Carnegie Mellon University
Software Engineering Institute

Security Engineering Risk Analysis (SERA) Threat Archetypes

Christopher Alberts
Carol Woody, PhD

December 2020

CERT Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

DM20-1087

Table of Contents

List of Figures

List of Tables

1 Introduction

In today's operational environments, multiple organizations often are required to work collaboratively in pursuit of a single mission, creating management complexity that is difficult to control effectively. Successful execution of a multi-organizational mission demands management approaches that effectively coordinate task execution and risk management activities among all participating groups. Organizations across government and industry are beginning to implement mission assurance programs in an effort to coordinate mission execution and help ensure mission success.

Department of Defense (DoD) Directive 3020.40, titled *Mission Assurance (MA)*, defines *mission assurance* as "*a* process to protect or ensure the continued function and resilience of capabilities and assets, including personnel, equipment, facilities, networks, information and information systems, infrastructure, and supply chains, critical to the execution of DoD mission-essential functions in any operating environment or condition" [DoD 2018].

This directive requires DoD components to prioritize mission assurance efforts in support of critical DoD strategic missions. While weapon system acquisition falls outside the direct scope of the DoD's mission assurance directive, the directive does outline important actions relevant to system acquisition:

- Risk management must be addressed as early as possible in the acquisition of information technology across the lifecycle.
- Acquisition programs must integrate mission assurance goals and activities with acquisition guidance.

As a result, mission assurance must be considered during the acquisition of DoD software-intensive systems, such as weapon systems. From a cyber perspective, acquisition programs should start managing cybersecurity risk early in the system-acquisition lifecycle. A complicating factor is that most software-intensive systems are networked. While networking offers many operational efficiencies to a system's stakeholders, it also expands a system's cyber-risk profile.

A network of independently managed software-intensive systems, referred to as a *system of systems (SoS)*, provides information and services that are essential for successful mission execution. Cyber attacks with the potential for mission impact can target any system within an SoS environment, creating complex attack vectors that must be considered during cyber-risk analysis.

Cyber attacks are designed to exploit weaknesses and vulnerabilities in a system's software components, which makes software the focal point for early lifecycle cyber-risk analysis. Software must be architected and designed with the knowledge that it must function as intended in an increasingly contested, challenging, and interconnected cyber environment. Therefore, software assurance is essential for achieving mission assurance.

Software assurance is defined as a level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the lifecycle [NIA 2010]. Software assurance is becoming increasingly important to organizations across all sectors because of software's increasing influence in business- and mission-critical systems.

For example, consider how the size of flight software[1] has increased over the years. Between 1960 and 2000, the degree of functionality provided by software to the pilots of military aircraft increased from 8% to 80%. At the same time, the size of software in military aircraft grew from 1,000 lines of code in the F-4A to 1.7 million lines of code in the F-22. This growth trend is expected to continue over time [NASA 2009]. As software exerts more control over complex systems, like military aircraft, the potential risk posed by cybersecurity vulnerabilities will increase in kind.

In 2005, researchers from the CERT Division of Carnegie Mellon University's Software Engineering Institute (SEI) started investigating how to enable mission assurance in SoS environments [Alberts 2005]. A major conclusion of this research was that system-oriented cyber-risk methods would not readily scale to SoS environments. New analysis approaches were needed to complement traditional system-oriented analysis activities.

In 2013, CERT researchers started investigating how to conduct cyber-risk analysis early in the acquisition lifecycle (i.e., during requirements, architecture, and design). The product of this project was the Security Engineering Risk Analysis (SERA) Method, a scenario-based approach for analyzing complex cybersecurity risks in SoS environments.

The SERA Method is designed to integrate system and software engineering with operational security across the lifecycle and supply chain. The information generated from the SERA Method provides several benefits to a program's acquisition and engineering activities, such as enabling analysts to do the following:

- Find gaps in security requirements.
- Identify weaknesses in system and software architectures.
- Identify risks that code analysis tools cannot find.
- Analyze complex cyber attacks that current compliance-based cyber-risk approaches cannot handle.

Over the past five years, we at the SEI used the SERA Method to conduct multiple SoS cyber-risk analyses for DoD and federal system acquisition programs. Based on our pilot experiences, we identified the development of cyber-risk scenarios as the key to a successful assessment. At the same time, we observed that scenario development can be a time-consuming and difficult task since analysts must have sufficient knowledge, skills, and abilities to develop and evaluate cyber-risk scenarios. When analysts do not understand the ways in which software, hardware, and firmware can be compromised, important scenarios can be poorly constructed or even overlooked.

An overarching goal of our research is to teach others to apply the SERA Method. To facilitate the transition of the SERA Method to adopters throughout the cybersecurity community, we explored more systematic ways of developing scenarios. As a result, we chartered a research task to explore the concept of using patterns of threats, called *threat archetypes*, to facilitate the process of scenario development.

[1] Flight software is a type of embedded real-time software used in avionics.

This report examines the concept of threat archetypes and how analysts can use them during scenario development. Before diving into the details of threat archetypes, we begin by providing a short overview of the SERA Method and cyber-risk scenarios.

2 SERA Method Overview

The SERA Method defines a scenario-based approach for analyzing complex cybersecurity risks in cyber-physical systems across the lifecycle and supply chain [Alberts 2016]. The SERA Method incorporates a variety of diagrams or models that can be analyzed at any point in the acquisition-and-development lifecycle to (1) identify security threats and vulnerabilities and (2) construct security risk scenarios. An organization can then use those scenarios to focus its resources on controlling its most significant security risks.

The person or group responsible for acquiring and developing a software-reliant system can apply the SERA method, or external parties can facilitate applying the method on behalf of the responsible person or group. Either way, a small team of approximately three to five people, called the *Analysis Team*, is responsible for implementing the method and reporting findings to stakeholders.

The Analysis Team is an interdisciplinary team that requires members who have diverse skill sets. Examples of skills and experience that should be considered when forming a team include the following: security-engineering risk analysis, systems engineering, software engineering, operational cybersecurity and physical/facility security. The exact composition of the Analysis Team depends on the point in the lifecycle in which the SERA Method is being applied and the nature of the engineering activity being pursued. Table 1 highlights the four tasks that the team performs when conducting the method [Alberts 2016].

Table 1: SERA Method Tasks

Task 1: Establish Operational Context			
Description	Task 1 defines the operational context for the analysis. The Analysis Team compiles/develops operational views that (1) define the mission and (2) document how systems and software support mission execution. Task 1 sets the context and scope for subsequent risk analysis activities. This task is important because it defines a performance baseline for each selected mission. This baseline establishes what is considered to be normal operational performance during mission execution as well as the systems and software components that support mission execution. The Analysis Team determines which mission to select as the basis for the analysis. The team then identifies the SoS that supports that mission. After further analysis of the SoS, the Analysis Team selects one or more entities of interest. An entity of interest is the system, subsystem, component, or software application that will be the focus of the cyber-risk analysis. The team then analyzes cyber risks for each entity of interest.		
Outputs	• mission thread(s) • SoS diagram(s) • system architecture diagrams	• software architecture diagrams • dataflow diagrams • use cases	• data security attributes • network topology diagrams • other diagrams as appropriate

Task 2: Identify Risk	
Description	Task 2 defines the cyber-risk identification activities for the SERA Method. In this task, the Analysis Team transforms security concerns into distinct, tangible cyber-risk scenarios that can be described and measured. The Analysis Team reviews the operational context documented in Task 1 as well as relevant data for each entity of interest. The team then develops a set of cyber-risk scenarios for each selected entity of interest.
Outputs	cyber-risk scenarios
Task 3: Analyze Risk	
Description	Task 3 focuses on the cyber-risk analysis. The Analysis Team evaluates each cyber-risk scenario in relation to predefined criteria to determine its probability, impact, and risk exposure.
Outputs	risk measures (e.g., probability, impact, risk exposure) for cyber-risk scenarios
Task 4: Develop a Control Plan	
Description	Task 4 establishes a plan for controlling a selected set of cyber-risk scenarios, which the team prioritizes based on their risk measures. The team then determines the basic approach for controlling each risk (i.e., accept or plan) based on predefined criteria and current constraints (e.g., resources and funding available for control activities). Finally, the Analysis Team develops a control plan for each risk that is not accepted.
Outputs	• prioritized cyber-risk scenarios • a control plan for each high-priority cyber-risk scenario

The SERA Method incorporates a top-down analysis approach, which establishes a line of sight from a mission to the hardware, software, and firmware that support it. Given the size, scale, and complexity of today's cyber-physical systems, the Analysis Team must focus on the most critical components of a system when conducting an assessment. As a result, setting a manageable scope for the assessment is essential for conducting a successful SERA.

Task 1 of the SERA Method includes two scoping activities:

1. The Analysis Team must determine which mission threads to include in the analysis. A mission thread sets the overall scope of the assessment. If the Analysis Team selects more than one mission thread to analyze, the team will need to conduct a SERA for each mission thread.
2. The Analysis Team selects entities of interest. The team selects the systems, subsystems, components, or software applications that are critical to the execution of a mission thread. A comprehensive cyber-risk analysis of each entity of interest is then performed. SERA's scoping activities thus provide the basis for conducting a deep-dive cyber-risk assessment of a mission thread's most critical assets.

Setting the scope of the assessment is just one aspect of Task 1. Once the assessment's scope is determined, the Analysis Team establishes a baseline of expected operational performance for the mission. At this point, the Analysis Team compiles (and in some cases develops) operational views that (1) define the mission and (2) document how systems, hardware, software, and firmware support the execution of that mission. These operational views include diagrams or models that describe the mission thread, system and software architectures, interfaces and dataflows, use cases, and the network topology (among others). In Task 1, the operational baseline anchors the subsequent cyber-risk analysis, where the Analysis Team develops and evaluates scenarios designed to subvert expected operational performance and disrupt the mission.

The SERA Method's scenario-based analysis is a distinguishing feature of the method that enables the Analysis Team to evaluate cybersecurity risks in complex, highly networked SoS environments. The details of SERA cyber-risk scenarios are presented in the next section.

3 Cyber-Risk Scenarios

According to the *Continuous Risk Management Guidebook*, a general definition for the term *risk* is “the probability of suffering harm or loss” [Dorofee 1996]. Based on this definition, risk comprises two core components: (1) an event that might or might not occur and (2) the magnitude of the resulting loss. This general definition of risk is applied universally, but different audiences attach different meanings to it. For example, safety professionals view risk management in terms of reducing the number of accidents and injuries. A hospital administrator views risk management as part of the organization’s quality assurance program. The insurance industry views risk as a basis for setting insurance rates. Each industry uses a definition that is tailored to its context. Consequently, no universally accepted definition of risk exists.

From a cybersecurity perspective, risk is viewed as a measure of the (1) likelihood that a threat will produce an adverse consequence or loss, and (2) magnitude of the consequence or loss. Based on this perspective, cybersecurity risk comprises the following components:

- *threat*—a cyber-based act, occurrence, or event with the potential to harm an information system through unauthorized access, destruction, disclosure, modification of data, and/or denial of service[2]
- *consequence*—the harm or loss resulting from the occurrence of a threat

The SERA Method uses scenarios to describe cybersecurity risks. The concepts of threat and consequence provide the basis for documenting cyber-risk scenarios.

3.1 Structure of Cyber-Risk Scenarios

A *cyber-risk scenario* tells a story of how one or more actors can cause adverse mission consequences for stakeholders by exploiting vulnerabilities or weaknesses (either deliberately or accidentally) in one or more interconnected software-reliant systems. A SERA cyber-risk scenario comprises the following three essential parts:

1. *SoS attack vector*—how a threat actor exploits one or more vulnerabilities to traverse an SoS environment and gain access to the target of the cyber attack (An SoS attack vector is also referred to as an *access path*.)
2. *cyber attack*—steps that a threat actor takes to launch a cyber attack on the selected target (i.e., entity of interest) by exploiting weaknesses or vulnerabilities and a description of the direct outcome of the attack (e.g., data disclosure, data modification, data unavailability)
3. *mission consequence*—description of a cyber attack’s impact on the mission (based on mission thread analysis)

Figure 1 illustrates the relationship between a SERA cyber-risk scenario and the components of cybersecurity risk. The threat component of risk includes the SoS attack vector and cyber attack. From the SERA perspective, the consequence of interest is the mission consequence. The

[2] This definition of *threat* is derived from *Security and Privacy Controls for Federal Information Systems and Organizations* [NIST 2013].

scenario's impact (i.e., measure of harm or loss) is based on an assessment of the mission consequence.

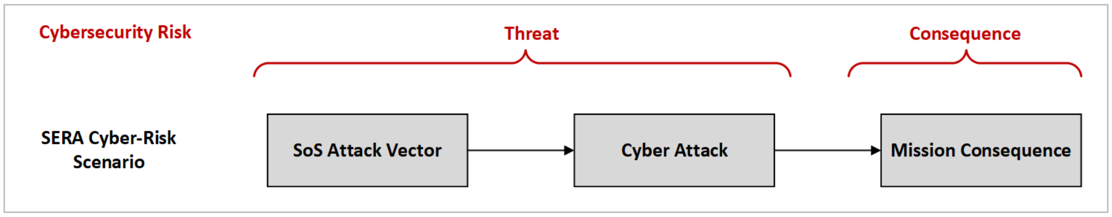

Figure 1: Components of Cybersecurity Risk Mapped to a SERA Cyber-Risk Scenario

3.2 Developing and Analyzing Cyber-Risk Scenarios

The Analysis Team officially starts to develop cyber-risk scenarios during Task 2 of the SERA Method. Each scenario is based on the structure defined above: SoS attack vector, cyber attack, and mission consequence. While the fundamental structure and content of a scenario is established during Task 2, the context needed to build it is collected during Task 1. In addition, the Analysis Team documents analysis data for each scenario during Task 3 (evaluation and prioritization) and Task 4 (control plan development), which the team uses to further refine the scenario. As a result, scenario development spans all four SERA Tasks. The remainder of this section examines how cyber-scenarios are developed, using their structure as a touchstone.

During Task 1, the Analysis Team establishes the operational context for the cyber-risk analysis. As noted in Section 2, Task 1 requires the team to compile or develop operational views that establish a line of sight from a mission thread to the hardware, software, and firmware that support it. Before moving to Task 2, the team selects one or more entities of interest. Within the context of the SERA Method, an entity of interest is defined as the system, subsystem, component, or software application that will be the focus of the cyber-risk analysis. Cyber-risk scenarios are subsequently developed and analyzed for each entity of interest.

Task 2 of the SERA Method focuses on cyber-risk identification. The team begins by reviewing the operational context documented in Task 1 of the SERA Method as well as relevant data for each entity of interest. Relevant data for an entity of interest can include requirements specifications; architecture documentation; and the results of safety assessments, security assessments, and risk assessments that were conducted. The Analysis Team uses the data to understand how the entity of interest is supposed to function within its mission context. Team members use this knowledge to devise ways to subvert expected operational performance. In other words, team members adopt the mindset of a malicious actor when developing cyber-risk scenarios.

The end goal of a malicious actor is to disrupt the mission thread. From the actor's perspective, a successful cyber attack leads to mission failure. To produce the desired mission consequence via a cyber attack, the actor must first gain access to the entity of interest (i.e., the attack target). Malicious actors often must traverse circuitous routes though an SoS environment to gain access to the entity of interest. Many systems in an SoS attack vector do not directly support the mission thread. In other words, many systems exploited in a given SoS attack vector do not have an obvious line of sight to the targeted mission thread. These systems might be part of enterprise information systems, supply chain systems, development systems, and test systems (among others). Network topology diagrams are useful for identifying SoS attack vectors.

Once an actor gains access to an entity of interest, they are in position to execute a cyber attack on that entity. The actor ultimately seeks to violate the security attributes of mission data with the hope of causing a range of indirect, negative consequences for mission stakeholders. Data security attributes indicate what qualities of a data asset are important to protect; they also provide insight into an actor's cyber-attack goal. A data asset has three basic security attributes: (1) confidentiality, (2) integrity, and (3) availability.[3] For a given cyber attack, an actor generally is trying to produce one or more of the following outcomes:

- data disclosure (confidentiality)
- data modification (integrity)
- insertion of false data (integrity)
- destruction of data (availability)
- interruption of access to data (availability)

The direct consequence of a cyber attack typically produces indirect consequences on the system, operational SoS, and the mission thread (i.e., mission consequences). In the SERA Method, the impact of a cyber-risk scenario is based on an analysis of the mission consequences. When developing a cyber-risk scenario, the Analysis Team must determine the extent to which the cyber attack will lead to mission degradation or mission failure. Once all relevant scenarios are developed and documented, Task 2 is considered to be complete.

Task 3 builds on the results of Task 2 by requiring the Analysis Team to evaluate the probability, impact, and risk exposure of each cyber-risk scenario. Finally, during Task 4, the team establishes a plan for controlling each high-priority cyber-risk scenario, marking the conclusion of the SERA Method.

Based on our experiences from piloting the SERA Method, we identified the development of cyber-risk scenarios as the key to a successful assessment. Scenarios are effective for capturing the complexities and nuances of cybersecurity risks in SoS environments [Alberts 2016]. However, scenario-based analysis is not without its drawbacks. In particular, we observed the following issues:

- Scenario development is time consuming.
- The composition of the Analysis Team affects the quality of the scenarios it develops.

Because of these observations, we explored more systematic ways of developing scenarios. As a result, we developed the concept of threat archetypes to (1) accelerate cyber-risk scenario development and (2) ensure the consistency of SERA results across Analysis Teams.

[3] *Confidentiality* is defined as keeping proprietary, sensitive, or personal information private and inaccessible to anyone who is not authorized to see it. *Integrity* is defined as the authenticity, accuracy, and completeness of data. *Availability* is defined as the extent to which, or frequency with which, data mist be present or ready for use.

4 SERA Threat Archetypes

The term *archetype* is defined as the original pattern or model of which all things of the same type are representations or copies [Merriam-Webster 2018]. Peter Senge applied the concept of archetypes to systems in 1991 by describing nine patterns of counterproductive behavior that affect system performance [Senge 1991]. SEI researchers applied the concept of archetypes to the DoD's acquisition of software-intensive systems [Novak 2010]. In developing the SERA Method, we use archetypes to describe a range of threats that Analysis Teams should consider during risk identification and analysis.

As used within the SERA context, a *threat archetype* is a pattern or model that describes a cyber-based act, occurrence, or event with the potential to harm an information system through unauthorized access, destruction, disclosure, or modification of data, and/or denial of service. A threat archetype defines the essence of the threat, not the specific steps required to gain access to a target and execute a cyber-attack.

When developing a cyber-risk scenario, the Analysis Team analyzes how a threat archetype's pattern can be realized in the target operational environment. Based on this analysis, the team then documents a tangible sequence of threat steps that a malicious actor can execute to access the entity of interest and execute a cyber-attack.

4.1 Threat Archetype: Structure and Elements

A key output of our SERA transition work is the prototype structure for describing threat archetypes.[4] As illustrated in Figure 2, a threat archetype comprises the following elements:

- actor
- threat type
- access type
- access point
- attack pattern[5]
- direct consequence

Figure 2 shows the relationship among threat archetype elements and the two threat components of SERA cyber-risk scenarios (i.e., SoS attack vector and cyber attack). Figure 2 also highlights the set of attributes used to describe relevant characteristics (i.e., qualities) of each element. Attributes can be customized, as appropriate, based on an organization's mission context and technology environment. The attributes presented in this section are a generic set.

[4] The structure used to describe threat archetypes is a protype structure that is still under development. The format and structure might change over time as we pilot threat archetypes and collect lessons learned.

[5] Each attack pattern is followed by a number. That number corresponds to MITRE's Common Attack Pattern Enumeration and Classification (CAPEC™) [MITRE 2020a].

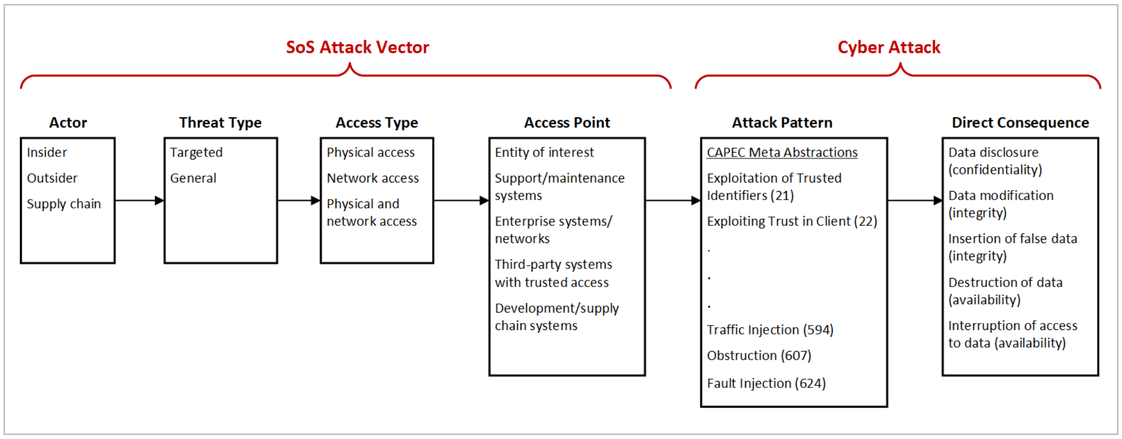
SoS Attack Vector
Cyber Attack
Actor
Insider
Outsider
Supply chain
Threat Type
Targeted
General
Access Type
Physical access
Network access
Physical and network access
Access Point
Entity of interest
Support/maintenance systems
Enterprise systems/ networks
Third-party systems with trusted access
Development/supply chain systems
Attack Pattern
CAPEC Meta Abstractions
Exploitation of Trusted Identifiers (21)
Exploiting Trust in Client (22)
.
.
.
Traffic Injection (594)
Obstruction (607)
Fault Injection (624)
Direct Consequence
Data disclosure (confidentiality)
Data modification (integrity)
Insertion of false data (integrity)
Destruction of data (availability)
Interruption of access to data (availability)

Figure 2: Threat Archetype Elements

In the remainder of this section, we describe each threat archetype element and its associated attributes. We conclude the section by providing examples of threat archetypes we developed.

Actor

From the SERA perspective, an *actor* is defined as the person or group that initiates a cyber attack. The actor's motive is assumed to be intentional/malicious in nature; accidental actions are beyond the scope of the SERA Method. The prototype archetype structure includes the following choices for actor:

- insider—An actor has legitimate access (cyber or physical) to the entity of interest and the mission it supports.
- outsider—An actor is not authorized to use the entity of interest and does not have physical access to it.
- supply chain—An actor is from an organization that provides products or services related to the entity of interest and the mission it supports.

Threat Type

The *threat type* describes an actor's focus when initiating a cyber attack. This element is related to the actor's motive. An actor desiring to inflict specific damage on an entity of interest directs the cyber-attack on that entity of interest or one of its components (e.g., a networking device). An actor with a general motive of vandalism (i.e., not directed toward a specific system or organization) might initiate an email phishing attack that randomly targets a broad audience. The prototype archetype structure includes the following choices for threat type:

- targeted—An actor directs a cyber-attack on an entity of interest or one of its components.
- general—An actor initiates a broad-based cyber attack that does not target a specific entity of interest or mission.

Access Type

As defined in the SERA Method, an attack vector describes how an actor traverses an SoS environment to gain access to the target of the cyber attack. *Access type* describes the path an actor uses to execute a cyber attack. The prototype archetype structure includes the following choices for access type:

- physical access—The actor has direct physical access to the target of the attack (i.e., physical access to the entity or interest of one of its components).
- network access—The actor launches a cyber-attack using network access only (i.e., no physical access to the entity of interest or one of its components).
- physical and network access—The actor uses a combination of physical and network access to attack the entity of interest. For example, the actor might use physical access to a system (e.g., support systems for the entity of interest) to initiate a network-based attack on the entity of interest.

Access Point

An SoS attack vector often requires an actor to move from system to system in an attempt to get in position to launch a cyber attack. The *access point* marks the beginning of the attack vector. It defines the initial breach in a potential chain of events that ultimately leads to a cyber attack on the entity of interest. The prototype archetype structure includes the following choices for access point:

- entity of interest—The actor accesses the system, subsystem, component, or software application that the actor is targeting.
- support/maintenance systems—The actor accesses systems used to support the entity of interest (e.g., software update system).
- enterprise systems/networks—The actor accesses systems and networks in the organization that operate the entity of interest; these systems might not directly support the entity of interest.
- third-party systems with trusted access—The actor accesses systems owned and operated by third parties (e.g., suppliers, vendors) that have trusted access to enterprise systems and networks operated by the organization.[6]
- development/supply chain systems—The actor accesses systems used in the development of an entity of interest's components (e.g., subsystems, components, or software applications).

Attack Pattern

An *attack pattern* describes a common method or approach for exploiting known weaknesses in software-enabled systems. SERA threat archetypes incorporate attack patterns documented in MITRE's Common Attack Pattern Enumeration and Classification (CAPEC™) [MITRE 2020a]. CAPEC provides a publicly available catalog of common attack patterns that helps users understand how adversaries exploit weaknesses in applications and other cyber-enabled capabilities.

CAPEC includes multiple ways to view attack patterns. One such view is the *meta abstraction*. A meta-level attack pattern in CAPEC provides an abstract characterization of a specific methodology or technique used in an attack. A meta abstraction provides a high-level understanding of a cyber attack and typically does not include specific technologies or implementations. Meta-level attack patterns are particularly useful for architecture-and-design threat modeling exercises

[6] Third-party systems with trusted access include services delivered by a cloud service provider (CSP).

[MITRE 2020b]. Table 2 provides a list of the CAPEC meta abstractions included in SERA threat archetypes.[7] Details about the CAPEC meta abstractions are found on MITRE's CAPEC website [MITRE 2020a].[8]

Table 2: Attack Patterns (CAPEC Meta Abstractions)

CAPEC Meta Abstractions			
Exploitation of Trusted Identifiers (21) Exploiting Trust in Client (22) Forced Deadlock (25) Leveraging Race Conditions (26) Fuzzing (28) Manipulating User State (74) Man in the Middle Attack (94) Brute Force (112) API Manipulation (113) Authentication Abuse (114) Authentication Bypass (115) Excavation (116) Interception (117) Privilege Abuse (122)	Buffer Manipulation (123) Shared Data Manipulation (124) Flooding (125) Pointer Manipulation (129) Excessive Allocation (130) Resource Leak Exposure (131) Parameter Injection (137) Content Spoofing (148) Identity Spoofing (151) Input Data Manipulation (153) Resource Location Spoofing (154) Infrastructure Manipulation (161) File Manipulation (165) Footprinting (169) Action Spoofing (173)	Code Inclusion (175) Configuration/Environment Manipulation (176) Software Integrity Attack (184) Reverse Engineering (188) Protocol Analysis (192) Functionality Misuse (212) Communication Channel Manipulation (216) Fingerprinting (224) Sustained Client Engagement (227) Privilege Escalation (233) Resource Injection (240) Code Injection (242) Command Injection (248) Protocol Manipulation (272) Bypassing Physical Security (390)	Information Elicitation (410) Manipulate Human Behavior (416) Modification During Manufacture (438) Manipulation During Distribution (439) Hardware Integrity Attack (440) Malicious Logic Insertion (441) Physical Theft (507) Contaminate Resource (548) Local Execution of Code (549) Functionality Bypass (554) Object Injection (586) Traffic Injection (594) Obstruction (607) Fault Injection (624)

A SERA threat archetype includes one or more attack patterns, depending on the nature and complexity of the attack.

[7] CAPEC provides a publicly available catalog of attack patterns along with a comprehensive schema and classification taxonomy. Each CAPEC attack pattern is associated with a unique CAPEC-ID Number. The numbers in Table 2 are the CAPEC-ID Numbers for the meta abstractions.

[8] https://capec.mitre.org/

Direct Consequence

A *direct consequence* of a threat describes the effect of a cyber attack on the data that are processed, stored, and transmitted by an entity of interest. The prototype archetype structure includes the following choices for direct consequence:

- data disclosure (confidentiality)
- data modification (integrity)
- insertion of false data (integrity)
- destruction of data (availability)
- interruption of access to data (availability)

Each direct consequence in the above list is mapped to its related security attribute (i.e., confidentiality, integrity, or availability).

4.2 Example Threat Archetypes

We developed the concept of threat archetypes to facilitate constructing cyber-risk scenarios. A threat archetype highlights the key characteristics (i.e., essence) of a cyber threat. After we developed the structure for documenting threat archetypes, we started reviewing cyber-risk scenarios that we developed during previous SERA pilots and identified potential patterns. We extracted threat attributes from those scenarios and removed pilot-specific details, producing an initial compilation of threat archetypes. In the remainder of this section, we present the following two examples of threat archetypes:

- Insider Uploads Malicious Code to Initiate a Denial-of-Service (DoS) Attack
- Insider Performs Reconnaissance Activities

We start by examining the details for an insider uploading malicious code to initiate a DoS attack.

Threat Archetype 1: Insider Uploads Malicious Code to Initiate a DoS Attack

Table 3 depicts a threat archetype for an insider uploading malicious code via a support/maintenance system to initiate a DoS attack. Attributes in the first four rows of Table 3 describe the SoS attack vector. Based on those attributes, the SoS attack vector can be phrased as *An insider uses physical access to the software maintenance system to upload malicious code designed to execute a targeted attack on the entity of interest*.

Table 3: Threat Archetype for Insider Uploads Malicious Code to Initiate a DoS Attack

Element	Attribute
Actor	Insider
Threat Type:	Targeted
Access Type	Physical
Access Point	Support/maintenance systems
Attack Pattern	Local Execution of Code (CAPEC-549) Flooding (CAPEC-125)
Direct Consequence	Interruption of access to data (availability)

The last two rows of Table 3 (Attack Pattern and Direct Consequence) include attributes that describe a cyber attack. The table documents two CAPEC attack patterns, Local Execution of Code (CAPEC-549) and Flooding (CAPEC-125). *Local Execution of Code* is a situation where an adversary installs and executes malicious code on the target system in an effort to achieve a negative technical impact [MITRE 2020c]. With *Flooding*, an adversary consumes the resources of a target by rapidly engaging in a large number of interactions with the target [MITRE 2020d]. The direct consequence noted in the table is "interruption of access to data" (i.e., the result of a DoS attack).

We can use the above information about the SoS attack vector and cyber attack to document a summary statement and narrative text for a threat archetype. The summary statement for the threat archetype from Table 3 is *An insider uses physical access to the software maintenance system to upload malicious code designed to flood the entity-of-interest's network with traffic and prevent the entity of interest from performing its function*.

The narrative text elaborates on the summary statement and is based on the attributes of the threat archetype. However, the text can include additional information for context and clarity. The following narrative is based on the attributes listed in Table 3:

> *An insider with malicious intent decides to execute a DoS attack on the entity of interest. The insider develops malicious code designed to flood the entity's network with traffic. The insider has physical access to the software maintenance system. The insider uploads the malicious code to the entity of interest via the software maintenance system.*
>
> *After the entity of interest begins its mission, the malicious code monitors network traffic. When the malicious code detects specific triggers in the network traffic, it initiates the cyber attack. The malicious code floods the network with illegitimate traffic. Processing illegitimate requests consumes available network resources, which creates a DoS. The entity of interest cannot perform its function.*

As illustrated in this narrative, a threat archetype defines the high-level characteristics of a cyber-based act, occurrence, or event that can lead to an undesirable outcome, such as the unauthorized access, destruction, disclosure, or modification of data, and/or denial of service. This threat archetype documents a pattern where an insider triggers a DoS attack on an entity of interest. The next threat archetype explores how an insider can gain unauthorized access to sensitive documents.

Threat Archetype 2: Insider Performs Reconnaissance Activities

Table 4 depicts a threat archetype for an insider performing research and reconnaissance activities. In this archetype, the insider's goal is to view information that they are not authorized to see (i.e., a data confidentiality breach). The SoS attack vector for this threat archetype is *An insider uses physical and network access to enterprise systems and networks to execute a targeted attack*.

Table 4: Threat Archetype for Insider Performs Reconnaissance Activities

Element	Attribute
Actor	Insider
Threat Type	Targeted
Access Type	Physical and network
Access Point	Enterprise systems/networks
Attack Pattern	Privilege Abuse (CAPEC-122) Bypassing Physical Security (CAPEC-390) Research and Reconnaissance
Direct Consequence	Data disclosure (confidentiality)

Table 4 highlights three attack patterns for the threat archetype: (1) Privilege Abuse (CAPEC-122), (2) Bypassing Physical Security (CAPEC-390), and (3) Research and Reconnaissance. *Privilege Abuse* is when an adversary is able to access resources that are intended only for higher level users because access control mechanisms are absent or misconfigured. With *Bypassing Physical Security*, an adversary is able to evade building security and surveillance methods and circumvent electronic or physical locks used to secure entry points. *Research and Reconnaissance* is a custom attack pattern. Here, an adversary uses access to documentation about a targeted system (e.g., architecture, design, configuration manuals) to understand how it operates. CAPEC does not define an attack pattern specifically for this type of research-and-reconnaissance activity.[9] As illustrated in the table, the result of the attack is disclosure of sensitive data to an unauthorized party. The summary statement for the threat archetype from Table 4 is *An insider bypasses physical and cybersecurity controls to view sensitive documents about the entity of interest*.

The following text provides a more detailed description of the threat archetype documented in Table 4:

> *An insider with malicious intent wants to examine sensitive documents for the entity of interest. The insider bypasses physical security controls (e.g., building security and surveillance methods, electronic or physical) to gain access to hard copies of documentation for the entity of interest. Using legitimate network access to enterprise systems, the insider exploits weaknesses or vulnerabilities in enterprise systems to view electronic documentation for the entity of interest.*

Analysts can develop a library of threat archetypes, such as those featured in this section, for their organizations. A relevant subset of those organizational archetypes can be selected for a given

[9] *Footprinting* (CAPEC-169) defines a situation where an adversary engages in probing and exploration activities to identify constituents and the properties of a target. Footprinting describes a variety of information-gathering techniques used by an adversary to prepare for an attack. It includes using tools to learn as much as possible about the composition, configuration, and security mechanisms of a targeted application, system, or network. Footprinting defines a tool-based activity for learning about a target of an attack. *Research and Reconnaissance* is focused on using a target's documentation to learn more about its design and use.

assessment based on the entity of interest and the mission it supports. Those selected archetypes form the basis for the cyber-risk scenarios that are created and analyzed during a SERA.

The next sections of this report provide an example of how we use threat archetypes when developing SERA cyber-risk scenarios. The section begins with the example for SERA Task 1, where the Analysis Team establishes the operational context for the assessment.

5 Example for SERA Task 1: Establishing Operational Context

During Task 1 of the SERA Method, the Analysis Team compiles/develops operational views that (1) define the mission and (2) document how systems and software support the execution of that mission. Task 1 thus sets the context and scope for subsequent risk analysis activities. Threat archetypes are not used during Task 1. However, the data generated during Task 1 provides the basis for SERA's threat modeling and cyber-risk scenario development activities featured in Task 2. The application of threat archetypes begins with a description of the context in which those archetypes will be used.

This example is a composite of several cyber-risk analyses that we conducted for the DoD during the past five years using the SERA Method. The operational context described in this section is for illustrative purposes only; it does not reflect actual results from any of our SERA pilots. We also simplified the architectural views to highlight key points related to scenario development.

This example features a system that integrates tactical data from multiple sources and creates a single graphical representation of a battlespace. This section describes key operational views compiled or developed during Task 1 of the SERA Method. The starting point for Task 1 is the development of the mission thread(s) that the system supports.

5.1 Data Fusion System (DFS) Mission Thread

A command-and-control group is acquiring a Data Fusion System (DFS) to support strategic and tactical decision making. The DFS will provide a single graphical representation of the battlespace by integrating tactical data from multiple sources, including data-link networks, ground networks, intelligence networks, and sensor networks. The DFS accomplishes its mission by

- collecting data from multiple sources
- analyzing the collected data
- presenting a unified picture of the battlespace on 10 mission consoles staffed by operators
- forwarding analyzed data to other systems

Figure 3 depicts the command-and-control mission thread that the DFS supports.[10]

[10] We used a swimlane diagram to document the DFS mission thread. A swimlane diagram provides a visual representation of a mission thread. It defines the sequence of end-to-end activities that take place to achieve a mission as well as who performs each activity. The activities in a swimlane diagram are grouped visually by placing them in *lanes*. Parallel lines divide the diagram into multiple lanes, with one lane for each mission-thread actor (i.e., person, group, or sub-process). Each lane is labeled to show who is responsible for performing the activities assigned to that lane. In Figure 3, the boxes represent the activities that are performed by each mission-thread actor. Lines between the activities establish the relationships among and sequencing of the activities.

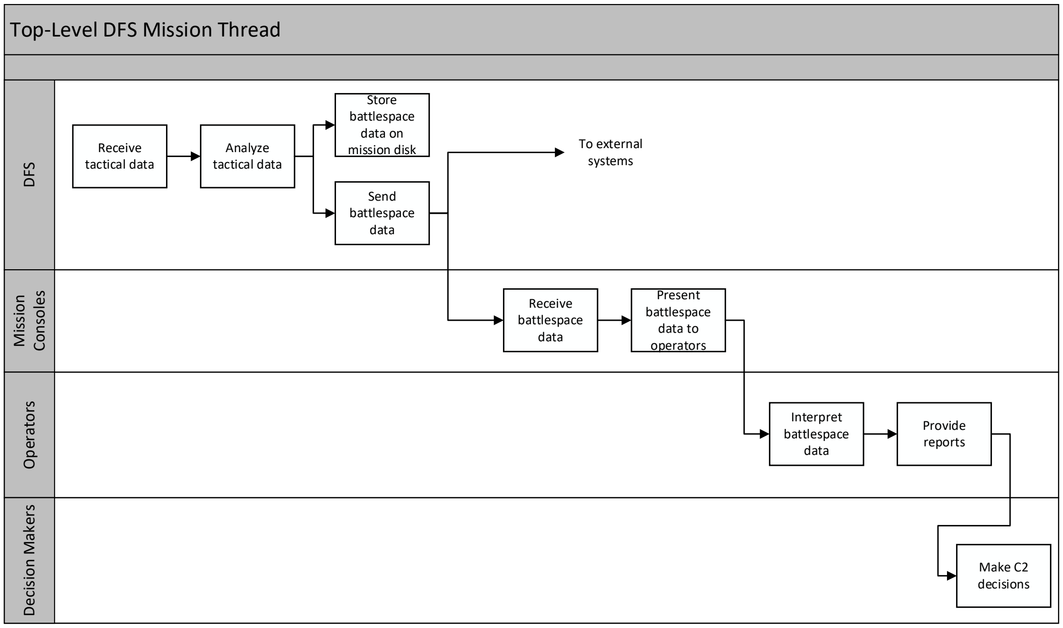

Figure 3: Top-Level DFS Mission Thread

During a mission, the DFS receives tactical data from data-link networks, ground networks, intelligence networks, and sensor networks. The DFS incorporates sensor integration software that analyzes tactical data and develops an integrated view of the battlespace. The DFS stores battlespace data on a mission disk and also forwards that data to mission consoles and external systems. Mission consoles receive and format battlespace data, presenting a graphical representation of the battlespace to operators. Operators analyze the battlespace picture and provide recommendations to senior leaders. Those leaders make command-and-control decisions about the battlespace based on the operators' recommendations.

5.2 DFS Operations and Support

Figure 4 depicts the interfaces between the DFS and two systems that support it: (1) the Mission Planning System (MPS) and (2) the Operations and Maintenance System (OMS). The MPS is a classified system with a direct interface to the DFS via a classified ground network. The MPS supports mission planning and preparation, mission briefing/debriefing activities, and the transfer of mission data. For system maintenance, the OMS provides mechanisms for uploading software patches and upgrades to the DFS.

A DSF-supported mission begins when senior leaders initiate planning and preparation activities in the MPS. Once all planning and preparation activities are complete, mission operators send the mission plan to the DFS via the classified ground network (Step 1 in Figure 4). Operators are briefed on mission details as they prepare to support the mission.

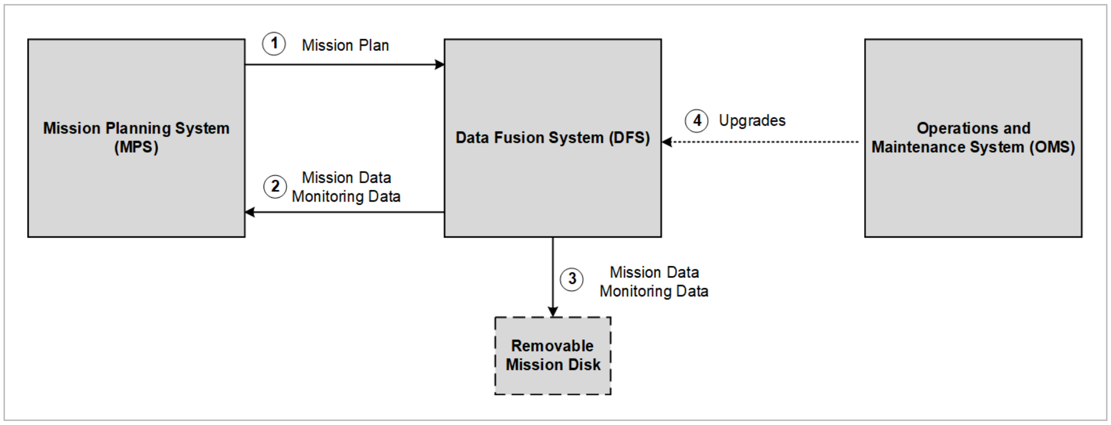
Mission Planning System (MPS)
1 Mission Plan
Data Fusion System (DFS)
4 Upgrades
Operations and Maintenance System (OMS)
2 Mission Data Monitoring Data
3 Mission Data Monitoring Data
Removable Mission Disk

Figure 4: DSF Interfaces

When it is time to execute the mission, a removeable system disk is inserted into the DFS. As the mission is being conducted, the DFS generates mission and monitoring data. Mission data include battlespace data, while monitoring data include information about the performance of DFS components and the DFS network. During mission execution, all mission and monitoring data are sent to the MPS via the classified ground network (Step 2 in Figure 4) and are recorded in the removable mission disk (Step 3). After the mission is complete, the mission disk is removed from the DFS and securely stored. Mission staff access the MPS to analyze mission data, looking for any trends, issues, and lessons learned.

In between missions, maintenance personnel analyze monitoring data from recent missions and perform maintenance activities on DFS software and hardware as required. Software updates, including software enhancements and security patches, are stored on the OMS by software maintenance personnel. After receiving appropriate approvals, software maintenance personnel prepare a disk with the software updates and upload the contents of that disk to a designated laptop. The laptop containing the software updates is connected to a network access port on the DFS, and the software is then uploaded to the DFS (Step 4).[11]

5.3 DFS Architecture

The network architecture of the DFS, as shown in Figure 5, comprises the following components:

- two local area network (LAN) switches
- two compute servers
- two storage servers
- one removeable mission disk

[11] The DFS is air gapped from the OMS. Software enhancements and security patches are transferred via a secure laptop connected to a network access port on the DFS. However, both the DFS and OMS are networked with other external systems. The DFS interfaces directly with the MPS and other external systems via the classified ground network. The OMS is connected to the command-and-control organization's enterprise network.

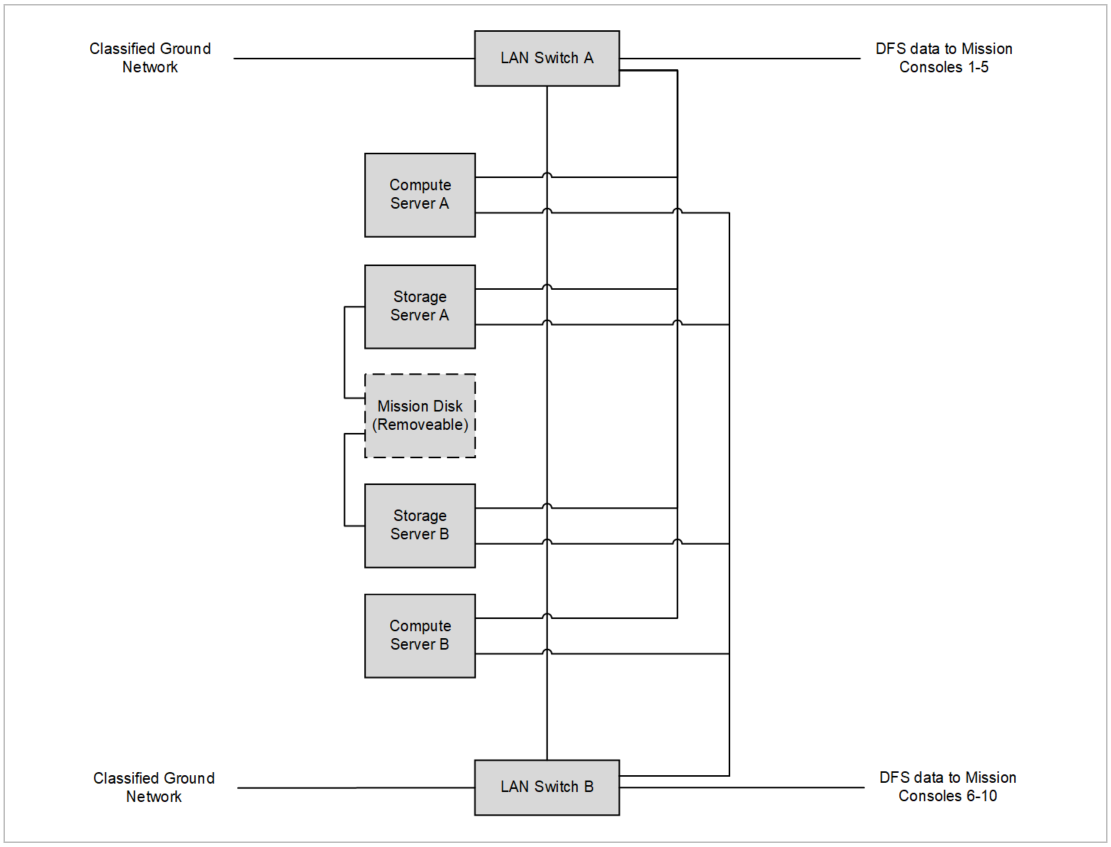
Classified Ground Network
LAN Switch A
DFS data to Mission Consoles 1-5
Compute Server A
Storage Server A
Mission Disk (Removeable)
Storage Server B
Compute Server B
Classified Ground Network
LAN Switch B
DFS data to Mission Consoles 6-10

Figure 5: DSF Network Architecture

During a mission, the DFS receives sensor data from air, maritime, and ground assets via a classified ground network. Sensor data are received by the two LAN switches and forwarded to the compute servers and storage servers. The compute servers execute the sensor-integration software that analyzes tactical data and develops an integrated view of the battlespace. The battlespace data are then forwarded to the following destinations via the two LAN switches:

- two storage servers
- ten mission consoles
- the MPS via the classified ground network
- external systems via the classified ground network

The storage servers also copy battlespace data to the removable mission disk. As the mission is being executed, system and network monitoring data are continually generated and sent to the MPS. In addition, monitoring data are forwarded to the two storage servers, which then copy the monitoring data to the removable mission disk.

Mission consoles 1-5 receive battlespace data from LAN switch A, while consoles 6-10 receive data from LAN switch B. Once the battlespace data are received, the mission consoles format the data and present a graphical representation of the battlespace to operators.

5.4 DFS in an SoS Context

An SoS environment describes how a software-reliant system must function as part of a multi-system ecosystem to achieve stakeholders' mission objectives. In an SoS environment, each system is managed independently from the others, which creates considerable operational complexity. From the SoS perspective, the DFS is the software-reliant system we are focusing on. The SoS environment for the DFS includes the following systems and assets:

- systems providing data via the classified ground network, including
  - data-link networks
  - ground networks
  - intelligence networks
  - sensor networks
  - the MPS
- the OMS
- enterprise systems networked with the OMS, including
  - enterprise business systems
  - DFS development systems
  - DFS test systems
  - the DFS engineering repository

When conducting a cyber-risk analysis, it is essential to understand the broader SoS environment in which a software-reliant system operates. Many attack vectors targeting a specific system include weaknesses exploited in the SoS environment. In the next section, we show how to use the mission context to develop cyber-risk scenarios.

6 Example for SERA Tasks 2-4: Analyzing Cyber-Risk Scenarios

In this section, we present the cyber-risk scenario that we developed for the DFS and its associated mission (SERA Task 2). We also briefly describe the results of the risk analysis that we performed for the scenario (SERA Task 3) and present a summary of controls that can be implemented to mitigate the risk posed by the scenario (SERA Task 4). We begin by examining how to apply threat archetypes when developing cyber-risk scenarios.

6.1 Applying Threat Archetypes During Scenario Development

We propose two alternatives for using threat archetypes when developing cyber-risk scenarios.

Alternative 1: Build a scenario using archetypes documented in a library of threat archetypes. In this approach, an organization defines and organizes a set of common threat archetypes for Analysis Teams to consider. When conducting a SERA, the Analysis Team selects threat archetypes that are applicable to the given mission context and uses them as the basis for developing cyber-risk scenarios.

Alternative 2: The Analysis Team selects attributes that describe a specific or unique threat pattern that has not been predefined in a library of threat archetypes. The team uses the threat-archetype structure presented sin Section 4.1 to create custom archetypes that it can incorporate into its analysis activities.

The Analysis Team can combine these two alternatives when developing cyber-risk scenarios. For example, the team could use several predefined archetypes and supplement those with additional custom-developed archetypes. The key is to select archetypes that fit the mission context and architecture. In the remainder of this section, we describe our approach for applying threat archetypes using the DFS example as a touchstone.

Applying Threat Archetype 1

Task 1 of the SERA Method requires the Analysis Team to define the scope of the analysis by selecting one or more entities of interest. For this example, we selected the DFS as our initial entity of interest. The DFS integrates intelligence data with surveillance, target acquisition, and reconnaissance data to improve a commander's situational awareness and decision making. These decisions ultimately support deployed air, maritime, and land forces. Cyber attacks on the DFS have the potential to disrupt the mission thread supported by the DFS.

During SERA Task 2, the Analysis Team reviews the operational context documented during Task 1 as well as relevant data for each entity of interest (e.g., system, subsystem, component, or software application) selected for detailed analysis. The team then develops a set of cyber-risk scenarios for each entity of interest. When conducting a cyber-risk analysis, the Analysis Team must identify ways to subvert the mission, leading to mission degradation or outright mission failure. As shown in Figure 6, we identified the following cyber-attack goals for the DFS mission thread:

- Candidate Goal 1: Prevent battlespace data from being sent to mission consoles (availability issue).
- Candidate Goal 2: Generate an inaccurate picture of the battlespace (integrity issue).

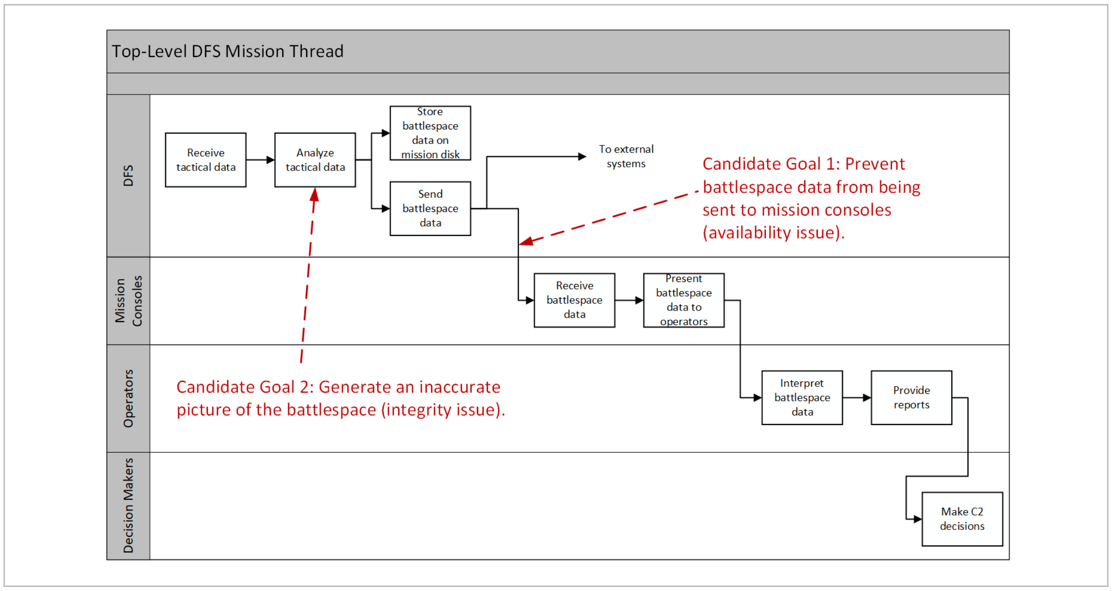
Top-Level DFS Mission Thread
DFS
Mission Consoles
Operators
Decision Makers
Receive tactical data
Analyze tactical data
Store battlespace data on mission disk
Send battlespace data
To external systems
Candidate Goal 1: Prevent battlespace data from being sent to mission consoles (availability issue).
Receive battlespace data
Present battlespace data to operators
Candidate Goal 2: Generate an inaccurate picture of the battlespace (integrity issue).
Interpret battlespace data
Provide reports
Make C2 decisions

Figure 6: Candidate Cyber-Attack Goals for the DSF Mission

In this section, we illustrate the development of a cyber-risk scenario for Candidate Goal 1, *Prevent battlespace data from being sent to mission consoles*. This goal, if successfully completed, will prevent operators from being able to observe and analyze battlespace data. The mission consequence for Candidate Goal 1 is the inability to complete the DFS mission (i.e., mission failure). Once the mission consequence is established, the Analysis Team turns its attention to identifying and analyzing threats that can produce that mission consequence (referred to as *threat modeling*).

For this example, we start with the first threat archetype that we presented earlier in this report. The summary statement for that archetype is *An insider uses physical access to the software maintenance system to upload malicious code designed to flood the entity-of-interest's network with traffic and prevent the entity of interest from performing its function.*[12]

Once an archetype is selected, the Analysis Team determines *how* to implement that archetype's pattern in the entity of interest and its supporting SoS environment. We typically start by focusing on the cyber attack, which includes the following two archetype elements: attack pattern and direct consequence. The selected archetype for this example comprises two attack patterns: (1) Local Execution of Code (CAPEC-549) and (2) Flooding (CAPEC-125). The direct consequence for the selected archetype is the interruption of access to data (i.e., the result of a DoS attack).

[12] A threat archetype defines the core characteristics of a cyber attack on a system, subsystem, component, or software application.

Developing an approach for implementing a flooding attack and creating a DoS in the DFS requires us to examine the DFS architecture in greater detail. We turn our attention toward the network diagram for the DFS, shown in Figure 7. As illustrated in the figure, the DFS architecture comprises

- two local area network (LAN) switches
- two compute servers
- two storage servers
- one removeable mission system disk

By analyzing the DFS network diagram and other operational information collected during Task 1, we identified several cyber-attack strategies targeting the DFS. The following targets were of particular interest:

- DFS switches
- DFS compute servers

As shown in Figure 7, we selected the flooding attack on the DFS switches as the basis for the cyber-risk scenario that we are developing. The switches are commercial-off-the-shelf devices with well-documented vulnerabilities. An attacker would have several tactical options for planting malicious code in the DFS and implementing a flooding attack.

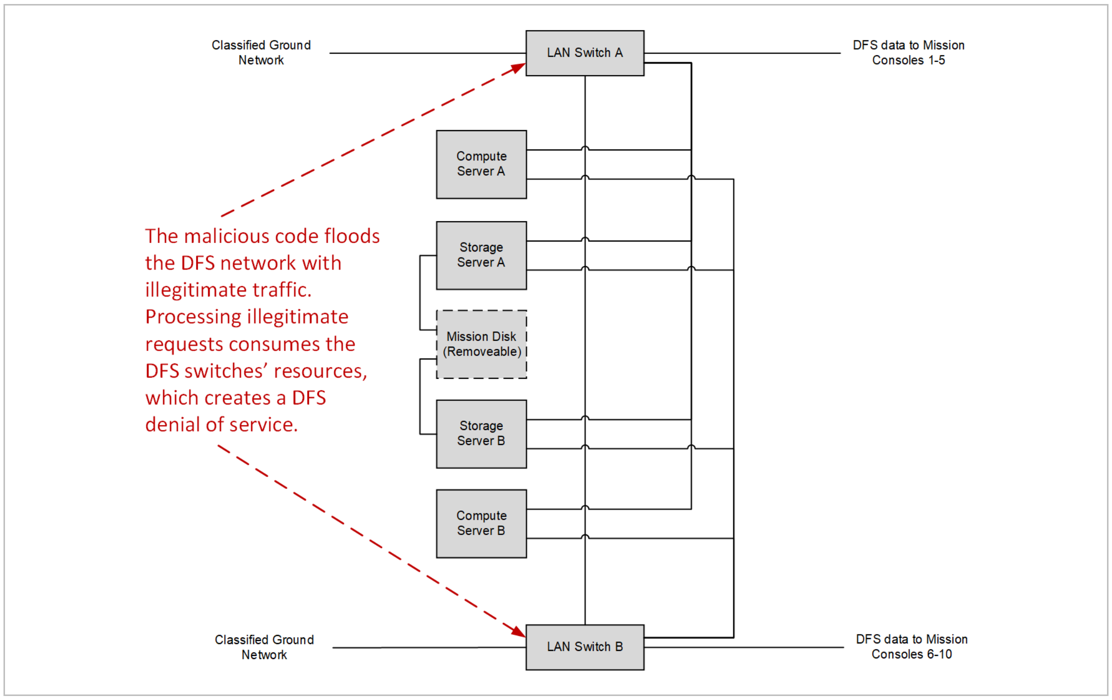
Classified Ground Network
LAN Switch A
DFS data to Mission Consoles 1-5
Compute Server A
Storage Server A
Mission Disk (Removeable)
Storage Server B
Compute Server B
The malicious code floods the DFS network with illegitimate traffic. Processing illegitimate requests consumes the DFS switches' resources, which creates a DFS denial of service.
Classified Ground Network
LAN Switch B
DFS data to Mission Consoles 6-10

Figure 7: Flooding Attack on DFS Switches

The final aspect of scenario development is the SoS attack vector or access path. The SoS attack vector included in the selected archetype is *An insider uses physical access to the software maintenance system to upload malicious code designed to execute a targeted attack on the entity of interest*.

Incorporating the SoS attack vector in the scenario begins with a review of SERA Task 1 data. As shown in Figure 8, an insider can use the software upgrade process to upload malicious code to the DFS.

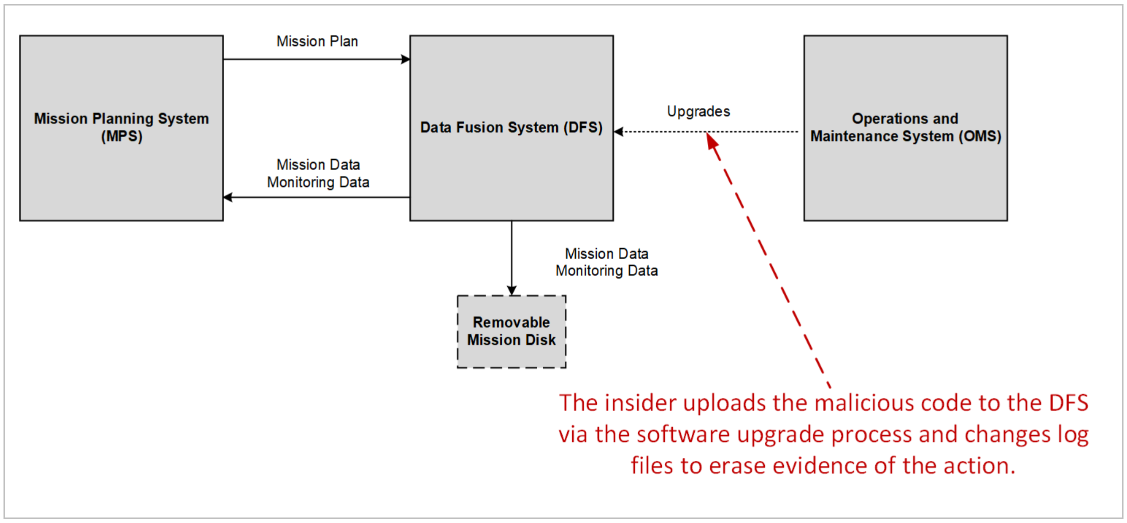
Mission Planning System (MPS)
Mission Plan
Data Fusion System (DFS)
Mission Data
Monitoring Data
Upgrades
Operations and Maintenance System (OMS)
Mission Data
Monitoring Data
Removable Mission Disk
The insider uploads the malicious code to the DFS via the software upgrade process and changes log files to erase evidence of the action.

Figure 8: SoS Attack Vector

The insider would perform the following actions to upload the malicious code to the DFS via the software upgrade process:

- Transfer the malicious code to the designated DFS laptop.
- Connect the laptop to a network access port on the DFS.
- Upload the malicious code to the DFS.
- Change log files to erase evidence of the action.

At this point, we addressed the three essential parts of a cyber-risk scenario: (1) the SoS attack vector, (2) the cyber attack, and (3) the mission consequence. We have enough information to develop a cyber-risk scenario. However, we can augment those three elements by adding additional steps, such as reconnaissance, planning, and preparation activities, as well as additional context. Next, we explore the addition of reconnaissance, planning, and preparation activities to the core scenario.

Applying Threat Archetype 2

The research and reconnaissance activities described in threat archetype 2 do not have a direct impact on the DFS mission thread. However, knowledge of the DFS architecture and operational specifications is information that an insider can use when developing a cyber attack. As a result, threat archetype 2 is considered to be a precursor to threat archetype 1. When adapting threat archetype 2 to the DFS mission context, we start by identifying the attack goal: *Collect information about the DFS architecture and operational specifications*.

The summary statement for threat archetype 2 is *An insider bypasses physical and cybersecurity controls to access and view sensitive documents about the entity of interest*. Threat archetype 2 comprises the following three attack patterns: (1) Privilege Abuse (CAPEC-122), (2) Bypassing Physical Security (CAPEC-390), and (3) Research and Reconnaissance. The direct consequence is disclosure of sensitive data (i.e., DFS architecture and operational specifications) to an unauthorized party (i.e., the insider).

To tailor threat archetype 2 to the DFS context, we need to identify where an insider can gain access to DFS architecture and operational specifications. From the SoS context provided in Section 5.4, we identified the following enterprise systems that support the DFS:

- enterprise business systems
- DFS development systems
- DFS test systems
- the DFS engineering repository

We examined the above systems and DSF mission context in greater detail and selected the DFS engineering repository and physical work space as likely targets of an attacker.[13] The insider can use the following attack vectors to perform research and reconnaissance on the DFS:

- cyber access to the DFS engineering repository due to insufficient access control mechanisms
- physical access to the DFS engineering organization's work space to view unsecured hard copies of DFS engineering documents

At this point, we have sufficient information to incorporate the research-and-reconnaissance archetype into a cyber-risk scenario. The next section provides a cyber-risk scenario that includes information that we developed using the two threat archetypes.

6.2 Cyber-Risk Scenario

When developing a cyber-risk scenario, we document the sequence of steps performed by the actor(s) when executing the threat and the mission consequences. For the DFS cyber-risk scenario, we identified the following nine threat steps:

1. An insider with technical skills and administrative access to the DFS becomes disgruntled after being passed over for a promotion and not receiving a bonus.
2. The insider begins to behave aggressively and abusively toward co-workers.
3. After a while, the insider decides to execute a cyber attack on the DFS. The insider's goal is to execute a DoS attack on DFS switches.
4. The insider uses cyber access to the DFS engineering repository (resulting from insufficient access control mechanisms) to view engineering documents. The insider uses physical access to the DFS engineering organization's work space to view unsecured hard copies of DFS engineering documents.
5. The insider develops a plan for the cyber attack based on the available information.
6. The insider uses the organization's resources to develop malicious code designed to flood the DFS network with traffic.

[13] The DFS engineering repository stores electronic copies of all DFS architecture documents and specifications. The physical work space for the DFS engineering organization contains hard copy versions of architecture documents and specifications that are not secured and tracked properly. Details about the DFS engineering repository and physical work space are beyond the scope of this report.

7. The insider transfers the malicious code to the designated DFS laptop, connects the laptop to a network access port on the DFS; uploads the malicious code to the DFS, and changes log files to erase evidence of the action.
8. After the DFS begins its mission, the malicious code monitors DFS network traffic.
9. When the data indicate that the DFS is receiving mission data, the malicious code's attack is triggered. The malicious code floods the DFS network with illegitimate traffic. Processing illegitimate requests consumes the DFS switches' resources, which creates a DFS DoS.

Steps 7-9 are based on threat archetype 1. Steps 4-6 deal with reconnaissance, planning, and preparation activities from threat archetype 2. Finally, Steps 1-3 provide context regarding the actor's motivation for executing the cyber attack on the DFS.[14] During Task 2 of the SERA Method, the Analysis Team documents the sequence of threat steps as well as the conditions and circumstances that facilitate the occurrence of each threat step (called *enablers*). The sequence of threat steps and associated enablers are documented in a *Threat Sequence Table*. (The appendix provides a Threat Sequence Table for the DFS cyber-risk scenario.)

The direct consequence of the above threat is the inability of the DFS to process legitimate routing requests (i.e., a DoS). This DFS DoS attack also produces indirect consequences that affect the mission thread supported by the DFS. Because these indirect consequences affect the mission thread, they are referred to as *mission consequences*. The DFS cyber-risk scenario also includes the following mission consequences:

- The mission supported by the DFS cannot be completed.
- The government suspends future use of the DFS pending the execution of the disaster recovery plan.

During SERA Task 2, the Analysis Team documents each mission consequence as well as conditions and circumstances that can increase the impact of that mission consequence (called *amplifiers*). The mission consequences and their associated amplifiers are documented in a Mission Consequence Table. (The appendix provides a Mission Consequence Table for the DFS cyber-risk scenario.)

The final step of developing a cyber-risk scenario is documenting the scenario in narrative form. We have found that some audiences prefer to start with the narrative and then delve into a scenario's details (provided in the tables). (We include a narrative for the DFS cyber-risk scenario in the appendix.)

[14] For the cyber-risk scenario, we added context regarding the insider's motivation at the request of the program for which we were performing the SERA. The program's stakeholders wanted to consider controls for early detection of an insider threat. We customized the scenario accordingly. This context was based on our review of SEI insider threat case studies.

6.3 Risk Analysis

The focus of SERA Task 3 is a cyber-risk analysis. During this task, the Analysis Team evaluates each cyber-risk scenario in relation to predefined criteria to determine its impact, probability, and risk exposure.

Impact is a measure of the loss that occurs when a risk is realized. For the DFS cyber-risk scenario, we measured impact using a five-point scale (maximum, high, medium, low, minimal). We defined *maximum* impact to be mission failure (i.e., the mission thread cannot be completed). When we developed the DFS cyber-risk scenario, we selected a cyber-attack that would cause the mission to fail. By definition, the impact value for the DFS cyber-risk scenario is *maximum*.

Probability is a measure of the likelihood that the risk will occur. We gauged probability using a five-point scale (frequent, likely, occasional, remote, rare). We defined a *rare* event as one that is uncommon or unusual. A rare event is not frequently experienced (less than one occurrence every five years). Given the complexity of the DFS cyber-risk scenario and the skills (and access) needed to execute it, we judged its probability value to be rare.

Risk exposure provides a measure of the magnitude of a risk based on the current values of probability and impact. Our risk-exposure criteria translated a *maximum* impact and *rare* probability to a *medium* risk exposure.

6.4 Candidate Controls

During Task 4, the Analysis Team establishes a plan for controlling all high-priority cyber-risk scenarios. We used the following guidelines to prioritize DFS cyber-risk scenarios:

- Impact is the primary factor for prioritizing cyber-risk scenarios. Scenarios with the *maximum* impacts are deemed to be the highest priority.
- Probability is the secondary factor for prioritizing cyber-risk scenarios. Probability is used to prioritize risks that have equal impacts. Risks of equal impact with the highest probabilities are considered to be the highest priority.

We determined that the DFS cyber-risk scenario featured in this report had *maximum* impact; as a result, we developed a control plan for that scenario. For each threat step documented in the Threat Sequence Table, we selected cybersecurity actions (i.e., cybersecurity controls) designed to mitigate the threat enablers associated with each step. Similarly, we determined which cybersecurity controls would dampen the consequence amplifiers associated with each consequence listed in the Mission Consequence Table.[15]

In both tables, we also mapped each candidate control to cybersecurity controls documented in the following documents from the National Institute of Standards and Technology (NIST):

- Security and Privacy Controls for Federal Information Systems and Organizations (NIST Special Publication 800-53 Revision 4) [NIST 2013]
- Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1 [NIST 2018]

[15] A Threat Sequence Table and Mission Consequence Table, including associated cybersecurity controls, are provided in the appendix.

The controls identified for the DFS cyber-risk scenario are summarized in Table 5.

Table 5: Candidate Controls for DFS Cyber-Risk Scenario

Category	Control
Access Control	The organization manages and protects physical access to information and resources.
	The organization manages access permissions and authorizations for computing resources.
Change Management	The organization implements a change management/configuration management system to track changes to the code base.
Code Analysis	The organization analyzes its code base for the presence of malicious code.
Disaster Recovery	The organization executes a recovery plan after a cybersecurity incident occurs.
Human Resources	The organization’s managers provide constructive feedback on performance issues.
	The organization's managers recognize inappropriate behavior when it occurs and respond appropriately.
	The organization's managers recognize an employee's escalating frustration and proactively work to defuse the situation.
	The organization performs targeted monitoring of individuals with suspected behavioral issues.
Incident Response	The organization responds appropriately when abnormal activity is detected.
	Mission personnel respond appropriately when abnormal system activity is detected.
Monitoring	The organization monitors the physical environment for abnormal activity.
	The organization monitors systems and networks for abnormal activity.
	Mission personnel monitor the system for abnormal activity during the mission.
System Architecture	The organization ensures that the system has adequate capacity to ensure availability is maintained.
	The system has mechanisms (e.g., hot backup system) to achieve mission resilience in normal and adverse situations
Training	The organization provides role-based security training to designated mission personnel.
	The organization provides role-based security training to the disaster recovery team.

This concludes the example illustrating how we use threat archetypes to develop cyber-risk scenarios. (The detailed analysis results for this cyber-risk scenario are provided in the appendix.) The next section summarizes the key lessons we learned from piloting threat archetypes and outlines potential next steps for this work.

7 Summary and Next Steps

Over the past five years, we conducted multiple pilots of the SERA Method for DoD and federal system acquisition programs. Based on our experiences from piloting the SERA Method, we identified the development of cyber-risk scenarios as the key to a successful assessment. We observed that scenario development is a time-consuming activity. We also observed that the composition of the Analysis Team affects the quality of the scenarios developed.

To facilitate the transition of the SERA Method to the cybersecurity community, we explored more systematic ways of developing scenarios. As a result, we developed the concept of threat archetypes to (1) accelerate cyber-risk scenario development and (2) ensure the consistency of SERA results across Analysis Teams.

As used within the SERA context, a *threat archetype* is a pattern or model that describes a cyber-based act, occurrence, or event with the potential to harm an information system through the unauthorized access, destruction, disclosure, or modification of data, and/or denial of service. A threat archetype defines the essence of the threat, not the specific steps required to gain access to a target and execute a cyber attack. We developed a prototype structure for threat archetypes comprising the following elements:

- actor
- threat type
- access type
- access point
- attack pattern
- direct consequence

Our structure for describing threat archetypes is a protype that is still under development. The format and structure might change over time as we pilot threat archetypes and collect lessons learned. In the near term, we plan to use threat archetypes to facilitate scenario development in our upcoming pilots. In particular, we will focus on the following aspects of scenario development:

- *time reduction in scenario development*—Decreasing the time required to develop cyber-risk scenarios is a transition goal for our SERA research. In future pilots, we plan to explore the extent to which threat archetypes reduce the time it takes the Analysis Team to develop SoS cyber-risk scenarios.
- *reproducibility of SERA results*—Ensuring reproducibility of SERA results is another transition goal. In future pilots, we plan to analyze scenarios produced by different Analysis Teams to assess the consistency of results.

Finally, the following are candidate next steps for future SERA transition tasks:

- *conduct additional pilots*—Additional pilots will enable us to enhance and improve the SERA Method based on lessons learned.
- *continue SERA transition activities*—Continuing SERA transition activities could be the focus of future SEI research-and-development activities. Examples of future transition activities include (1) developing and refining a generic SERA library of threats and (2) exploring ways

to automate the SERA Method (e.g., developing a SERA support tool, incorporating text analytics).

- *develop SERA Method training for practitioners*—Training is a core component of method transition. Currently, a SERA awareness tutorial is available as part of the CERT Cybersecurity Engineering and Software Assurance Professional Certificate program.[16] The tutorial provides a general overview of the SERA Method, which focuses on SERA principles and concepts; it does not teach participants how to conduct the method. SERA Method training for practitioners would focus on how to execute the method's four tasks, providing detailed step-by-step guidance for prospective Analysis Team members.

This list highlights several candidate next steps that will support the transition of the SERA Method. We plan to evaluate each candidate and determine which, if any, should be pursued in future phases of SERA Method development.

[16] Visit the SEI website for information about this tutorial: https://www.sei.cmu.edu/education-outreach/courses/course.cfm?coursecode=V46.

Appendix: DFS Example Details

This appendix documents the following data for the DSF cyber-risk scenario featured in this report:

- *Cyber-Risk Scenario Narrative*—textual description of the cyber-risk scenario for the DFS and its associated mission thread
- *Threat Sequence Table*—the series of actions taken by the actor(s) when executing the threat (This table [Table 6] also includes enablers of each threat action and candidate controls.)
- *Mission Consequences Table*—the effects of a threat on the mission thread (This table [Table 7] also includes consequence amplifiers and candidate controls.)

Cyber-Risk Scenario Narrative

The following narrative describes the cyber-risk scenario for the DFS and its associated mission thread:

> *An insider with technical skills and administrative access to the Data Fusion System (DFS) becomes disgruntled after being passed over for a promotion and not receiving a bonus. The insider begins to behave aggressively and abusively toward co-workers. After a while, the insider decides to execute a cyber attack on the DFS. The insider's goal is to execute a denial-of-service (DoS) attack on DFS switches.*
>
> *The insider uses cyber access to the DFS engineering repository (resulting from insufficient access control mechanisms) to view engineering documents. The insider uses physical access to the DFS engineering organization's work space to view unsecured hard copies of DFS engineering documents. The insider develops a plan for the cyber attack based on the available information. The insider develops malicious code designed to flood the DFS network with traffic. The insider transfers the malicious code to the designated DFS laptop, connects the laptop to a network-access port on the DFS, uploads the malicious code to the DFS, and changes log files to erase evidence of the action.*
>
> *After the DFS begins its mission, the malicious code monitors DFS network traffic. When the data indicate that the DFS is receiving mission data, the malicious code's attack is triggered. The malicious code floods the DFS network with illegitimate traffic. Processing illegitimate requests consumes the DFS switches' resources, which creates a DFS DoS. The DFS cannot perform its function, and the mission that it supports cannot be completed. As a result, the government suspends the future use of the DFS pending the execution of the disaster recovery plan.*

Threat Sequence Table

Table 6: Threat Sequence Table for DFS Cyber-Risk Scenario

Step	Enabler	Candidate Control	NIST Mapping
1. An insider with technical skills and administrative access to the Data Fusion System (DFS) becomes disgruntled after being passed over for a promotion and not receiving a bonus.	Insufficient feedback about employee performance	The organization's managers provide constructive feedback on performance issues.	NIST CSF: PR.IP-11 NIST 800-53: PS-1. PS-2, PS-3, PS-4, PS-5, PS-6, PS-7, PS-8, SA-21
2. The insider begins to behave aggressively and abusively toward co-workers.	Tolerance for inappropriate employee behavior	The organization's managers recognize inappropriate behavior when it occurs and respond appropriately.	NIST CSF: PR.IP-11 NIST 800-53: PS-1. PS-2, PS-3, PS-4, PS-5, PS-6, PS-7, PS-8, SA-21
3. After a while, the insider decides to execute a cyber attack on the DFS. The insider's goal is to execute a denial-of-service (DoS) attack on DFS switches.	No resolution to underlying employee issue	The organization's managers recognize an employee's escalating frustration and proactively work to defuse the situation.	NIST CSF: PR.IP-11 NIST 800-53: PS-1. PS-2, PS-3, PS-4, PS-5, PS-6, PS-7, PS-8, SA-21
4. The insider uses cyber access to the DFS engineering repository (resulting from insufficient access control mechanisms) to view engineering documents. The insider uses physical access to the DFS engineering organization's work space to view unsecured hard copies of DFS engineering documents.	Insufficient access control for information and resources (physical and cyber)	The organization manages and protects physical access to its information and resources.	NIST CSF: PR.AC-2 NIST 800-53: PE-2, PE-3, PE-4, PE-5, PE-6, PE-8
		The organization manages access permissions and authorizations for computing resources.	NIST CSF: PR-AC-4 NIST 800-53: AC-1, AC-2, AC-3, AC-5, AC-6, AC-14, AC-16, AC-24
	Insufficient monitoring of the organizational environment for abnormal activity (physical and cyber)	The organization monitors the physical environment for abnormal activity.	NIST CSF: DE.CM-2 NIST 800-53: CA-7, PE-3, PE-6, PE-20
		The organization monitors systems and networks for abnormal activity.	NIST CSF: DE.CM-1 NIST 800-53: AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
		The organization performs targeted monitoring of individuals with suspected behavioral issues.	NIST CSF: DE.CM-3 NIST 800-53: AC-2, AU-12, AU-13, CA-7, CM-10, CM-11
		The organization responds appropriately when abnormal activity is detected.	NIST CSF: RS.MI-1, RS-MI-2 NIST 800-53: IR-4

Step	Enabler	Candidate Control	NIST Mapping
5. The insider develops a plan for the cyber attack based on the available information.	Technical knowledge, skills, and abilities required to interpret technical information about the system and design a cyber attack	---	NIST CSF: --- NIST 800-53: ---
6. The insider uses the organization’s resources to develop malicious code designed to flood the DFS network with traffic.	Technical knowledge, skills, and abilities required to inflict damage on systems and networks	---	NIST CSF: --- NIST 800-53: ---
	Ability to use organizational computing resources in illicit activities	The organization performs targeted monitoring of individuals with suspected behavioral issues.	NIST CSF: DE.CM-3 NIST 800-53: AC-2, AU-12, AU-13, CA-7, CM-10, CM-11
		The organization monitors systems and networks for abnormal activity.	NIST CSF: DE.CM-1 NIST 800-53: AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
		The organization responds appropriately when abnormal activity is detected.	NIST CSF: RS.RP-1, RS.MI-1, RS-MI-2 NIST 800-53: CP-2, CP-10, IR-4, IR-8
7. The insider transfers the malicious code to the designated DFS laptop, connects the laptop to a network access port on the DFS, uploads the malicious code to the DFS, and changes log files to erase evidence of the action.	Insufficient access control and monitoring of physical assets (i.e., DFS laptop)	The organization manages and protects physical access to its information and resources.	NIST CSF: PR.AC-2 NIST 800-53: PE-2, PE-3, PE-4, PE-5, PE-6, PE-8
		The organization monitors the physical environment for abnormal activity.	NIST CSF: DE.CM-2 NIST 800-53: CA-7, PE-3, PE-6, PE-20
	Insufficient organizational change management/configuration management capability	The organization implements a change management/configuration management system to track changes to the code base.	NIST CSF: PR-IP-3 NIST 800-53: CM-3, CM-4, SA-10
		The organization analyses its code base for the presence of malicious code.	NIST CSF: DE.CM-4 NIST 800-53: SI-3, SI-8
		The organization responds appropriately when abnormal activity is detected.	NIST CSF: RS.RP-1, RS.MI-1, RS-MI-2 NIST 800-53: CP-2, CP-10, IR-4, IR-8
	Insufficient monitoring of the organization’s systems and	The organization monitors systems and networks for abnormal activity.	NIST CSF: DE.CM-1 NIST 800-53: AC-2, AU-12, CA-7, |CM-3, SC-5, SC-7, SI-4

Step	Enabler	Candidate Control	NIST Mapping
	networks for abnormal activity during system maintenance	The organization performs targeted monitoring of individuals with suspected behavioral issues.	NIST CSF: DE.CM-3 NIST 800-53: AC-2, AU-12, AU-13, CA-7, CM-10, CM-11
		The organization responds appropriately when abnormal activity is detected.	NIST CSF: RS.RP-1, RS.MI-1, |RS-MI-2 NIST 800-53: CP-2, CP-10, IR-4, IR-8
8. After the DFS begins its mission, the malicious code monitors DFS network traffic.	Insufficient monitoring of the DFS for abnormal activity during the mission	Mission personnel monitor the system for abnormal activity during the mission.	NIST CSF: DE.CM-1 NIST 800-53: AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
		Mission personnel respond appropriately when abnormal system activity is detected.	NIST CSF: RS.RP-1, RS.MI-1, RS-MI-2 NIST 800-53: CP-2, CP-10, IR-4, IR-8
	Mission personnel lack the technical skills needed to perform DFS monitoring activities	The organization provides role-based security training to designated mission personnel.	NIST CSF: PR.AT-2 NIST 800-53: AT-3, PM-13
9. When the data indicate that the DFS is receiving mission data, the malicious code's attack is triggered. The malicious code floods the DFS network with illegitimate traffic. Processing illegitimate requests consumes the DFS switches' resources, which creates a DFS DoS.	Mission personnel not prepared to respond to a DFS DoS attack	The organization provides role-based security training to designated mission personnel.	NIST CSF: PR.AT-2 NIST 800-53: AT-3, PM-13
	Lack of a response plan during mission execution	Mission personnel respond appropriately when abnormal system activity is detected.	NIST CSF: RS.RP-1, RS.MI-1, RS-MI-2 NIST 800-53: CP-2, CP-10, IR-4, IR-8
	DFS system architecture vulnerable to a DoS attack	The organization ensures that the system has adequate capacity to maintain its availability.	NIST CSF: PR.DS-4 NIST 800-53: AU-4, CP-2, SC-5

Mission Consequence Table

Table 7: Mission Consequence Table for DFS Cyber-Risk Scenario

Consequence	Amplifier	Candidate Control	NIST Mapping
The mission supported by the DFS cannot be completed.	Mission dependency on a single system (i.e., a single point of failure)	The system has mechanisms (e.g., hot backup system) to achieve mission resilience in normal and adverse situations.	NIST CSF: PR.PT-5 NIST 800-53: CP-7, CP-8, CP-11, CP-13, PL-8, SA-14, SC-6
The government suspends future use of the DFS pending the execution of the disaster recovery plan.	Insufficient disaster recovery planning slows recovery activities and amplifies recovery time and cost	The organization executes a recovery plan after a cybersecurity incident occurs.	NIST CSF: RC.RP-1 NIST 800-53: CP-10, IR-4, IR-8
	Organizational personnel not prepared for recovery activities, slowing recovery activities and amplifying recovery time and cost	The organization provides role-based security training to the disaster recovery team.	NIST CSF: PR.AT-2 NIST 800-53: AT-3, PM-13

References

[Alberts 2005]

Alberts, Christopher & Dorofee, Audrey. *Mission Assurance Analysis Protocol (MAAP): Assessing Risk in Complex Environments*. CMU/SEI-2005-TN-032. Software Engineering Institute, Carnegie Mellon University. 2005. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=7505

[Alberts 2016]

Alberts, Christopher; Dorofee, Audrey; & Woody, Carol. *Wireless Emergency Alerts Commercial Mobile Service Provider (CMSP) Cybersecurity Guidelines*. CMU/SEI-2016-SR-009. Software Engineering Institute, Carnegie Mellon University. 2016. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=463988

[DoD 2018]

Office of the Secretary of Defense for Policy. *Mission Assurance (MA)*. DoD Directive 3020.40. Washington, DC. 2018. https://fas.org/irp/doddir/dod/d3020_40.pdf

[Dorofee 1996]

Dorofee, A.; Walker, J.; Alberts, C.; Higuera, R.; Murphy, R.; & Williams, R. *Continuous Risk Management Guidebook*. Software Engineering Institute, Carnegie Mellon University. 1996. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=30856

[Merriam-Webster 2018]

Merriam-Webster. *Merriam-Webster.com*. 2018 [accessed December 2018]. https://www.merriam-webster.com.

[MITRE 2020a]

The MITRE Corporation. *Common Attack Pattern Enumeration and Classification: A Community Resource for Identifying and Understanding Attacks*, Bedford, MA. 2020. https://capec.mitre.org/index.html

[MITRE 2020b]

The MITRE Corporation. *About CAPEC*. Bedford, MA. 2020. https://capec.mitre.org/about/glossary.html

[MITRE 2020c]

The MITRE Corporation. *CAPEC-549: Local Execution of Code*. Bedford, MA. 2020. https://capec.mitre.org/data/definitions/549.html

[MITRE 2020d]

The MITRE Corporation. *CAPEC-125: Flooding*. Bedford, MA. 2020. https://capec.mitre.org/data/definitions/125.html

[NASA 2009]

National Aeronautics and Space Administration (NASA). *Final Report, NASA Study on Flight Software Complexity*. NASA Jet Propulsion Laboratory, Systems and Software Division. Pasadena, CA. 2009. http://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf

[NIA 2010]

Committee on National Security Systems. *National Information Assurance (IA) Glossary CNSS Instruction*. CNSS Instruction No. 4009. Fort George G. Meade, MD. 2010. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[NIST 2013]

National Institute of Standards and Technology. *Security and Privacy Controls for Federal Information Systems and Organizations*. NIST Special Publication 800-53 Revision 4. National Institute of Standards and Technology. 2013. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[NIST 2018]

National Institute of Standards and Technology. *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*. Gaithersburg, MD. National Institute of Standards and Technology. 2018. https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11

[Novak 2010]

Novak, William E. & Levine, Linda. *Success in Acquisition: Using Archetypes to Beat the Odds* CMU/SEI-2010-TR-016. Software Engineering Institute, Carnegie Mellon University. 2010. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9519

[Senge 1991]

Senge, Peter. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Doubleday. 1991. ISBN: 0385260954. https://www.penguinrandomhouse.com/books/163984/the-fifth-discipline-by-peter-m-senge/9780385517256