - Are the use of software assurance databases and techniques (e.g., Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), Common Weakness Enumeration (CWE), static analysis, and penetration testing) included?

13.10.2.3. System Requirements review (SRR)

Relevant objectives include:

- System security (including criticality analysis and software assurance) and Supply Chain Risk Management (SCRM) concerns are considered in the development of the system performance requirements and non-tailorable design requirements across the full system context (e.g., including SoS).
- Initial threat and vulnerability assessments are performed and used in an updated criticality analysis.
- Lists are developed for initial Critical Program Information (CPI), critical functions (and

potential components), selected countermeasures, and potential sub-countermeasures.
- Relevant Supply Chain Risk Management (SCRM) key practices, such as defensive design, are being applied.

Recommended Criteria:

- Where system security and Supply Chain Risk Management (SCRM) concerns considered in the development of the system performance requirements and non-tailorable design requirements, across the full system context (e.g., System of Systems (SoS))?
    - Are the system security and Supply Chain Risk Management (SCRM) requirements mutually understood between the Government and contractor(s)?
    - Are they testable or otherwise verifiable?
    - Will they lead to a final system that is operationally secure and consistent with cost and schedule?
- Is Systems Security Engineering (SSE) an integral part of the Technology Development (TD) phase systems engineering analysis?
    - Have contractor(s) performed and summarized their initial criticality analysis (which updates the Government provided initial criticality analysis, if available)?
    - In the absence of contractors, has the Government performed an updated criticality analysis?
    - Does it include rationale for the selection of Critical Program Information (CPI) and critical functions and potential components?
    - Have lists of initial Critical Program Information (CPI), critical functions (and potential components), selected countermeasures, and potential sub-countermeasures been developed?
    - Have initial threat and vulnerability assessments have been performed, tied to the contractors initial criticality analysis summary?
- Is/are the contractor(s) effectively fulfilling Technology Development (TD) phase Statement of Work (SOW) tasks for Systems Security Engineering (SSE) and Supply Chain Risk Management (SCRM)?
    - Are contractor-refined system security requirements derived from the countermeasures, sub-countermeasures, and defensive design or runtime features selected (e.g., design diversity and least privilege)?
    - Is there a draft allocation of sub-countermeasures and defensive requirements to preliminary design (architecture)? Does that design (architecture) extend to the full system context (e.g., System of Systems (SoS))?
    - Does the Systems Engineering Management Plan (SEMP) describe Systems Security Engineering (SSE) processes, with process updates derived from the countermeasures, sub-countermeasures, and controls selected?
    - Is there a draft allocation of process sub-countermeasures to the acquisition time line and to management and Systems Engineering (SE) sub-processes?
    - Does the contractors review package include planning to address the government provided residual security risk assessment (divided into acquisition, operational, and sustainment sections)?
    - Are tasks, funding, and schedule allocated in order to implement the Systems

Security Engineering (SSE) and Supply Chain Risk Management (SCRM) requirements for the system and for management and SE sub-processes?
    - Are appropriate software assurance databases and techniques (e.g., Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), Common Weakness Enumeration (CWE), static analysis, and penetration testing) being planned and used?
- Are relevant Supply Chain Risk Management (SCRM) key practices being applied?
    - What development and configuration management tools are being used, how are they being protected (physically and operationally), and how are tools and data managed (including hardware, software, and data configuration management)?
    - Is defensive design being used across the full system context to anticipate potential ways that an element, system, or process could fail or be misused, so that the architecture and requirements specification of the element, system, or process can minimize failures and misuse?
    - How are critical elements and processes being protected?
    - How are trustworthy elements being selected?
    - How are supply chain assurance concerns being incorporated into the requirements?
    - Does the contract language cover Supply Chain Risk Management (SCRM) considerations (e.g., the right to subcontract, etc.)?

13.10.2.4. System Functional Review (SFR)

Relevant objectives include:

- System security and Supply Chain Risk Management (SCRM) concerns are considered in establishing the functional baseline.
- An updated criticality analysis is performed, together with updated threat and vulnerability assessments, as required.
- Lists are updated for candidate Critical Program Information (CPI), critical functions and components, selected countermeasures, and potential sub-countermeasures.
- Relevant Supply Chain Risk Management (SCRM) key practices are being applied.

Recommended Criteria:

- Has an updated criticality analysis summary been generated, including rationale for Critical Program Information (CPI) and critical-function selection?
    - Does it derive and allocate critical functions?
    - Does it update the system design based on critical functions and design trade-offs?
    - Are updated threat and vulnerability analyses included, as required?
- Have derived Systems Security Engineering (SSE)/protection functional requirements been flowed into updated subsystem and preliminary component specifications?
    - Has a draft allocation of sub-countermeasures and defensive functions to preliminary functional and physical design been performed? Does this allocation extend across the full system context?

- Have candidate Critical Program Information (CPI) and critical-function countermeasure and sub-countermeasure functional requirements in the system spec been traced to lower level specifications?
    - Have Critical Program Information (CPI) and critical-function countermeasure and sub-countermeasure trade-off analyses been conducted with respect to cost, benefit, and risk?
    - Have Critical Program Information (CPI) and critical-function residual vulnerability risks been assessed?
- Are the cost and schedule of lower-level Systems Security Engineering (SSE) tasks identified and included in the lower level cost and schedule plans?
    - Are detailed Systems Security Engineering (SSE) activities in agreement with the Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)?
    - Do planning packages include required resources to complete Systems Security Engineering (SSE) tasks?
- Are relevant Supply Chain Risk Management (SCRM) key practices being applied?
    - What development tools are being used by suppliers, how are they being protected (physically and operationally), and how are tools and data managed (including hardware, software, and data configuration management)?
    - Is defensive design being applied to include defensive functions and to maximize resilience? Does this extend across the full system context?
    - How are critical elements and processes being protected?
    - How are trustworthy elements being selecting?
    - How thoroughly are suppliers and their supply chain(s) being evaluated?
    - Are the plans to promote awareness and provide personnel training on supply chain risks being executed?

13.10.2.5. Preliminary Design Review (PDR)

Relevant objectives include:

- System security and Supply Chain Risk Management (SCRM) concerns are considered in establishing the system allocated baseline.
- Preliminary system design, including security, is ready to proceed into detailed design; and, stated security performance requirements can be met within cost, schedule, risk, and other system constraints.
- An updated criticality analysis is performed and an updated list of Critical Program Information (CPI), critical functions and components, selected countermeasures, and sub-countermeasures is produced, with rationale.
- Relevant Supply Chain Risk Management (SCRM) key practices are being applied.

Recommended Criteria:

- Have system security and Supply Chain Risk Management (SCRM) concerns been considered in establishing the system allocated baseline?
    - Has an updated criticality analysis summary been generated, with rationale for Critical Program Information (CPI) and critical component selection?

    - Was an updated threat and vulnerability assessment summary, with respect to the updated criticality analysis summary, included, and were supply chain risks included? Does this extend across the full system context (e.g., System of Systems (SoS))?
    - Does the updated Critical Program Information (CPI)/critical component list include countermeasures and sub-countermeasures?
    - Are inherited Critical Program Information (CPI) and horizontal protection adequately assessed, are they being addressed consistently at system and subsystem levels, and are they documented in the updated Acquisition Strategy (AS), Test and Evaluation Management Plan (TEMP) , and Program Protection Plan (PPP)?
    - Have Systems Security Engineering (SSE) and Supply Chain Risk Management (SCRM) processes been updated, based on the selected countermeasures, sub-countermeasures, and controls?
- Does the preliminary system design appropriately include and address security, and is it ready to proceed into detailed design?
    - Where System Requirements Document (SRD) security requirements trades based on the Program Managers (PM) assessment of cost, schedule, performance, and supply chain risks?
    - Were the security requirements specifications, updated for subsystems and components, derived from the countermeasures, sub-countermeasures and defensive design or runtime features selected (e.g., defense in depth)?
    - Was an updated residual security risk assessment performed for the summary-level critical functions and Critical Program Information (CPI), covering acquisition, operations, and sustainment activities (for both system and processes, after sub-countermeasures are applied), including supply chain considerations? Does this extend across the full system context (e.g., System of Systems (SoS))?
    - Has an Anti-Tamper (AT) plan been generated (if it is a contract deliverable)?
    - Are appropriate software assurance databases and techniques (e.g., Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), Common Weakness Enumeration (CWE), static analysis, and penetration testing) used to assess vulnerabilities and exposures to attack, common destructive attack patterns, and weaknesses in the software architecture and design?
    - Have Critical Program Information (CPI) and critical components and sub-components that were categorized as Critical Technology Elements (CTE) been demonstrated at a Technology Readiness Level (TRL) 6 or better?
- Was an allocation of sub-countermeasures and defensive functions to the design/architecture below the counterintelligence (CI) level performed?
    - Was the critical functionality of each Hardware Configuration Item (HWCI) and Computer Software Configuration Item (CSCI) allocated to lower level components?
    - Were Systems Security Engineering (SSE) fault isolation tree and system response analysis techniques used to define appropriate sub-countermeasures?
    - Does the allocated design effectively implement appropriate sub-countermeasures?

    - Where system designs that could expose critical functionality to vulnerability assessed, and were architecture trade-offs evaluated, in order to formulate the allocated baseline?
    - Were external and subsystem interface requirements vulnerabilities assessed and used as input to the sub-countermeasure selection?
    - Do planned sub-countermeasures for design and implementation include software assurance (e.g., fail-safe defaults, defense in depth, purging of temporary data, avoidance of unsafe coding constructs, secure languages and libraries, and static and dynamic code analysis)?
- Are relevant Supply Chain Risk Management (SCRM) key practices being applied?
    - What development tools are being used by suppliers, how are they being protected (physically and operationally), and how are tools and data managed (including hardware, software, and data configuration management)?
    - How are critical elements and processes being protected?
    - How are supplier roles constrained and access limited?
    - How thoroughly are suppliers and their supply chain(s) being evaluated?
    - Are the plans to promote awareness and provide personnel training on supply chain risks being executed?

13.10.2.6. Critical Design Review (CDR)

Relevant objectives include:

- System security and Supply Chain Risk Management (SCRM) concerns are considered in establishing the system product baseline.
- Detailed system design, including security, is ready to proceed into fabrication/development; and, stated security performance requirements can be met within cost, schedule, risk, and other system constraints.
- An updated criticality analysis is performed and updated lists of Critical Program Information (CPI), critical components and sub-components, selected countermeasures, and specific sub-countermeasures are produced, with rationale. This extend across the multiple systems that support the end-to-end mission threads.
- Relevant Supply Chain Risk Management (SCRM) key practices are being applied.

Recommended Criteria:

- Is Systems Security Engineering (SSE) an integral part of the Engineering and Manufacturing Development (EMD) phase systems engineering analysis?
    - Has the contractor updated and summarized their criticality analysis, with rationale, to include a final list of updated Critical Program Information (CPI) and critical components and sub-components, together with associated countermeasures and explicit sub-countermeasures?
    - Were inherited Critical Program Information (CPI) and horizontal protection adequately assessed in the updated criticality analysis?
    - Were adequately robust Systems Security Engineering (SSE) tools used in establishing the product baseline; e.g., the use of an updated system response

matrix and Systems Security Engineering (SSE) fault isolation tree techniques?
    - Were appropriate software assurance databases and techniques (e.g., Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), Common Weakness Enumeration (CWE), static analysis, and penetration testing) used to reassess vulnerabilities and exposures and to reexamine weaknesses in the software architecture, design, and code?
    - Do sub-countermeasures for implementation and testing include software assurance (e.g., purging of temporary data, avoidance of unsafe coding constructs, secure languages and libraries, static and dynamic code analysis, fault injection, and patch management)?
    - Has a residual vulnerability risk assessment been performed to assess, mitigate and re-assess weaknesses in the detailed design, including an assessment of security in the operational environment?
- Does the detailed system design include and appropriately address security and Supply Chain Risk Management (SCRM) considerations, and is it ready to proceed into fabrication/development?
    - Have all Systems Security Engineering (SSE) requirements been flowed down and mapped to the detailed system design and the lifecycle processes?
    - Has an allocation of specific sub-countermeasures to sub-components and lower-level items in counterintelligence (CI) specifications and Statement of Work requirements been performed?
    - Have appropriate Systems Security Engineering (SSE) countermeasures and sub-countermeasures been allocated to the design with validation criteria established (e.g., engineering-in-depth for separation and layering of critical elements, addition of defensive function layers, and handling of authentication methods)?
    - Does the detailed design incorporate good Systems Security Engineering (SSE) practices, such as minimizing the attack surface, the number of critical components, and/or the number of potential weaknesses? Do these Systems Security Engineering (SSE) practices extend across the full system context (e.g., System of Systems (SoS))?
    - Are quantifiable measures being used to assess the detailed design for security and for application of countermeasures (corrective actions) to address identified deficiencies?
    - Has manufacturability been assessed, including the availability and identification of accredited suppliers for secure fabrication of Application-specific integrated circuits (ASICs), Field-programmable gate array (FPGAs), and other programmable devices?
    - Has validation and verification of system security and Supply Chain Risk Management (SCRM) requirements been finalized and reflected in the Test and Evaluation Management Plan (TEMP), preliminary test plans, Systems Engineering Plan (SEP) , and other operational Concept of Operations (CONOPS) documents?
- Are relevant Supply Chain Risk Management (SCRM) key practices being applied?
    - What development tools are being used by suppliers, how are they being protected (physically and operationally), and how are tools and data managed (including hardware, software, and data configuration management)?

- Was diversification of standard interfaces and defensive design used for architecting the system?
- How will critical elements and processes be protected throughout the lifecycle, including disposal of items?
- How will trustworthy elements continue to be selected throughout the lifecycle?
- How are supplier roles constrained and access limited?
- How thoroughly are suppliers, their supply chain(s), and their delivery processes being evaluated?
- How are Government supply chain delivery mechanisms being protected?
- Are the plans to promote awareness and provide personnel training on supply chain risks being executed?

13.10.2.7. Test Readiness Review (TRR)

Relevant objectives include:

- System security and Supply Chain Risk Management (SCRM) concerns are considered in establishing readiness of the system to begin formal acceptance testing, and this extends across the full system context (e.g., System of Systems (SoS)).
- The system test plans and objectives, including scope, procedures, and test facilities, are adequate for appropriately verifying all system security requirements.

Recommended Criteria:

- Is there a documented, comprehensive mapping of all security requirements to their test/verification methods, to include bi-directional traceability?
    - Does it include all system security requirements, clearly defined from threat analysis/modeling, and vulnerabilities identified in residual vulnerability risk assessments? Does this extend to the multiple systems that support the end-to-end mission threads?
    - Are all countermeasures for all identified vulnerabilities implemented in the detailed design included and mapped?
    - Do system verification and validation plans (including flow-down from the Test and Evaluation Management Plan (TEMP) to test plans and procedures) adequately ensure that coding and fabrication of designed components provide the required system security?
    - Is it possible to pull the thread from the initial criticality analysis identifying comprehensive security requirements across the lifecycle to the verification/validation efforts (comprising, for example, a set of organized pointers into the Program Protection Plan (PPP), System Requirements Document (SRD), flow-down requirements specifications, design documents, requirements traceability matrix, and the Test and Evaluation Management Plan (TEMP), test plans, and test procedures)?
- Are system test plans and objectives, including scope, procedures, and test facilities, adequate for appropriately verifying all system security and Supply Chain Risk Management (SCRM) requirements, across the full system context (e.g., System of

Systems (SoS))?

- Are security threat and attack scenarios included in testing?
- Does system testing include penetration testing, testing for confidentiality of users and uses, and configuration of elements to limit access and exposure?
- Are appropriate security test facilities and test equipment, schedule, and personnel planned in the Test and Evaluation Management Plan (TEMP) and lower level test plans, Integrated Master Plan (IMP), and Systems Engineering Plan (SEP); and, are they adequate and available?
- Do planned measures for verification testing and operational use include software assurance sub-countermeasures/techniques (e.g., static and dynamic code analysis, fault injection, patch management, white- and black-box testing, penetration testing, sandboxing, and honey pot systems)?
- Have Critical Program Information (CPI) and critical components and sub-components that were categorized as Critical Technology Elements (CTEs) been demonstrated at a Technology Readiness Level (TRL) 7 or better?

13.10.2.8. System Verification Review (SVR) / Functional Configuration Audit (FCA)

Relevant objectives include:

- The production system is compliant with all functional baseline system security requirements and provides full functionality, without exploitable vulnerabilities (or with security and supply chain risks mitigated to an acceptable level). This extends to the full system context (e.g., System of Systems (SoS)).
- An Updated Program Protection Plan (PPP) is being developed for delivery at Full Rate Production (FRP).

Recommended Criteria:

- Is the production system compliant with all functional baseline system security requirements?
  - Is full system functionality provided without exploitable vulnerabilities, or with security and supply chain risks mitigated to an acceptable level? Does this include the multiple systems that support the end-to-end mission threads, as defined in the Concept of Operations (ConOps)?
  - Are system protection requirements adequate against the system specifications flowed from the Initial Capabilities Documents (ICD)?
  - Is there a complete traceability of capabilities to system security requirements to detailed design to protection verification methods to results?
  - Has a review of all analysis, inspection, demonstration, and test reports of compliance to meet security and Supply Chain Risk Management (SCRM) requirements been conducted, and do all items meet verification needs?
  - Have all planned Systems Security Engineering (SSE) activities/products been implemented, including software assurance, system assurance, and Supply Chain Risk Management (SCRM); and, have the results been documented?
  - Has a residual vulnerability risk assessment been performed to assess, mitigate,

and re-assess weaknesses in the system, including an assessment of security in the operational environment?
- Are plans in place to update the residual risk assessment periodically during sustainment?

13.10.3. Verification and Validation (V&V)

Some program protection activities have requirements testing, verification, and validation built into their processes (e.g. Anti-Tamper, Information Assurance). Further guidance on more general integration of Program Protection into Verification and Validation (V&V) activities will be provided in the next Defense Acquisition Guidebook (DAG) update.

13.10.4. Sustainment

While the primary emphasis of Program Protection is on the design and acquisition phases of a system lifecycle, some sustainment considerations must be addressed if the protection profile is to survive system delivery. Repair depots, for example, should be aware of Critical Program Information (CPI) and mission-critical functions and components on systems they are maintaining so as to appropriately protect these items from compromise. Further guidance on sustainment considerations for Program Protection will be provided in the next Defense Acquisition Guidebook (DAG) update.

13.11. Compromises

13.12. Costs

13.12.1. Security Costs

13.12.2. Acquisition and Systems Engineering Protection Costs

13.13. Contracting

13.13.1. Request for Proposal (RFP) Guidance for all Phases

13.13.1.1. System Requirements Document

13.13.1.2. Statement of Work

13.13.1.3. Instructions, Conditions and Notice to Offeror's (Section L)

13.13.1.4. Evaluation Factors for Award (Section M)

13.11. Compromises

Incidents of loss, compromise, or theft of proprietary information or trade secrets involving Critical Program Information (CPI), are immediately reported in accordance with Section 1831

et seq. of Title 18 of the United States Code , DoD Instruction 5240.04 , and DoD Directive 5200.01 . Such incidents are immediately reported to the Defense Security Service (DSS), the Federal Bureau of Investigation (FBI), the applicable DoD Component counterintelligence (CI) and law enforcement organizations. If the theft of trade secrets or proprietary information might reasonably be expected to affect DoD contracting, Defense Security Service (DSS) should notify the local office of the Federal Bureau of Investigation (FBI).

DSS presently has responsibility for protecting Critical Program Information (CPI) that is classified. However, the contract may specifically assign Defense Security Service (DSS) responsibility to protect Critical Program Information (CPI) that is controlled unclassified information. Consequently, Defense Security Service (DSS) would receive reporting on unclassified Critical Program Information (CPI) incidents if it had specific protection responsibility or the incident could involve foreign intelligence activity or violate the International Traffic in Arms Regulations (ITAR) or Export Administration Regulations (EAR) .

13.12. Costs

13.12.1. Security Costs

The cost of implementing the selected countermeasures that exceed the normal National Industrial Security Program Operating Manual (NISPOM) costs are recorded in this section of the Program Protection Plan (PPP).

13.12.2. Acquisition and Systems Engineering Protection Costs

A cost benefit risk trade-off is used to decide upon which countermeasures to implement for each of the Critical Program Information (CPI) and Critical Function (CF). Based upon the criticality analysis results and Critical Program Information (CPI) identification (Section 13.3), the threats (Section 13.4), the vulnerability assessment (Section 13.5), the risk analysis, and the list of potential countermeasures (Section 13.6) the program is now ready to prepare the cost-benefit verses risk trade-off. For each Level I and selected level II Critical Function (CF) components and each Critical Program Information (CPI) along with the associated risk analysis a cost and schedule implementation estimate is prepared for each potential countermeasure. Also estimated is the residual (remaining) risk to the Critical Function (CF) or Critical Program Information (CPI) after the countermeasure has been implemented.

Based upon this analysis the program manager can select the countermeasure or combination of countermeasures that best fit the needs of the program. It may be to implement the optimum countermeasure(s) do not fit within the programs constraints and other countermeasures can reduce the risk to an acceptable level. In some cases the program may choose to accept the risk and not implement any countermeasures. The emphasis of this analysis is to allow the program manager to perform an informed countermeasure trade-off with an awareness of the vulnerabilities and risks to the system. A summary of the trade-off analysis along with the rationale for the decision needs to be documented in this section of the Program Protection Plan (PPP).

13.13. Contracting

13.13.1. Request for Proposal (RFP) Guidance for all Phases

Comprehensive program protection needs to be addressed during the Request for Proposal (RFP) development and the system design to ensure that security is designed into the system. Program protection includes features that are included in the system design as wells as elements to be included in the processes used to develop the system. As a result program protection needs to be reflected in the system requirements document (SRD) and the statement of work (SOW) portions of the Request for Proposal (RFP) package. It may also be included in the instructions, conditions and notices to offeror's (section L) and the evaluations factors for award (section M) of the Request for Proposal (RFP).

The program protection analysis needs to be performed iteratively prior to Milestone A and prior to each of the planned Systems Engineering Technical Reviews (SETRs) to ensure that the security features are considered and traded-off in conjunction with the other "ilities" and system performance. The program protection analysis begins with the identification of the Critical Program Information (CPI) and Mission Critical Functions and components (described in Section 13.3) followed by the identification of vulnerabilities (described in Section 13.5), a risk assessment and the identification of potential countermeasures (described in Section 13.7).

13.13.1.1. System Requirements Document

The system requirements document should include security requirements based upon the initial countermeasures identified at Milestone A and good security practices. For example, if a particular Critical Program Information (CPI) component requires anti-tamper protection it may have a requirement to have seals, encryption, environmental and logging requirements for the component (see Section 13.7.1). An Information Assurance countermeasure example may be a requirement to include one of the controls specified in the DoD Information Assurance Certification and Accreditation Process (DIACAP) in the component (see Section 13.7.2).

Examples of software assurance countermeasures include requirements for exception handling and degraded mode recovery. There may also be requirements for specific secure coding practices for critical function components such as input validation, default deny, address execution prevention and least privilege (see Section 13.7.3). A supply chain countermeasure example for critical function components may be a requirement for redundancy diversity or checksum validation during startup (see Section 13.7.4)

13.13.1.2. Statement of Work

During the Request for Proposal (RFP) development not all of the system security requirements and design features have been determined. As a result it is necessary to transfer a major part of the program protection analysis, specification, and countermeasure implementation to the contractor to protect the system from loss of advanced technology, malicious insertion, tampering and supply chain risks. The following responsibilities should be considered for

inclusion in the Statement of Work:

- The contractor shall perform or update a criticality analysis, vulnerability assessment, risk assessment, and countermeasure selection and implementation, with assumptions, rationale, and results, before each of the Systems Engineering Technical Reviews (SETRs) defined for the program.
- For each level I and level II component (in accordance with Table 13.3.2.1.T1), the contractor shall identify the associated logic-bearing hardware, software and firmware that implements critical functions or introduces vulnerability to the associated components (designated as "critical components).
- The contractor shall demonstrate that the contractor has visibility into its supply chain for critical components, understands the risks to that supply chain, and has implemented or plans to implement risk mitigations to counter those risks.

- The contractor shall plan for and implement countermeasures which mitigate foreign intelligence, technology exploitation, supply chain and battlefield threats and system vulnerabilities that result in the catastrophic (Level I) and critical (Level II) protection failures, including:
    1. The application of supply chain risk management best practices, applied as appropriate to the development of the system. Supply chain risk management key practices may be found in the National Institute of Standards and Technology (NIST) Interagency Report 7622, *Piloting Supply Chain Risk Management for Federal Information Systems* , and the National Defense Industrial Association Guidebook, *Engineering for System Assurance* , both publicly available.
    2. The enumeration of *potential* suppliers of critical components, as they are identified, including cost, schedule and performance information relevant for choice among alternates and planned selection for the purpose of engaging with the government to develop mutually-agreeable risk management plans for the suppliers to be solicited.
    3. The processes to control access by foreign nationals to program information, including, but not limited to, system design information, DoD-unique technology, and software or hardware used to integrate commercial technology.
    4. The processes and practices employed to ensure that genuine hardware, software and logic elements will be employed in the solution and that processes and requirements for genuine components are levied upon subcontractors.
    5. The process used to protect unclassified DoD information in the development environment.

- The preceding clauses shall be included in the solicitations and subcontracts for all suppliers, suitably modified to identify the parties.
- The contractor shall develop a set of secure design and coding practices to be followed for implementation of Level I and II critical components, drawing upon the top 10 secure coding practices (https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices) and the Common Weakness Enumeration (CWE)/SysAdmin, Audit, Network, Security (SANS) top 25 most dangerous software errors (

http://cwe.mitre.org/top25/index.html).

- The contractor shall develop a Program Protection Implementation Plan (PPIP) that addresses the following sections of the Program Protection Plan (PPP) outline and example:
    - Section 2 Identifying what to protect
    - Section 4 Vulnerabilities
    - Section 5 Countermeasures
    - Section 7 Program Protection Risks
    - Section 8 Managing and Implementing Program Protection Plan (PPP)
    - Section 9 --Process for Management and Implementation of Program Protection Plan (PPP)
    - Section 10 Process for Monitoring and Reporting Compromises
    - Appendix C: Criticality Analysis

13.13.1.3. Instructions, Conditions and Notice to Offeror's (Section L)

For many Request for Proposal (RFP) packages system security engineering may not have any explicit clauses in this section. If it is determined to identify specific program protection content for the proposal the following items should be considered:

- The offeror, as part of its technical proposal, shall describe the use of its systems security engineering process in specifying and designing a system that is protected from loss of advanced technology, malicious insertion, tampering and supply chain risks.
- The offer shall describe the offeror's Critical Program Information (CPI) identification, mission criticality analysis, vulnerability assessment, risk evaluation and countermeasure implementation in arriving at its system specification and design.
- The offeror shall describe the offeror's secure design and coding practices.

13.13.1.4. Evaluation Factors for Award (Section M)

For most Request for Proposal (RFP) packages systems security engineering may not rise to the level of an evaluation factor. If it does programs should consider following as evaluation criteria:

- The extent to which the offeror employs a disciplined, structured systems security engineering (SSE) process, including Critical Program Information (CPI) identification, criticality analysis, vulnerability assessment, risk analysis and countermeasure implementation in arriving at its system specification and design.

13.14. Detailed System Security Engineering

13.14.1. Systems Security Engineering (SSE) Organization

13.14.2. Systems Security Engineering (SSE) Process

13.14.2.1. Military Handbook 1785

13.14.2.2. Systems Security Engineering (SSE) Activities by Phase

13.14.2.2.1. Materiel Solution Analysis (MSA) Phase

13.14.2.2.2. Technology Development (TD) Phase

13.14.2.2.3. Engineering and Manufacturing Development (EMD) Phase

13.14.3. Security Engineering for International Programs

13.15. Program Protection Plan (PPP) Review/Approval

13.15.1. Review Process

13.15.1.1. Program-Level View of Program Protection Plan (PPP) Review Process

13.15.2. Reviewing Organizations

13.15.3. Approval Process

13.15.3.1. Coordination

13.15.3.2. Approval Authority

13.16. Program Protection Plan (PPP) Classification Guidance

13.14. Detailed System Security Engineering

An overview of System Security Engineering (SSE) is provided in Section 13.7.6 , including:

- A clear definition of Systems Security Engineering (SSE), taken from MIL-HDBK-1785 (Section 13.7.6.1)
- A discussion of the context of Systems Security Engineering (SSE) within Systems Engineering (SE) as a key sub-discipline, including how security requirements are mapped to the evolving system design, specifications, associated baselines, and Systems Engineering Technical Reviews (SETR) events (Section 13.7.6.2)
- An overview of the Systems Security Engineering (SSE) activity considerations by lifecycle phase (Section 13.7.6.3)

The subparagraphs below provide further details.

13.14.1. Systems Security Engineering (SSE) Organization

Systems Security Engineering (SSE) is performed by a variety of professionals. These professionals may have specific expertise in one or more areas (e.g., threat assessment, networking, expertise in one or more security technologies, software assurance, and vulnerability

assessment). While serving in the role of the system security engineer, these professionals are responsible for maintaining a comprehensive and holistic system-view while addressing stakeholder security requirements. Systems Security Engineering (SSE) leverages and adapts the principles and practices of Systems Engineering (SE) within the same system life cycle framework that governs Systems Engineering (SE) processes. The system security engineer should have a foundational understanding of systems engineering to include the roles and processes for which the systems engineer is responsible.

The program manager should utilize a Working-level Integrated Product Team (WIPT) to perform comprehensive program protection analysis. It is the responsibility of the program manager to ensure that the Working-level Integrated Product Team (WIPT) is comprised of appropriate representatives (including Systems Engineers (SEs), Systems Security Engineering (SSE) Subject Matter Experts (SMEs), logistics, system user representatives, and supporting counterintelligence, intelligence, foreign disclosure, and security personnel) to ensure a comprehensive analysis of system technology, hardware, software, firmware, and information. This Working-level Integrated Product Team (WIPT) or a sub-group (such as a System Security Engineering Working Group (SSEWG)) should focus on engineering aspects of security. They should define and identify all Systems Security Engineering (SSE) aspects of the system, develop an Systems Security Engineering (SSE) architecture, review the implementation of the architecture, and participate in design validation. This sub-Working Integrated Product Team (WIPT)/ System Security Engineering Working Group (SSEWG) should be formed as early in the acquisition process as possible.

13.14.2. Systems Security Engineering (SSE) Process

Systems Security Engineering (SSE) supports the development of programs and design-to specifications providing life-cycle protection for critical defense resources. Systems Security Engineering (SSE) secures the initial investment by "designing-in" necessary countermeasures and "engineering-out" vulnerabilities, and thus results in saving time and resources over the long term. During the system design phase, Systems Security Engineering (SSE) should identify, evaluate, and eliminate (or contain) known or potential system vulnerabilities, spanning the life cycle.

13.14.2.1. Military Handbook 1785

While very dated, MIL-HDBK-1785 contains still-useful information on pertinent Systems Security Engineering (SSE) activities and procedures and for contracting necessary Systems Security Engineering (SSE) efforts, including the practice of generating a System Security Management Plan (SSMP). The format and contents for the System Security Management Plan (SSMP) are outlined in the appropriate Data Item Descriptions listed in MIL-HDBK-1785. Current guidance calls for including most of the System Security Management Plan (SSMP) material in the Systems Engineering Plan (SEP) and the Program Protection Plan (PPP).

13.14.2.2. Systems Security Engineering (SSE) Activities by Phase

Systems Security Engineering (SSE) is the vehicle for interfacing research and technology

protection into the Systems Engineering (SE) acquisition process, whereby Systems Engineering (SE) activities prevent exploitation of critical functions and components of U.S. defense systems. The benefit of Systems Security Engineering (SSE) is derived after acquisition is complete by mitigation of threats against the system during deployment, operations, and support. Systems Security Engineering (SSE) also addresses threats during the acquisition process (typically through the supply chain) as well as the possible capture of the system by enemies during combat or hostile actions. Note that Systems Security Engineering (SSE) might be required in localized situations where stakeholder security requirements are addressed in the absence of full implementation of Systems Engineering (SE) activities. This can occur at any stage in the system lifecycle. Key Systems Security Engineering (SSE) criteria can be specified for each of the phases leading up to a major program Milestone, and it is important to establish these criteria across the full lifecycle in order to build security into the system.

13.14.2.2.1. Materiel Solution Analysis (MSA) Phase

During the Milestone A phase, most of the Systems Security Engineering (SSE) related activities, criteria, and results can be mapped to content of the Milestone A Program Protection Plan (PPP), as described in the Program Protection Plan (PPP) Outline. Associated Milestone A engineering analyses and Program Protection Plan (PPP) content include the following:

- Include system security in the architectural/design trade-offs and the construction of notional system designs
- Leverage available threat and vulnerability understanding in the engineering analysis to identify candidate mission-critical functions and define security requirements
- Evaluate concept of operations (CONOPS) and notional system architectures to identify mission-critical functions
    - Apply Systems Engineering (SE) tools to Systems Security Engineering (SSE); e.g., protection fault isolation tree methods and system response matrix analysis
- Perform an initial Criticality Analysis (CA) based on mission threads and system functions to identify and prioritize a reasonable and thorough list of mission-critical functions for protection
    - Identify rationale for inclusion or exclusion of system functions in the list
    - Ensure that comprehensive program protection is fulfilled by the identification of critical functions (completeness will comprise critical functions, critical information, and critical technology; indigenous/organic Critical Program Information (CPI) and inherited Critical Program Information (CPI))
- Identify candidate countermeasures and sub-countermeasures to be applied to each critical function, with emphasis on logic bearing components
- Perform trade-offs of design concepts and potential alternative countermeasures to minimize vulnerabilities, weaknesses, and implementation costs
- Examine residual vulnerability rationale for residual risks and plan for threat and vulnerability residual risk assessments after countermeasures have been applied
    - Use cost benefit trade-offs and other rationale

The threat analyses and plans/schedule to counter them, as captured in the PPP, should correlate with and point to the discussion provided in Section 2.3 of the Technology Development

Strategy (TDS) (see the Technology Development Strategy (TDS)-Acquisition Strategy (AS) Outline).

Other key Systems Security Engineering (SSE) activities during the Milestone A phase, not necessarily captured in specific documents, include:

- Ensure that criticality analyses and the development of security requirements extends to multiple systems that support the end-to-end mission threads, including System of Systems (SoS)/Family of Systems (FoS) interdependencies.
- For the trade-off analysis that considers implementation of critical functions via software, include an evaluation of Software Assurance countermeasures that uses:
    - Common Vulnerabilities and Exposures (CVE) To identify vulnerabilities that enable various types of attacks.
    - Common Attack Pattern Enumeration and Classification (CAPEC) To analyze development and operational environments, and potential interfaces, for common destructive attack patterns.
    - Common Weakness Enumeration (CWE) To examine software architectures and notional designs for weaknesses.
- Ensure that the preferred system concept(s) includes preliminary level security requirements derived from risk assessment and mitigation for critical functions. These requirements will typically be reflected in the preliminary System Requirements Document (SRD), if one exists. (See below for other expected System Requirements Document (SRD) content.)
- Ensure that candidate Critical Program Information (CPI) and potential critical components associated with mission-critical functions are mature enough to adequately address security; i.e., those that are potential Critical Technology Elements (CTE) (being matured to a Technology Readiness Level (TRL) of 4 by Milestone A) include validation of expected security in a laboratory environment.
- Consider Systems Security Engineering (SSE) in planning for technology maturation during the Technology Development (TD) phase, including:
    - Mitigation of candidate critical function risks, including countermeasures and candidate sub-countermeasures.
    - Inclusion of candidate critical functions in competitive prototyping plans for critical components.
    - Inclusion of threats in the definition of relevant environment for a Technology Readiness Level (TRL) of 6.
- Incorporate trade study results into the developing system requirements and the Request for Proposal (RFP) Statement of Work for the Technology Development (TD) phase (See Section 13.13.1 for other expected Request for Proposal (RFP) and Statement of Work content).

Other documents generated during the Milestone A phase should also contain Systems Security Engineering (SSE) relevant content. A thorough discussion of the Systems Engineering Plan (SEP) is given in Chapter 4 . Expected Systems Security Engineering (SSE) content in the Systems Engineering Plan (SEP) can be highlighted as follows:

- Description of Systems Security Engineering (SSE) within the overall Systems Engineering (SE) approach, including processes to be used by the Government and contractors.
- Identification of system level security requirements generation as part of the System Requirements process.
- Technical Risk Plan includes a summary of the mission-critical functions with risks, candidate countermeasures and sub-countermeasures, and residual risk (or a reference to the Program Protection Plan (PPP) if included there).
- Each identified Systems Engineering Technical Review (SETR) event includes Systems Security Engineering (SSE) criteria (see Section 13.10.2 for amplification).

The Test and Evaluation Strategy (TES) provides an overall system Verification and Validation (V&V) strategy; and, pertinent details for ensuring system security are further discussed in Section 13.10.3. Expected Systems Security Engineering (SSE) content in the Test and Evaluation Strategy (TES) is highlighted as follows:

- Prototype/risk reduction testing that involves candidate critical functions and components and/or material countermeasures: In Part I (Introduction), Part II (Program Management and schedule), and Part III (Test and Evaluation Strategy (TES)).
- Long-lead post Milestone-B special-item resources, such as test range facilities and tools, for security requirements testing: In Part IV (Resource Summary).

Security requirements are first baselined in the System Requirements Document (SRD); related Systems Security Engineering (SSE) criteria and requirements are flowed down to contractor(s) via a solid Statement of Work and Request for Proposal (RFP) as follows:

- Expected Systems Security Engineering (SSE) content in the Request for Proposal (RFP) for the Technology Development (TD) phase (if available):
    - Request for Proposal (RFP) Section C:
        - Detailed Statement of Work requirements (see below).
        - System Requirements Document (SRD) is included (see below for level of detail expected).
    - Request for Proposal (RFP) Section L: Request a lifecycle description of the Systems Security Engineering (SSE) and Program Protection (PP) processes with how they integrate into the overall Systems Engineering (SE) process.
    - Request for Proposal (RFP) Section M: Evaluate proposed disciplined, structured Systems Security Engineering (SSE) and Program Protection (PP) processes, including Criticality Analyses (CA(s)) to inform the system specification and design, which mitigates threats and vulnerabilities to system/mission effectiveness.
- The Technology Development (TD) phase will involve prototyping efforts and system design trade-off considerations for risk reduction. Ensure that the Statement of Work requires the following level of Systems Security Engineering (SSE) activities from contractor(s) engaged in these activities:
    - Use and maintain current Critical Program Information (CPI) and critical function and component threat assessments (current within 2 years).

    - Update the Criticality Analysis (CA) to identify critical functions and components, with rationale, and to allocate countermeasures and sub-countermeasures to the system design, the allocated baseline, and to follow-on development efforts (e.g., the product baseline).
    - Flow down the Systems Security Engineering (SSE) requirements from the System Requirements Document (SRD) to the System Specification, with verification criteria for risk reduction efforts.
    - Refine the allocation of countermeasures and sub-countermeasures to system critical components (features included in system design) as well as lifecycle phases (processes used to develop the system).
    - Include detailed Systems Security Engineering (SSE) criteria for Systems Engineering Technical Reviews (SETRs), which should be reflected in their System Engineering Management Plan (SEMP).
        - Include coverage of Criticality Analysis (CA) results and supply chain risk information (see Section 13.10.2 for further details).
    - Include security requirements and critical function/component artifacts within their Systems Engineering (SE) and design Contract Data Requirements Lists (CDRLs).
    - Protection of all critical function and Critical Program Information (CPI)-related prototyping and preliminary design work during technology maturation efforts.
    - Demonstrate visibility into the supply chain for hardware and software elements to be used during the technology maturation efforts. Allow Government oversight reviewers to inspect results of Systems Security Engineering (SSE) processes (including countermeasures and Systems Security Engineering (SSE) activities) upon request.
- If an System Requirements Document (SRD) is available, the following Systems Security Engineering (SSE) considerations apply:
    - System Requirements Document (SRD) contains security requirements derived from:
        - Operational performance requirements requiring protection.
        - Security focused mission threads in the Concept of Operations (CONOPS).
        - Threats and vulnerabilities in the Initial Capabilities Documents (ICD) and draft Capability Development Documents (CDD).
    - System security requirements in the verification matrix should be traceable to Joint Capabilities Integration Development System (JCIDS) requirements (Initial Capabilities Documents (ICD), draft Capability Development Documents (CDD)) and regulatory requirements.
    - Includes system level countermeasures and sub-countermeasures requirements.

13.14.2.2.2. Technology Development (TD) Phase

During the Technology Development (TD) phase, most of the Systems Security Engineering (SSE) related activities, criteria, and results can be mapped to content of the Milestone-B Program Protection Plan (PPP), as described in the Program Protection Plan (PPP) Outline. Associated Technology Development (TD) engineering analyses and Program Protection Plan

(PPP) content include the material covered in Section 13.7.6, as well as the following:

- Perform an updated Criticality Analysis (CA), based on the previous Criticality Analysis (CA) results and current (within 2 years) threat and vulnerability data, in order to refine the mission-critical function list and identify critical components and potential subcomponents (hardware, software, and firmware)
    - Performed by the Government and/or contractor(s)
    - Employ Systems Security Engineering (SSE) protection fault isolation tree analysis and examine system response matrix for protection of Preliminary Design Review (PDR) level design
    - Use security scenarios in the operational Concept of Operations (CONOPS)
    - Identify logic bearing elements and failure effects for impact in order to assign criticality failure levels
    - Assign levels of criticality in order to prioritize critical functions and components for further analysis, focusing on Level I (Catastrophic) and Level II (Critical)
    - Identify rationale for inclusion or exclusion of critical functions and components in the list
    - Ensure that comprehensive program protection is fulfilled by the identification of critical components (completeness will comprise critical functions and components, critical information, and critical technology; indigenous/organic Critical Program Information (CPI) and inherited Critical Program Information (CPI))
- Identify program protection required to the level of specific countermeasures and sub-countermeasures, with plans for using mature technology
- Ensure that prototyping efforts and system design trade-off considerations for risk reduction focus on minimizing the attack surface and system security risks, and on employing affordable, risk-based countermeasures
- Use allocation to the Work Breakdown Structure (WBS) in order to refine the list of countermeasures and sub-countermeasures to be applied to each critical function and component
- Examine residual vulnerability rationale for residual risks and plan for threat and vulnerability residual risk assessments after sub-countermeasures have been applied
    - Use cost benefit trade-offs and other rationale
- Include provisions for evaluations of threats and countermeasure effectiveness at each level of design

The threat analyses and plans/schedule to counter them, as captured in the Program Protection Plan (PPP), should correlate with and point to the discussion provided in paragraph 2.3 of the Acquisition Strategy (AS) (see the Technology Development Strategy (TDS)-Acquisition Strategy (AS) Outline).

Other key Systems Security Engineering (SSE) activities during the Technology Development (TD) phase, not necessarily captured in specific documents, include:

- Monitor contractor performance against the Systems Security Engineering (SSE) criteria for the Technology Development (TD) Statement of Work that were enumerated in

Section 13.7.6.1

- Analyze intra- and inter-system dependencies and plan for corresponding mitigation of exploitable vulnerabilities that could compromise mission-critical system components
- Ensure that criticality analyses and the development and implementation of security requirements extends to multiple systems that support the end-to-end mission threads, including System of Systems (SoS)/Family of Systems (FoS) interdependencies
- For software prototyping and design trade analyses, include an updated evaluation of Software Assurance countermeasures that uses Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), and Common Weakness Enumeration (CWE)
- Ensure that Critical Program Information (CPI) and critical components are mature enough to fulfill related security requirements; i.e., those that are Critical Technology Elements (CTEs) (being matured to an assessed Technology Readiness Level (TRL) of 6 by Milestone B) demonstrate all required security in a relevant environment
- Consider Systems Security Engineering (SSE) in planning for the Engineering and Manufacturing Development (EMD) phase, including:
    - Inclusion of updated threats and vulnerabilities in the definition of operational environment for a Technology Readiness Level (TRL) of 7

Other documents generated during the Technology Development (TD) phase should also contain Systems Security Engineering (SSE) relevant content. For example, Systems Security Engineering (SSE) tasks to implement requirements should be included in the Integrated Master Plan (IMP) and Integrated Master Schedule (IMS), including security verification tied to the Test and Evaluation Management Plan (TEMP) .

A thorough discussion of the Systems Engineering Plan (SEP), updated for Milestone B, is given in Chapter 4. Expected Systems Security Engineering (SSE) content in the updated Systems Engineering Plan (SEP) can be highlighted as follows:

- Updated description of Systems Security Engineering (SSE) within the overall Systems Engineering (SE) approach, including processes used by the Government and contractors, as well as Technology Development (TD) phase Systems Security Engineering (SSE) accomplishments and guidance for Engineering and Manufacturing Development (EMD)
- Refinement and allocation of system level security requirements as part of the System Requirements process
- Technical baseline management for system security requirements at the System Requirements Document (SRD) and System Specification level by the Government and lower level specifications by the contractor(s)
- Technical Risk Plan includes a summary of the mission-critical components with risks, countermeasures and candidate sub-countermeasures, and residual risk (or a reference to Program Protection Plan (PPP))
- Comprehensive end-to-end test approach for system security
- Each identified Systems Engineering Technical Reviews (SETR) event includes Systems Security Engineering (SSE) criteria (see Section 13.7.6 for amplification)

The Test and Evaluation Master Plan (TEMP) provides an integrated system plan for

Verification and Validation (V&V); and, pertinent details for ensuring system security are further discussed in Section 13.10.3. It should be noted, however, that the Test and Evaluation (T&E) associated with critical components and their testable sub-countermeasures will likely not be a part of a programs Test and Evaluation Management Plan (TEMP). A large portion of security Test and Evaluation (T&E) will be planned for and conducted by the contractor as part of the contracts Statement of Work. That said, expected Systems Security Engineering (SSE) content in the Test and Evaluation Management Plan (TEMP) is highlighted as follows:

- System security Developmental Testing (DT)
    - Key system security critical technical parameters (CTPs)
    - Verify system level security requirements as documented in the Requirements Verification Traceability Matrix (RVTM)
- System security Operational Testing (OT)
    - Include system security as a Measure of Suitability (MOS)
- Specific system security resources
    - Developmental Testing (DT)/Operational Testing (OT) test articles with sub-countermeasures
    - Test sites, instrumentation, and support equipment

Security requirements and related system functions are baselined in the Government System Requirements Document (SRD) and the contractors System Specification. Related Systems Security Engineering (SSE) criteria and requirements are flowed down to contractors via a solid Statement of Work and Request for Proposal (RFP) as follows:

- Expected Systems Security Engineering (SSE) content in the Request for Proposal (RFP) for the Engineering and Manufacturing Development (EMD) phase:
    - Request for Proposal (RFP) Section C:
        - detailed Statement of Work requirements (see below)
        - final System Requirements Document (SRD) is included (see below for level of detail expected)
    - Request for Proposal (RFP) Section L: Request a lifecycle description of the Systems Security Engineering (SSE) and Program Protection (PP) processes with how they integrate into the overall Systems Engineering (SE) process. Provide specific security scenario(s) for bidders to describe their proposed system response
    - Request for Proposal (RFP) Section M: Evaluate proposed disciplined, structured Systems Security Engineering (SSE) and Program Protection (PP) processes, including Criticality Analysis (CA), in system specification, detailed design, build/code, and test, with emphasis on countermeasure and sub-countermeasure implementation
- The Engineering and Manufacturing Development (EMD) phase Statement of Work should require the following level of Systems Security Engineering (SSE) activities from contractor(s):
    - Update the Criticality Analysis (CA) to refine the identification of critical functions and components, with rationale
    - Work with appropriate agencies (e.g., Defense Intelligence Agency (DIA) and

National Security Agency (NSA)) to maintain Critical Program Information (CPI) and critical function and component threat assessments (current within 2 years)
    - Allocate sub-countermeasures to critical components and subcomponents (i.e., system detailed design and the product baseline) as well as to lifecycle phases for the processes used to develop the system.
    - For Software Assurance evaluations, use:
        - Common Vulnerabilities and Exposures (CVE) To identify vulnerabilities that enable various types of attacks
        - Common Attack Pattern Enumeration and Classification (CAPEC) To analyze environments, code, and interfaces for common destructive attack patterns
        - Common Weakness Enumeration (CWE) To examine software architectures, designs, and source code for weaknesses
    - Flow down the Systems Security Engineering (SSE) requirements from the System Requirements Document (SRD) to the System Specification and to lower-level specifications, with verification criteria for risk reduction efforts
        - Include detailed allocation of sub-countermeasures to lower-level specifications
    - Include detailed Systems Security Engineering (SSE) criteria for Systems Engineering Technical Reviews (SETRs), which should be reflected in the contractor Systems Engineering Management Plan (SEMP)
        - Include coverage of Criticality Analysis (CA) results and supply chain risk information (see Section 13.10.2 for further details)
    - Include security requirements and critical function/component artifacts within contractor Systems Engineering (SE) and design Contract Data Requirements Lists (CDRLs)
    - Demonstrate visibility into the supply chain for critical components and subcomponents. Allow Government oversight reviewers to inspect results of Systems Security Engineering (SSE) processes (countermeasures, sub-countermeasures, and activities) upon request
- Expected Systems Security Engineering (SSE) content in the System Requirements Document (SRD) and/or preliminary System Specification:
    - Specific security requirements to protect Critical Program Information (CPI) and critical functions and components, based on:
        - Operational performance requirements needing protection
        - Threats and vulnerabilities identified via system-specific assessments as well as those contained in the Capability Development Document
        - Use cases (including common-attack countermeasures) that are comprehensive and traceable to the Concept of Operations (CONOPS)
    - Each security requirement in the verification matrix should be traceable from Joint Capabilities Integration Development System (JCIDS) requirements (Initial Capabilities Document (ICD) and Capability Development Document (CDD)) to countermeasures and sub-countermeasures allocated to system requirements, and adjusted for associated cost, risk, and schedule
    - Identification of specific countermeasures and sub-countermeasures requirements. For example, for Software Assurance countermeasures to be applied to a specific

component, the identification of sub-countermeasures, such as:
    - Static code analysis (for development process application)
    - Secure exception handling (for built-in component security)
    - Sandboxing (for operational threat mitigation)
  - All identified Critical Program Information (CPI) and critical functions and components have specified countermeasure and sub-countermeasure requirements documented in the contractors Spec Tree, with justification of accepted risk

The contractors preliminary Subsystem and Component Specifications should:

- Contain derived system protection requirements that are comprehensive, verifiable, and traced from the System Requirements Document (SRD)
- Provide security and protection verification matrices that are traced and properly allocated from the System Requirements Document (SRD) so that system level protection requirements can be validated
- Reflect the trade-off analysis of the Preliminary Design Review (PDR) with respect to Criticality Analysis (CA), mission-critical functions and components, countermeasures and sub-countermeasures, updated vulnerability assessment, and allocation of sub-countermeasures to design and verification

13.14.2.2.3. Engineering and Manufacturing Development (EMD) Phase

During the Engineering and Manufacturing Development (EMD) phase, most of the Systems Security Engineering (SSE) related activities, criteria, and results can be mapped to content of the Milestone-C Program Protection Plan (PPP), as described in the Program Protection Plan (PPP) Outline. Associated Engineering and Manufacturing Development (EMD) engineering analyses and Program Protection Plan (PPP) content include the material covered in Section 13.7.6 , as well as the following:

- Perform an updated Criticality Analysis (CA), based on the previous Criticality Analysis (CA) results and current (within 2 years) threat and vulnerability data, in order to refine the list of critical components and subcomponents (hardware, software, and firmware) for comprehensive protection
  - Verify threat and vulnerability assessments are current against the product baseline and system operational concept
  - Identify rationale for inclusion or exclusion of system components and subcomponents in the list
- Update the threat and residual vulnerability risk assessment(s), consistent with the updated Criticality Analysis (CA) summary and rationale
  - Ensure that threat and vulnerability residual risk assessment after sub-countermeasures are applied have been tracked and mitigated
- Update the software evaluations using Common Weakness Enumeration (CWE), Common Vulnerabilities and Exposures (CVE), and Common Attack Pattern Enumeration and Classification (CAPEC)
- Update the identification of countermeasures and sub-countermeasures to be applied to specific critical functions and components

- Ensure, prior to Milestone C, that all Critical Program Information (CPI) and mission-critical functions and components are identified, together with all associated Systems Security Engineering (SSE) countermeasures and sub-countermeasures applied to them
- Assess Critical Program Information (CPI) and critical function/component countermeasures and sub-countermeasures for production, deployment, operations, and sustainment
    - Ensure countermeasures and sub-countermeasures have been integrated into the product baseline, production planning, and system operational concept
    - Ensure completeness, comprising critical functions/components/subcomponents, critical information, and critical technology, Indigenous/Organic Critical Program Information (CPI) and Inherited Critical Program Information (CPI)
- Apply residual vulnerability risk assessment after sub-countermeasures applied
- Examine verification results (Developmental Test and Evaluation (DT&E) and Operational Test and Evaluation (OT&E)) for security requirements
- Ensure that Systems Security Engineering (SSE) requirements flow down to detailed design elements with verification criteria:
    - Allocate sub-countermeasures to counterintelligence (CI) Specifications (detailed design) and to verification criteria in the Statement of Work
    - Update the Systems Security Engineering (SSE) protection fault isolation tree and system response matrix
    - Ensure flow down of key Systems Security Engineering (SSE) requirements to appropriate Systems Engineering Technical Reviews (SETR) criteria (Critical Design Review (CDR), Test Readiness Review (TRR), and System Verification Review (SVR)) (see Section 13.10.2 for amplification)

Other key Systems Security Engineering (SSE) activities during the Engineering and Manufacturing Development (EMD) phase include:

- Monitor contractor performance against the Systems Security Engineering (SSE) criteria for the Engineering and Manufacturing Development (EMD) Statement of Work that were enumerated in Section 13.13.1.2
- Ensure that criticality analyses and the implementation and testing of security requirements extends to multiple systems that support the end-to-end mission threads, including System of Systems (SoS)/Family of Systems (FoS) interdependencies
- Update the residual risk assessment after sub-countermeasures have been applied, examine residual vulnerabilities for prioritized risks, and apply mitigations and plans that will ensure system security through deployment and operations
- Ensure that system security is properly reflected in the product baseline and in production planning for Low rate initial production (LRIP) and beyond
    - Systems Security Engineering (SSE) requirements flow down to the product baseline, with application of sub-countermeasures to subcomponents verified and validated
    - Ensure that the Lifecycle Sustainment Plan includes periodic assessment of threats, vulnerabilities, and maintenance of countermeasures and sub-countermeasures
    - Ensure that the sustainment strategy contains sustainment related contracts to

ensure secure supply chain acquisitions correlated with paragraph 7.4.3.6 of the AS (see the Technology Development Strategy (TDS)-Acquisition Strategy (AS) Outline)
    - Ensure that all security-related logic bearing subcomponents have viable contractors
- Specify the impact upon program cost due to program protection and exportability features associated with the potential/plans for Foreign Military Sale correlated with paragraph 10.3 of the AS (see the Technology Development Strategy (TDS)-Acquisition Strategy (AS) Outline)
- Ensure that Critical Program Information (CPI) and critical components and subcomponents are matured to fulfill related security requirements; i.e., those that are Critical Technology Elements (CTEs) (being matured to an assessed Technology Readiness Level (TRL) of 7 by Milestone C) demonstrate all required security in an operational environment
    - Verify manufacturability of needed sub-countermeasures
- Ensure verification of system protection functional requirements:
    - Systems Security Engineering (SSE) functional requirements flow down to detailed design elements and are verified against valid criteria and verification methods
    - Countermeasures and sub-countermeasures are analyzed and tested throughout the detailed design and testing/verification phases
    - Physical and operational countermeasures and sub-countermeasures are included in system operational instructions and training documentation
    - Configuration management system is used to track vulnerability risks and mitigation via design changes
    - Performance of sub-countermeasures is verified against attacks
    - Updated test plans and procedures reflect additional verification requirements and stress attack scenarios
- Ensure that Systems Security Engineering (SSE) tasks to implement requirements are updated in the Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

Other relevant Engineering and Manufacturing Development (EMD) documents include the contractors Test Plans. Expected Systems Security Engineering (SSE) content includes:

- Test Plan activities are traced from the System Requirements Document (SRD) to system, subsystem, and lower-level component requirements, to verification testing needs
- Ensure that all Systems Security Engineering (SSE) testing requirements from system, subsystem, component, and subcomponent level documentation are included in the verification matrix according to agreed verification objectives
- Clear pass-fail criteria are identified for all tests as they apply to system security and protection
- Test processes and facilities, test equipment, Modeling and Simulation (M&S), and the software environment are adequately planned to validate protection countermeasure requirements
- Systems Security Engineering (SSE)-specific needs for personnel and schedule are considered and adequately addressed

- Test plans reflect the Test and Evaluation Management Plan (TEMP)
- Testing includes attack stress use cases
- Security and protection validation testing may require outside accreditation authorities, and appropriate schedule is allocated in the Test Plans

13.14.3. Security Engineering for International Programs

System Security Engineering should include an assessment of security criteria that sets limits for international cooperative programs, direct commercial sales, and/or foreign military sales cases. From this assessment, engineering and software alternatives (e.g., dial-down functionality, export variants, anti-tamper provisions, etc.) should be identified that would permit such transactions.

13.15. Program Protection Plan (PPP) Review/Approval

Program Protection Plan (PPPs) must be approved by the Milestone Decision Authority at Milestones A, B, C, and the Full-Rate Production decision (or business system equivalent). A final draft Program Protection Plan (PPP) must be submitted for the Pre-Engineering and Manufacturing Development (EMD) review prior to Milestone B. This guidance summarizes the approval process that Undersecretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) and DoD Chief Information Officer (CIO) jointly developed for programs under their cognizance. Programs under Component, Program Executive Office (PEO), or other oversight should consult their Milestone Decision Authority (MDA) for applicable approval guidance.

The Program Protection Plan (PPP) Review and Approval Process should be initiated approximately 180 days prior to the programs Defense Acquisition Board (DAB) to allow sufficient time for the comprehensive Office of the Secretary of Defense (OSD) review. The review process iterates as the program responds to Office of the Secretary of Defense (OSD) comments and resubmit' s the Program Protection Plan (PPP) for approval. Once all comments are resolved, the Program Protection Plan (PPP) will be coordinated and routed to the Milestone Decision Authority (MDA) for approval.

13.15.1. Review Process

When a Program Protection Plan (PPP) is ready for Office of the Secretary of Defense (OSD) review, the program will send the document to Deputy Assistant Secretary of Defense (Systems Engineering) (DASD (SE)) Major Program Support (MPS) Program Support Team Lead (PSTL) and Action Officer (AO). The Program Protection Plan (PPP) will be reviewed by SMEs across Office of the Secretary of Defense (OSD) to validate that the Program Protection Plan (PPP) sufficiently addresses all aspects of program protection planning and implementation. If comments are generated, consolidated comments from this comprehensive review are returned to the program for adjudication and resubmission for approval.

An important lesson learned is the program should act early to engage the Component Anti-Tamper community and address any concerns, as Anti-Tamper (AT) Plan approval is commonly

a holdup for overall Program Protection Plan (PPP) approval. Additionally, Program Managers (PMs) may delay receiving Program Executive Office (PEO) or Service Acquisition Executive (SAE) signatures on Program Protection Plans (PPPs) prior to initial Office of the Secretary of Defense (OSD) reviews, as many initial reviews generate comments requiring rework that may need to be re-approved at those levels.

13.15.1.1. Program-Level View of Program Protection Plan (PPP) Review Process

The program sends the draft Program Protection Plan (PPP) to its Program Support Team Lead (PSTL) and Action Officer (AO). Approximately three weeks later, the program will receive a comments matrix from the Program Support Team Lead (PSTL)/ Action Officer (AO) with comments the program needs to address. After addressing the comments, the program will submit an updated Program Protection Plan (PPP) with the adjudicated comments matrix to the Program Support Team Lead (PSTL) and Action Officer (AO). Once all comments have been addressed, the Program Protection Plan (PPP) Review Team will coordinate and staff the Program Protection Plan (PPP) for a Milestone Decision Authority (MDA) signature. Once it is signed, the approved Program Protection Plan (PPP) will be sent back to the program.

13.15.2. Reviewing Organizations

- Deputy Assistant Secretary of Defense (Systems Engineering) (DASD (SE)) from a systems engineering perspective
- DoD Anti-Tamper Executive Agent (ATEA) from an Anti-Tamper (AT) perspective
- DoD Chief Information Officer (CIO) from an Information Assurance and supply chain perspective
- Acquisition, Technology, and Logistics (AT&L) Industrial Policy from a supply chain perspective
- Under Secretary of Defense (Intelligence) from security and counterintelligence perspective
- Acquisition, Technology, and Logistics (AT&L) International Negotiations from an international cooperation perspective

13.15.3. Approval Process

13.15.3.1. Coordination

Once all Office of the Secretary of Defense (OSD) comments are adjudicated, the Program Protection Plan (PPP) is then sent out for Principal-level coordination across Office of the Secretary of Defense (OSD). The following organizations submit Principal-level coordination:

- DoD Anti-Tamper Executive Agent (ATEA)
- DoD Chief Information Officer (CIO)
- Acquisition, Technology, and Logistics (AT&L) Industrial Policy
- Acquisition, Technology, and Logistics (AT&L) International Negotiations
- Acquisition, Technology, and Logistics (AT&L) Acquisition Resources & Analysis
- Acquisition, Technology, and Logistics (AT&L) Strategic & Tactical Systems

- Office of the General Counsel

Once coordination is complete, the Program Protection Plan (PPP) is routed to the Milestone Decision Authority (MDA) for signature.

13.15.3.2. Approval Authority

The approval authority for Program Protection Plans (PPPs) is the Milestone Decision Authority (MDA). The Milestone Decision Authority (MDA) for Acquisition Category (ACAT) ID, special interest, and non-delegated Acquisition Category (ACAT) IAM programs, is Undersecretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)). The DoD Chief Information Officer (CIO) is the Milestone Decision Authority (MDA) for all other IAM programs . For Acquisition Category (ACAT) ICs and below, the Program Protection Plan (PPP) does not need to be reviewed at the Office of the Secretary of Defense (OSD) level.

13.16. Program Protection Plan (PPP) Classification Guidance

The Program Protection Plan (PPP) should be classified by content. There is no overarching Security Classification Guide for DoD Program Protection original classification authority for Critical Program Information (CPI), mission-critical functions and components, threats and vulnerabilities to those items, and protections of those items remain the responsibilities of their respective owners. Program Protection Plans (PPPs) are frequently developed with unclassified bodies and classified appendices as necessary.