REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01/01/2022	Technical Report	

4. TITLE AND SUBTITLE

Modeling Security Views with Unified Architecture Framework, Risk Assessment and Analysis Modeling Language, and Systems Modeling Language

5a. CONTRACT NUMBER
FA8702-22-C-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dandashi, Fatma;

5d. PROJECT NUMBER
100933.10.100.3JG0.DE0

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102

8. PERFORMING ORGANIZATION REPORT NUMBER
PRS-22-0326

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Cyber Mission Forces (CMF)

10. SPONSOR/MONITOR'S ACRONYM(S)
DoD CMF

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
Submission to the Object Management Group

14. ABSTRACT

This paper describes an approach to define and model Security Views by applying a Model-Based Systems Engineering (MBSE) approach using the Unified Architecture Framework (UAF), Risk Assessment and Analysis Modeling Language (RAAML), and the Systems Modeling Language (SysML). A summary description on the use of an analytical modeling tool for cyber resiliency analysis with traceability to the architecture model is also included. A sample model for civilian maritime Search and Rescue (SAR) is used to illustrate how Security Views are developed to integrate cyber resiliency analysis res

15. SUBJECT TERMS

Maritime Operations; Computer Security; Emergency Preparedness and Response; Risk Management; security; RAAML; search and rescue; cyber resiliency; risk assessment; Risk Assessment and Analysis Modeling Language; model-based sy

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		32	Susan Carpenito
					19b. TELEPHONE NUMBER *(Include area code)* 781-271-7646

MITRE

MTR220019
MITRE TECHNICAL REPORT

Modeling Security Views with Unified Architecture Framework, Risk Assessment and Analysis Modeling Language, and Systems Modeling Language

Fatma Dandashi

January 2022

McLean, VA

Abstract

This paper describes an approach to define and model Security Views by applying a Model-Based Systems Engineering (MBSE) approach using the Unified Architecture Framework (UAF), Risk Assessment and Analysis Modeling Language (RAAML), and the Systems Modeling Language (SysML). A summary description on the use of an analytical modeling tool for cyber resiliency analysis with traceability to the architecture model is also included. A sample model for civilian maritime Search and Rescue (SAR) is used to illustrate how Security Views are developed to integrate cyber resiliency analysis results within an architecture model.

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

1 Introduction

Model-Based Systems Engineering (MBSE) is used to support the modeling of requirements, design, analysis, verification, and validation associated with the development of complex systems. In contrast to document-centric engineering, MBSE puts models at the center of system design. Several types of models are encompassed within MBSE. Two that are relevant to this paper are introduced here.

Architecture models emphasize how components fit together into a consistent whole and are repository-based to capture structure, behavior, and information flows. Architecture models are a vehicle to integrate analysis products.

Analytical models emphasize specific aspects of performance, are mathematically based in computation or simulation, and can be applied in the validation and optimization of assessments. Analytical models are a vehicle to solve some computational problem.

In this paper, we use a sample architecture model to explain how to model cyber resiliency.[1] The model is described using the Unified Architecture Framework (UAF) [1], Risk Assessment and Analysis Modeling Language (RAAML) [2], and the Systems Modeling Language (SysML) [2]. Analytical tools that are geared towards cyber resiliency analysis also exist 19. Analytical tools provide consistent and repeatable assessments (vs. human based assessments) thereby improving resilience in design, and support confidence in rapid delivery of capabilities. Figure 1 illustrates how an architecture modeling tool and an analytical modeling tool may be chained together to maintain traceability from the systems architecture to resiliency analysis results and back to the capabilities, operational mission, requirements as described in the architecture model. One example of an analytical result may be the identification of systems with potential vulnerabilities, and a list of Security Controls (SCs)[2] to mitigate those vulnerabilities. Traceability from the identified systems and vulnerability mitigations back to the operational mission within the architecture model can only lead to an improved efficiency in securing capabilities.

[1] National Institute of Standards and Technology (NIST) Special Publication (SP) 800-160 Vol 2 defines resilience in the cyber domain, or Cyber Resilience as "The ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on system resources." [4]

[2] See NIST SP 800-53 Rev. 4 (April 2013 with updates through 1/15/14), Security and Privacy Controls for Federal Information Systems and Organizations" [5]

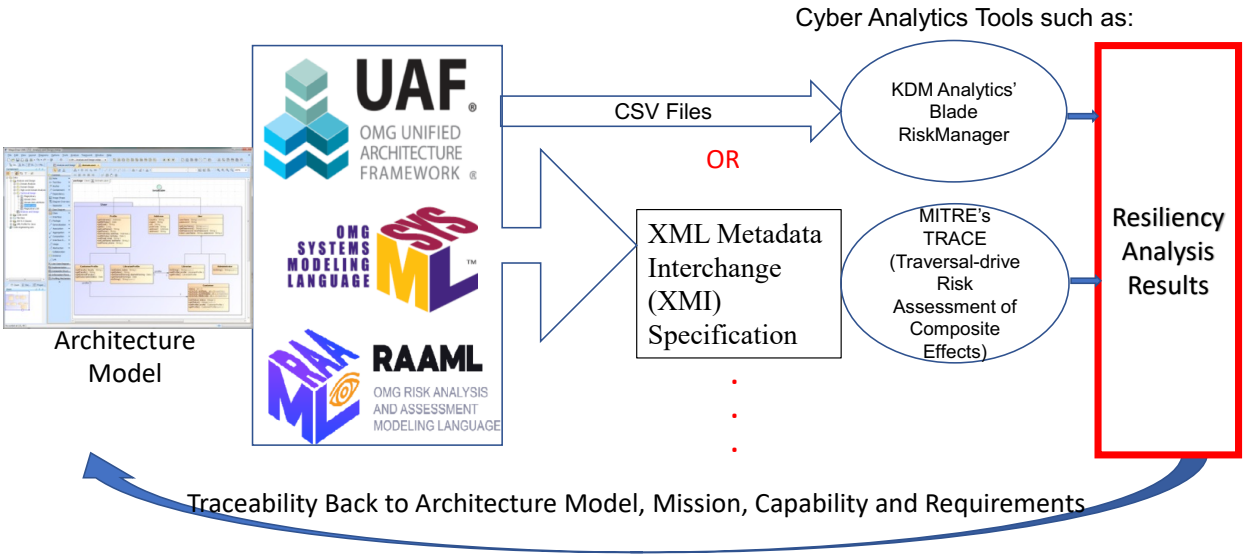

Figure 1: Architecture To Resiliency Analysis

For this paper, an analytical model was developed using MITRE's Traversal-drive Risk Assessment of Composite Effects (TRACE) [6], but TRACE use is not detailed in this paper. Results produced by TRACE are applied to the architecture model and the resulting Security Views are illustrated in later sections of this paper. In describing the sample model, the reader is assumed to have knowledge of UAF and SysML. A subset of RAAML features utilized in this paper are briefly explained. TRACE is a MITRE-developed analytical tool, referenced in this paper, but is not the subject of the approach explained here.

2 Sample Model

The problem domain for the sample model is civilian maritime Search and Rescue (SAR). Civilian SAR was selected for several reasons:

- SAR is an easily understood problem domain with easy-to-recognize typical scenarios.
- SAR mission and operations resemble other Department of Defense (DoD) missions but remain unclassified.
- The domain is sufficiently large and complex involving mixed human, software, and hardware solutions. As such, it includes enough complexity to enable simple, yet adequate cyber resiliency modeling.
- The sample model applies UAF to a common scenario in civilian maritime SAR operations—e.g., a yacht in distress. A monitoring unit picks up the distress signal from the yacht and passes it on to the Command and Control (C2) Center. The C2 Center coordinates SAR operations among helicopters, a naval ship, and a civilian voluntary sea rescue organization.

Caveats:

Not all views and elements normally included in a SAR architecture model are shown in the sample model used for this paper, The content selected for inclusion is that necessary to accomplish the stated goal. This allows us to communicate the use of UAF to model cyber resiliency, without the need for too much architecture detail. Consequently, there will be errors/omissions in the specifics of the SAR domain contained within this model.

1. This sample illustrates the use of UAF for modeling cyber resiliency. There is no substitute for using analytical models to conduct rigorous cyber resiliency analysis. An architecture model developed using UAF serves as the integrator that provides a digital reference to all other models needed during the complex design and analysis of systems.
2. Since this sample deals with cyber resiliency, the paper only references hardware and software solutions used within a SAR model. Other security analyses such as those for physical and personnel security are not included in this example, but the same principles apply.

Figure 2 is a hybrid diagram intended to provide context for the work detailed in this paper. It shows capability, operational, as well as Resource View elements in a single diagram.

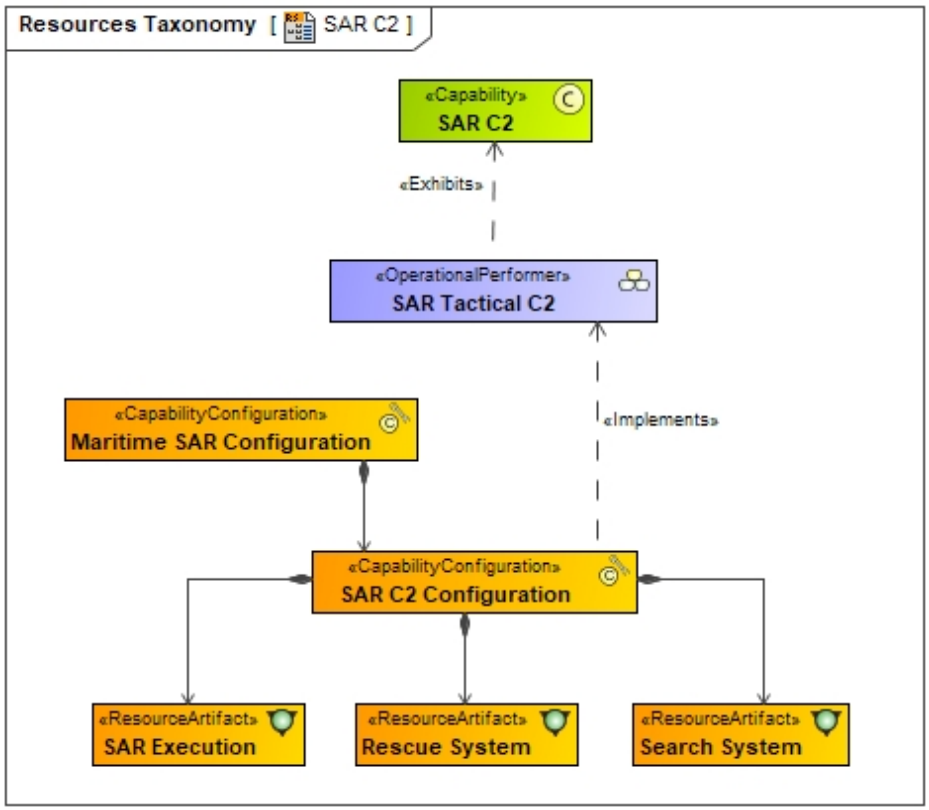

Figure 2: Scope Of System Elements

To conduct cyber resiliency analysis, we augment UAF with RAAML and model a fault tree (FT). The model elements defined for resiliency analysis are created in their own RAAML views (external to UAF). This is an example of the extensibility features provided by UAF and SysML enabling the easy creation of fit for purpose views. These elements were created using a SysML package stored within the Parameters View.

A “Cyber Context Block” (see Figure 3) is defined in this model to provide the context for the FT. Elements that are defined as parts for this block (and consecutively their parts) compose the scope for this analysis. Figure 3 shows the elements from the Resources View that are included in this analysis. Some common elements such as “Switch” and “GPS” are defined once, but instances of these are used by several resources as needed.

Figure 3 shows the elements from the SAR Resource Views that were used in this sample, and is only to be viewed as an example, not a comprehensive SAR architecture model, nor a comprehensive cyber resiliency modeling and analysis exercise. An internal block diagram (ibd) diagram is associated with the “Cyber Context Block” element and shows the SAR mission systems and their connectivity. It will be described later in this paper. It is provided in this section for clarity (see Figure 4 vs Figure 3).

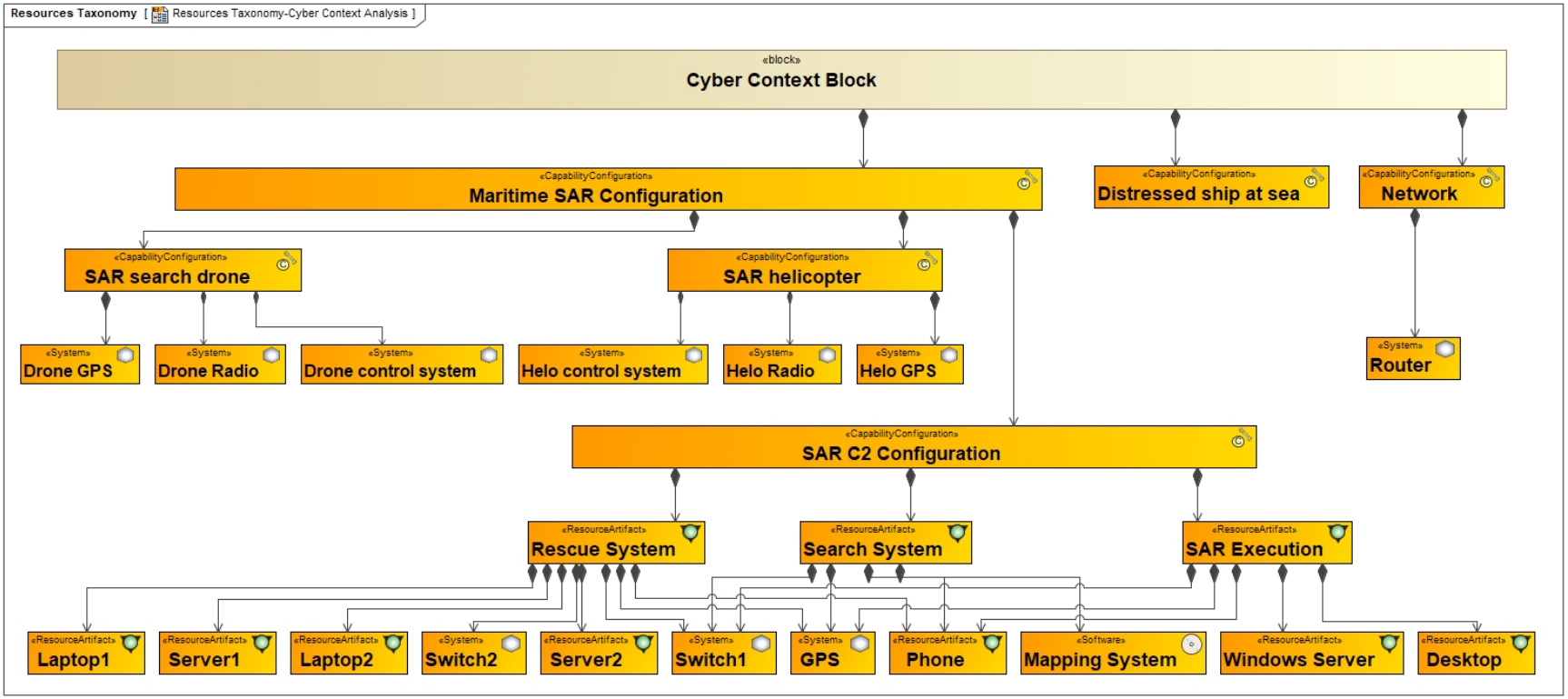

Figure 3: The Systems, Hardware, And Software Components For Cyber Resiliency Analysis

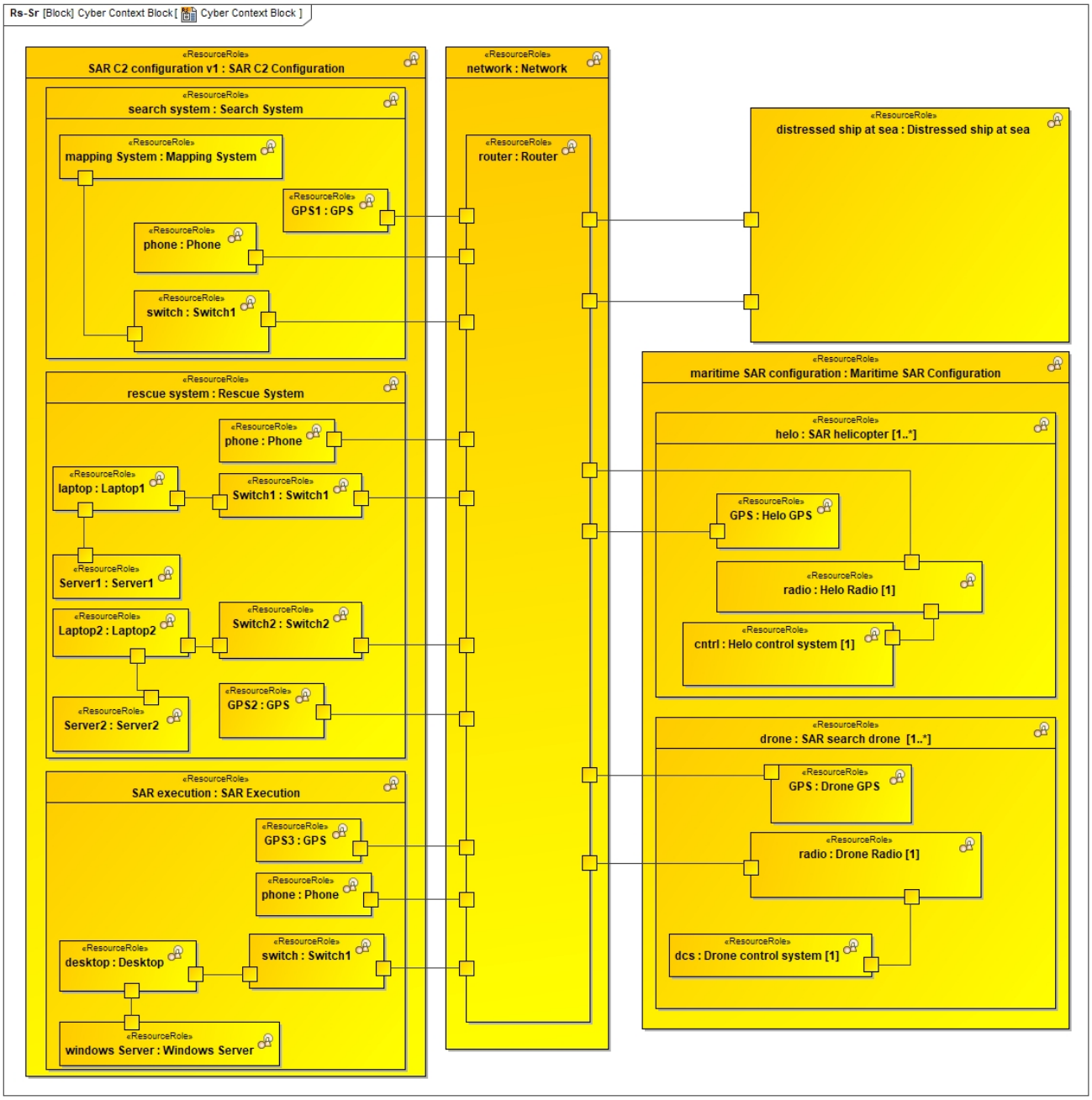

Figure 4: Resources Network Diagram For Cyber Resiliency Analysis

3 Cyber Resiliency Modeling Using RAAML With Traceability To A UAF Model

To augment UAF for modeling cyber resiliency, we use UAF Parameters Views to create a FT using the RAAML specification. We create a package to store RAAML elements and diagrams. Several RAAML stereotypes are used to define the FT. For modeling FT events, the RAAML library defines the Event element. There are three types of events defined in RAAML, which are *TopEvent, IntermediateEvent, BasicEvent*. Further, RAAML defines *AND, OR,* and other Logic gates to construct the FT. RAAML also defines associated relationships to tie the event-type elements to the architecture model elements where they apply. *BasicEvents* are commonly positioned at the leaves of a FT. *IntermediateEvents* behave like the Logic gates and provide inputs to upper-level gates to help construct the FT. There is only one top-level event allowed per FT (see Figure 8).

RAAML defines connections between FT elements and system model elements using the *RelevantTo* relationship. *RelevantTo* expresses the statement that the connected element is about the particular system part. The *RelevantTo* relationship client end is an FT *Event*. It can also be the *FTATree*—a block, inherited from *FTATree* RAAML library class. The *RelevantTo*[3] relationship supplier end is the system model item. It can be either a block or a property of the block.

The structure and connectivity of the FT is constructed through analysis of the systems, system functions, and potential system failures (outside the scope of this paper). Based on such an analysis, the FT is created by identifying the events that can lead to each undesired system behavior which may lead to a system failure. Based on the structure of system resources identified in Figure 3, the system components that are vulnerable are identified in Figure 5. In this sample model, the leaf level model elements in Figure 3 are identified as the first point of attack from an external (internet) connection.

Consequently, these system components are linked to *BasicEvents* (see Figure 5) that are created in this diagram to correspond to failure event types that may be associated with each type of system resource. We use the *RelevantTo* relationship to create the links between system resource elements and event to be used in the construction of the FT.

[3] *RelevantTo* applies to other RAAML types that are not mentioned in this paper.

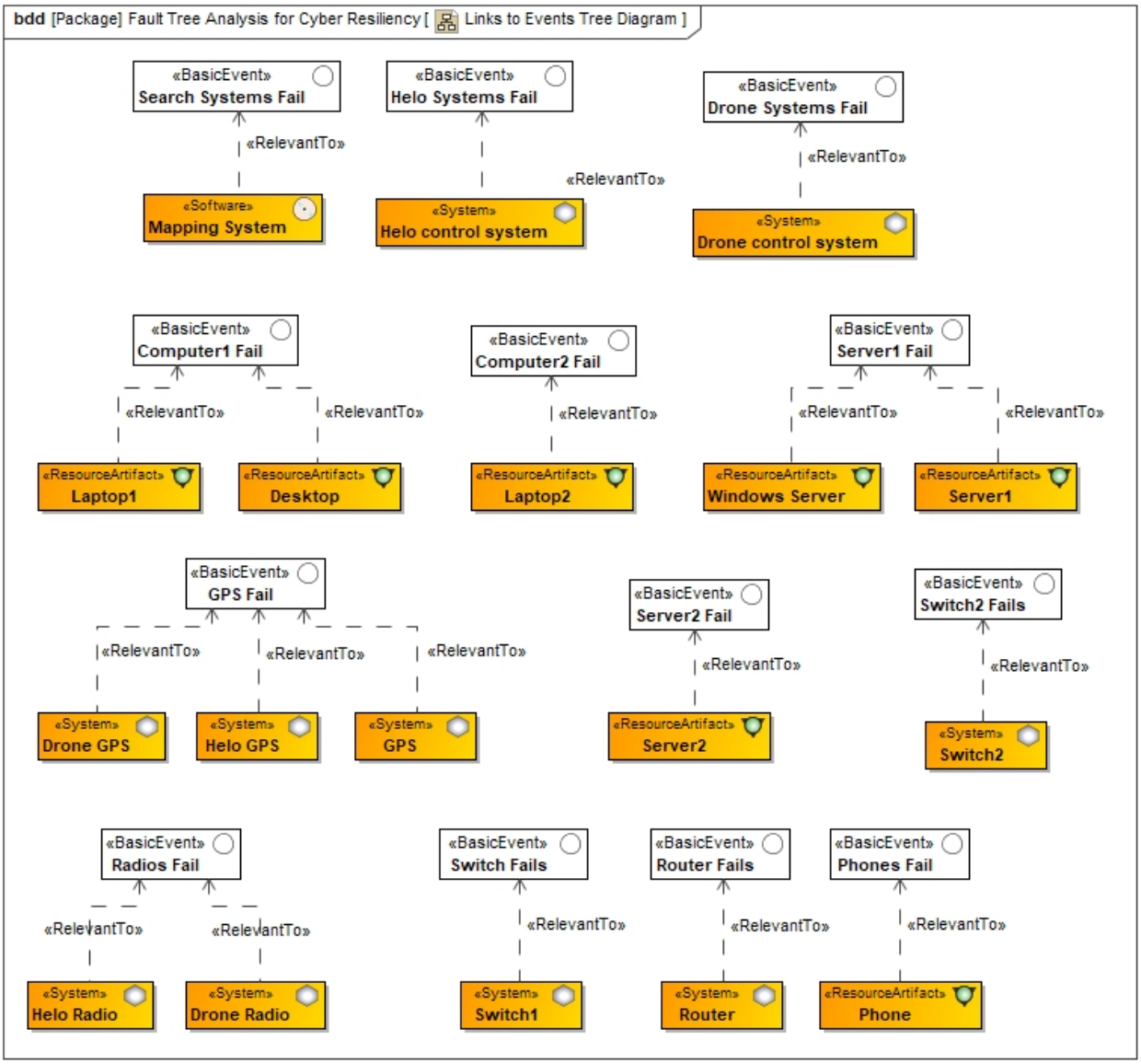

Figure 5: System Components Identified As First Contact Nodes

Other event types are shown in Figure 6. We create a context block (of type *FTATree*) for the FT called "Boat SAR Mission FT". An ibd is associated with the "Boat SAR Mission FT" element and will be described later (see Figure 8). Figure 6 shows the *TopEvent* which is called "Mission Fails". The intermediate events may not be clear to the reader at this point, but they are identified as the FT is constructed. The development of this sample model was iterative and the sequence of diagrams as they appear in this paper does not reflect the order in which the model was finalized. Several iterations are needed to finalize the events, their relations to model elements and the construction of the FT.

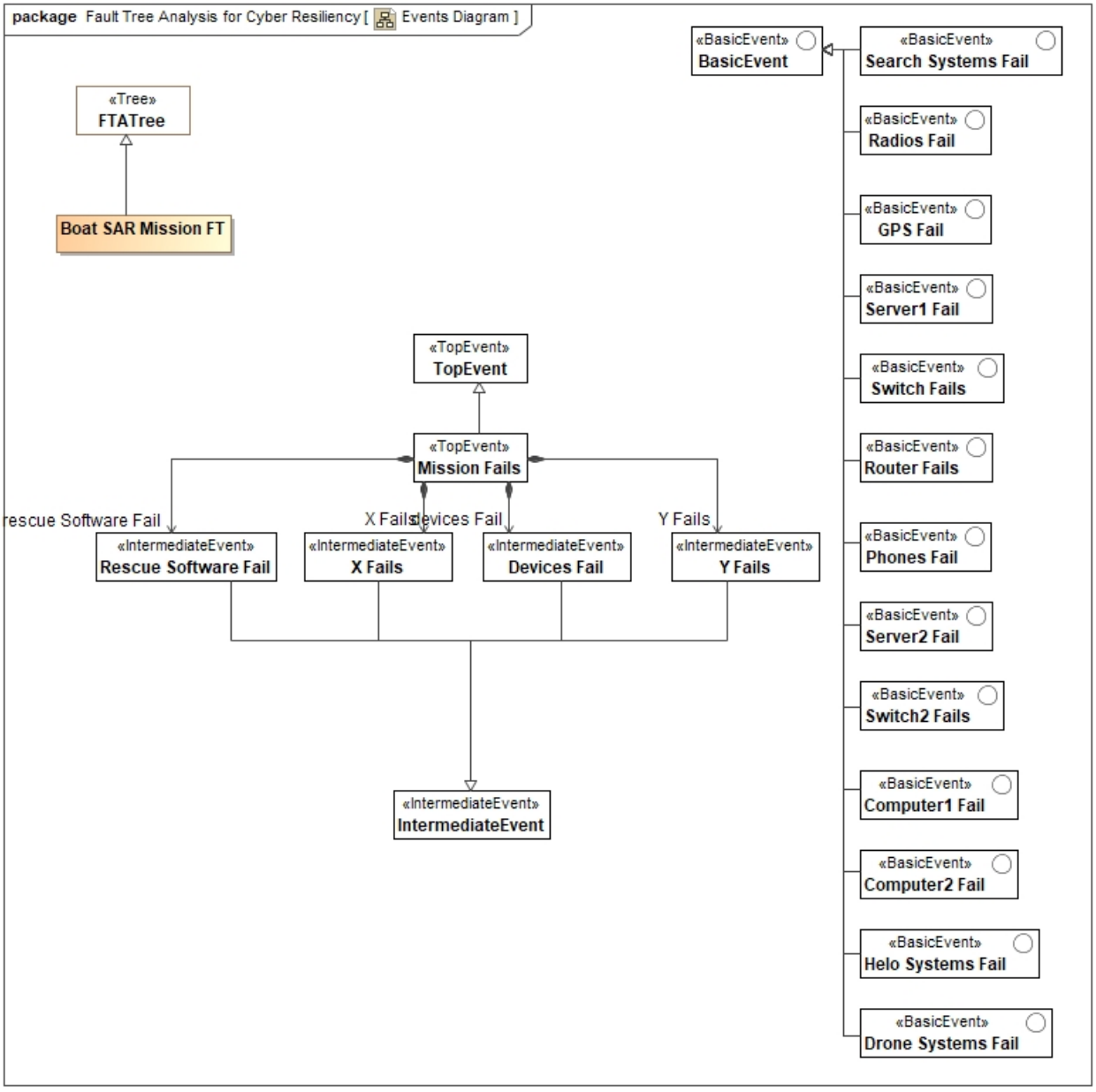

Figure 6: Events That Form The FT

The "Cyber Context Block" (see Figure 3) is used to generate the resource network diagram. This UAF Resource Structure (Rs-Sr) ibd (see Figure 7) details the system elements and their structure, as well as the communications links (which are created in this diagram) that might lead to cyber vulnerabilities originating externally (for simplicity, we do not consider internal cyber vulnerabilities in this sample model). The links from the basic events to the system elements were defined in Figure 5.

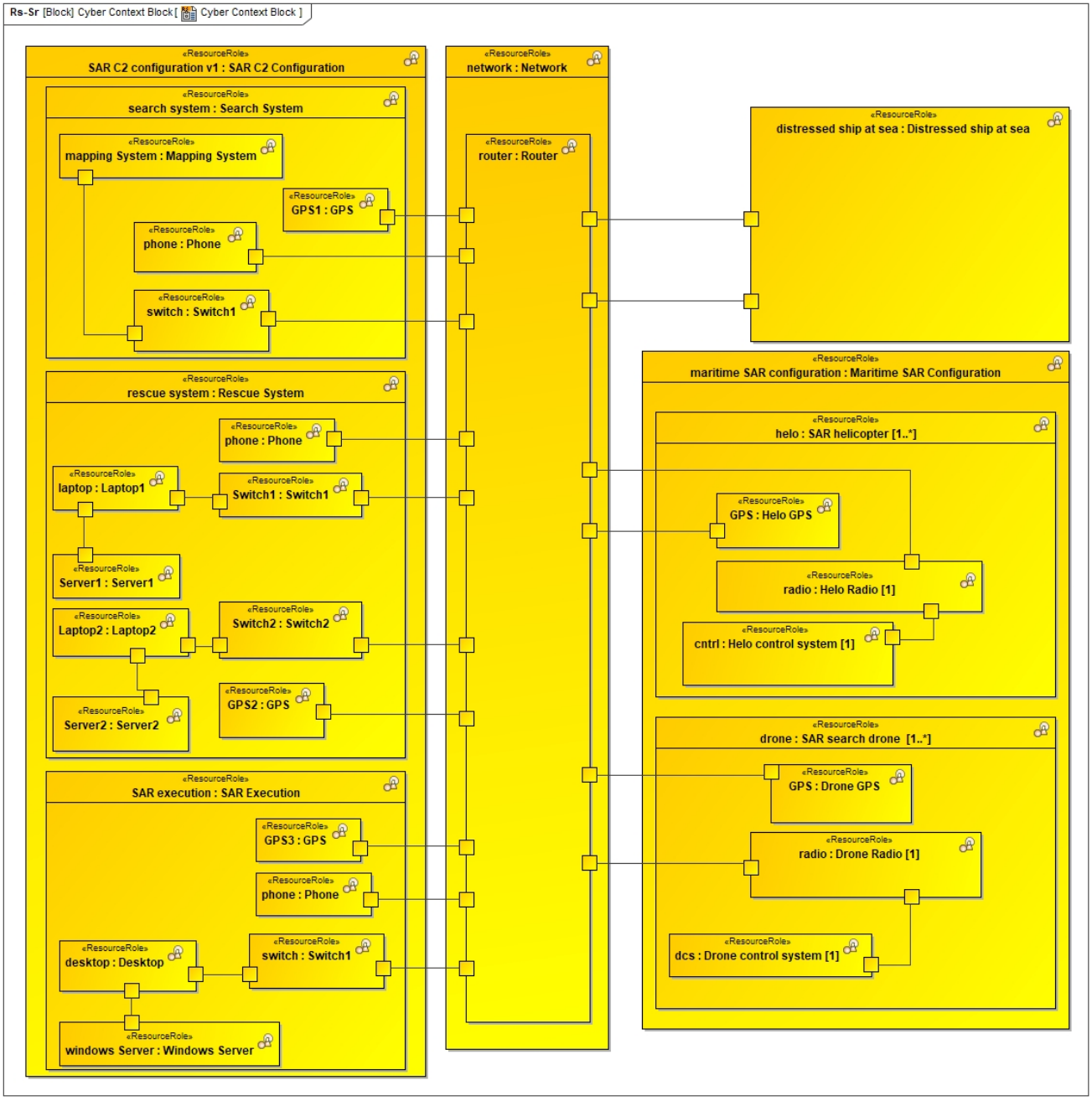

Figure 7: Resources Network Diagram For Cyber Resiliency Analysis

Next, an assessment of the systems is carried out to determine the combinations of events that could occur for the system to fail. The element "Boat SAR Mission FT" is a context block defined to create the SysML ibd for the FT. The FT is modeled using this ibd. Part properties in that diagram will be typed by elements defined in the RAAML library and thus will show part type icons. Figure 8 shows the FT that was created based on the resource connectivity analysis (see Figure 7). The *TopEvent* element is identified as "Mission Fails". The FT diagram defines the flow of failures from basic events leading to mission failure. Logic gates and intermediate events are used to define the FT starting with failures defined as basic events which combine to lead to mission failure.

To create this FT, we start the analysis by adding all event elements to the diagram. Then, based on the connectivity in Figure 7, we combine events using AND/OR logic gates depending on how one element (connected to the event under consideration) might affect the failure of the mission. For example, if a server fails in *either* of the SAR Rescue or SAR Execution systems, then the mission will fail, since the operators will not have access to critical information needed to conduct the mission. Further, due to the way the server is connected through a switch to the rest of the systems, a server, (or also a laptop), OR a switch failing will also lead to mission failure, we use AND logic gate for these three events. There is an OR gate between Laptop1, Server1, and Switch1 Group, and the Laptop2, Server2, and Switch2[4] Group, since they provide redundancy in the way they are networked in the Rescue System. Any other systems failing independently will also lead to mission failure, so each is OR-ed with all other events, leading to mission failure event. This approach to constructing the FT is not prescriptive and is provided here for illustrative purposes only. Generally, models are much more complex than the paired-down example provided in this paper. Therefore, FTs are more likely to be auto-generate using fault analysis tools.

[4] In this example, we assume that Laptop 2, Server2, and Switch2 are comprised of a different set of Hardware/software and configuration than the set Laptop1, Server1, and Switch1 to provide true redundancy.

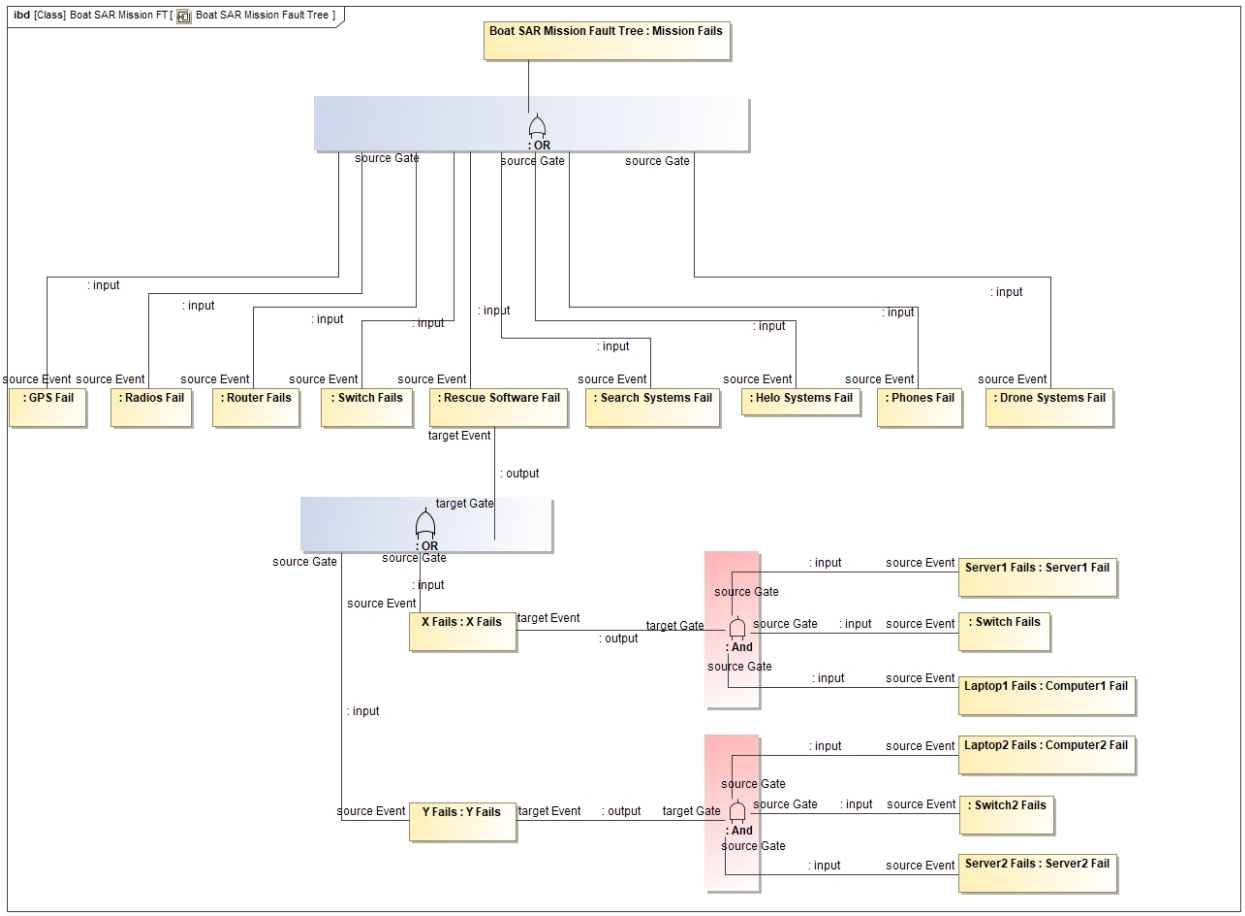

Figure 8: Fault Tree For Cyber Resiliency Analysis

4 Cyber Resiliency Analysis With Analytical tools

Once an FT is constructed, various analyses can be run (e.g., using a cyber resiliency analytical tool that can ingest the architecture model elements). Such an analysis tool will reveal specific risks and vulnerabilities, may identify applicable SCs, and may possibly define mitigations (see resource mitigation elements and associated security resource elements defined in UAF) to defend against the identified cyber vulnerabilities. Three such tools are introduced here to provide an overview of potential analytical tools by way of example. Other tools exist that may perform similar analyses. In addition to the analytical tool's algorithms and analysis results produced, the input format that each tool uses to ingest UAF model data is important, should be a standard format, and is therefore noted in the information on each tool below.

4.1 KDM Analytics: Blade RiskManager (BRM)

Blade RiskManager (BRM) [7] generates a risk assessment based on the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) and shows where to focus risk assessment resources. BRM provides a top-down operational view of risk. This enables organizations to identify, prioritize, and focus security assessment and risk mitigation to the most critical and risky components of a system. It ingests Department of Defense

Architecture Framework (DoDAF) architecture models in the form of Object Management Group's (OMG's) Extensible Markup Language (XML) Metadata Interchange (XMI®) [8].

4.2 MITRE's TRACE

TRACE is a tool developed by MITRE for a government sponsor. It considers likelihood and impact to mission, and ties asset failures to mission impact using FT analysis. The tool uses sources such as MITRE's ATT&CK® and overlays probability data from a Threat Concept Database and conducts Monte Carlo analysis to identify vulnerabilities and provides a list of SCs as output [6]. TRACE ingests OMG's XMI® [8].

5 Sample Model Analysis Results

In this paper, we used TRACE to conduct the cyber resiliency analysis. Using the SAR sample model, we utilized TRACE to address the influence of the stochastic space (such as the network architecture in a cyber offensive process). While TRACE also analyzes the deterministic space (the functional or mission dependency model), this sample model does not cover that scope. After TRACE is run, the analysis returns a report that includes a table listing system components and the corresponding SCs listed as requirements. This process is iterative. That is, once an initial analysis is run, the results will provide insights on where vulnerabilities exist, and what to do to *secure them* via SCs. Further, analysis results may lead to a redesign of the hardware/software configuration and connectivity (to be reflected in a new version of the architecture model, new FTs, and new analyses to be run). That is, the analysis results will guide the redesign on how to make the systems resilient by incorporating alternative or more robust systems and connections such that the mission can proceed (and not fail) albeit at a reduced level of efficiency (or speed, etc.).

Table 1 is a subset of a TRACE report that shows a list of SCs identified by TRACE for the "Switch" element only. Table 2 shows a list of SCs identified by TRACE for the "Router" element.

Table 1. SCs Identified By TRACE To Address Cyber Vulnerabilities For The Switch Element

Element	Mitigations
Switch	• Requirement SI-23 Information Fragmentation • Requirement SC-16 TRANSMISSION OF SECURITY AND PRIVACY ATTRIBUTES • Requirement SC-36 DISTRIBUTED PROCESSING AND STORAGE • Requirement SC-7(5) BOUNDARY PROTECTION | DENY BY DEFAULT / ALLOW BY EXCEPTION • Requirement SC-7(8) BOUNDARY PROTECTION | ROUTE TRAFFIC TO AUTHENTICATED PROXY SERVERS • Requirement SC-7(18) BOUNDARY PROTECTION | FAIL SECURE • Requirement SC-8(1) TRANSMISSION CONFIDENTIALITY AND INTEGRITY | CRYPTOGRAPHIC OR ALTERNATE PHYSICAL PROTECTION

Table 2: SCs Identified By TRACE To Address Cyber Vulnerabilities For The Router Element

Element	Mitigations
Router(19)	• Requirement SC-36 DISTRIBUTED PROCESSING AND STORAGE • Requirement SC-7 BOUNDARY PROTECTION • Requirement SC-7(5) BOUNDARY PROTECTION | DENY BY DEFAULT / ALLOW BY EXCEPTION • Requirement SC-7(7) BOUNDARY PROTECTION | PREVENT SPLIT TUNNELING FOR REMOTE DEVICES • Requirement SC-7(8) BOUNDARY PROTECTION | ROUTE TRAFFIC TO AUTHENTICATED PROXY SERVERS • Requirement SC-7(18) BOUNDARY PROTECTION | FAIL SECURE • Requirement SC-8(1) TRANSMISSION CONFIDENTIALITY AND INTEGRITY | CRYPTOGRAPHIC OR ALTERNATE PHYSICAL PROTECTION • Requirement SI-23 Information Fragmentation • Requirement SC-11 TRUSTED PATH • Requirement SC-16 TRANSMISSION OF SECURITY AND PRIVACY ATTRIBUTES • Requirement SC-17 PUBLIC KEY INFRASTRUCTURE CERTIFICATES • Requirement SC-19 VOICE OVER INTERNET PROTOCOL • Requirement SC-20 SECURE NAME / ADDRESS RESOLUTION SERVICE (AUTHORITATIVE SOURCE) • Requirement SC-21 SECURE NAME / ADDRESS RESOLUTION SERVICE (RECURSIVE OR CACHING RESOLVER) • Requirement AC-24 ACCESS CONTROL DECISIONS

After an analytical tool is used to conduct the vulnerability/risk assessment, more work can be done using UAF to integrate Security Views into the architecture model. The following section describes this additional work.

6 UAF Security Views

Cyber vulnerabilities are modeled in UAF using a "Risk" type element and SCs identified by TRACE were based on the FT. This sample model's content is not complete nor exhaustive, but the purpose is to illustrate how to use the UAF, RAAML, and SysML to model cyber resiliency. This section describes a series of Security Structure (Sc-Sr) Views that are created using UAF after the TRACE analysis is run. The views illustrate the identified risks and define *ResourceMitigation* elements (that satisfy SCs that were identified by TRACE analysis). Further development using UAF to create a complete security architecture model is also possible but not included. Such a security model can include security systems (facilities, personnel, hardware, software, etc.) and associated security processes, data and other Security Views that would comprise a security perspective aimed at protecting the resources described in the larger architecture model. The Sc-Sr Views described in this section show relationships defined from the security elements back to the system model elements (from FT) that they are aimed to protect. Figure 9 shows the SCs identified by TRACE for a subset of the system model, along with identified security risks. Figure 10 shows applicable enhanced SCs that are generated from the UAF SC Library.

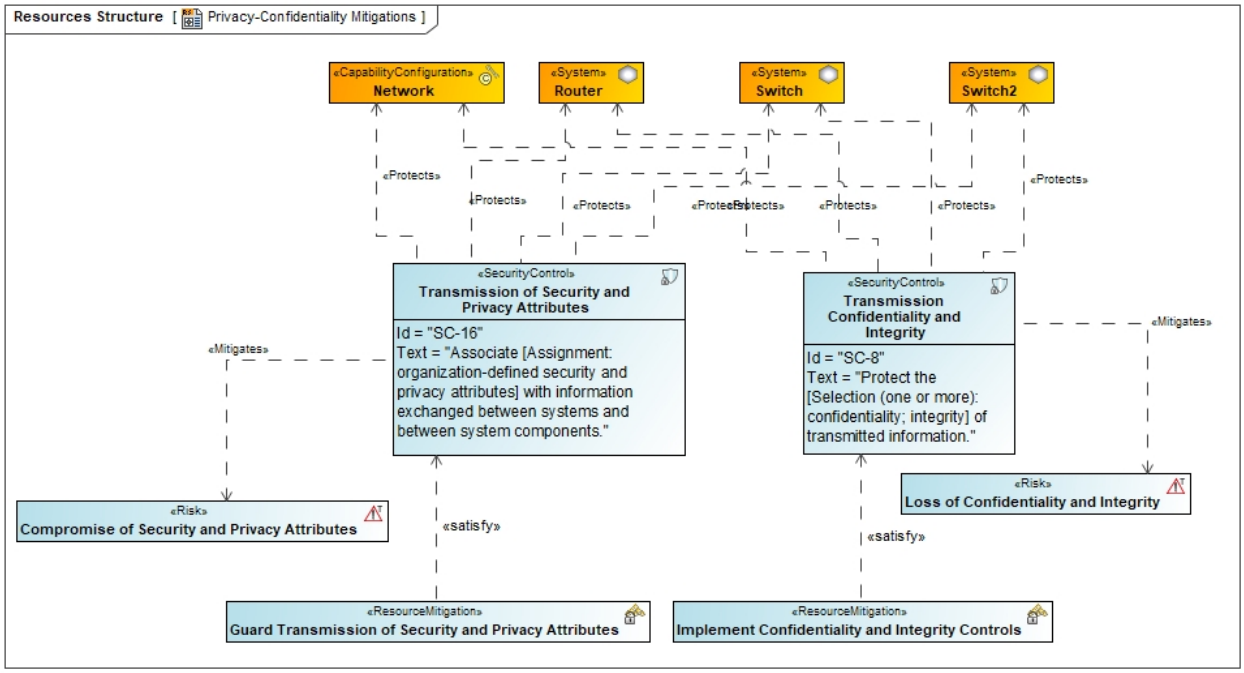

Figure 9: Security Controls For Switch And Other System Components

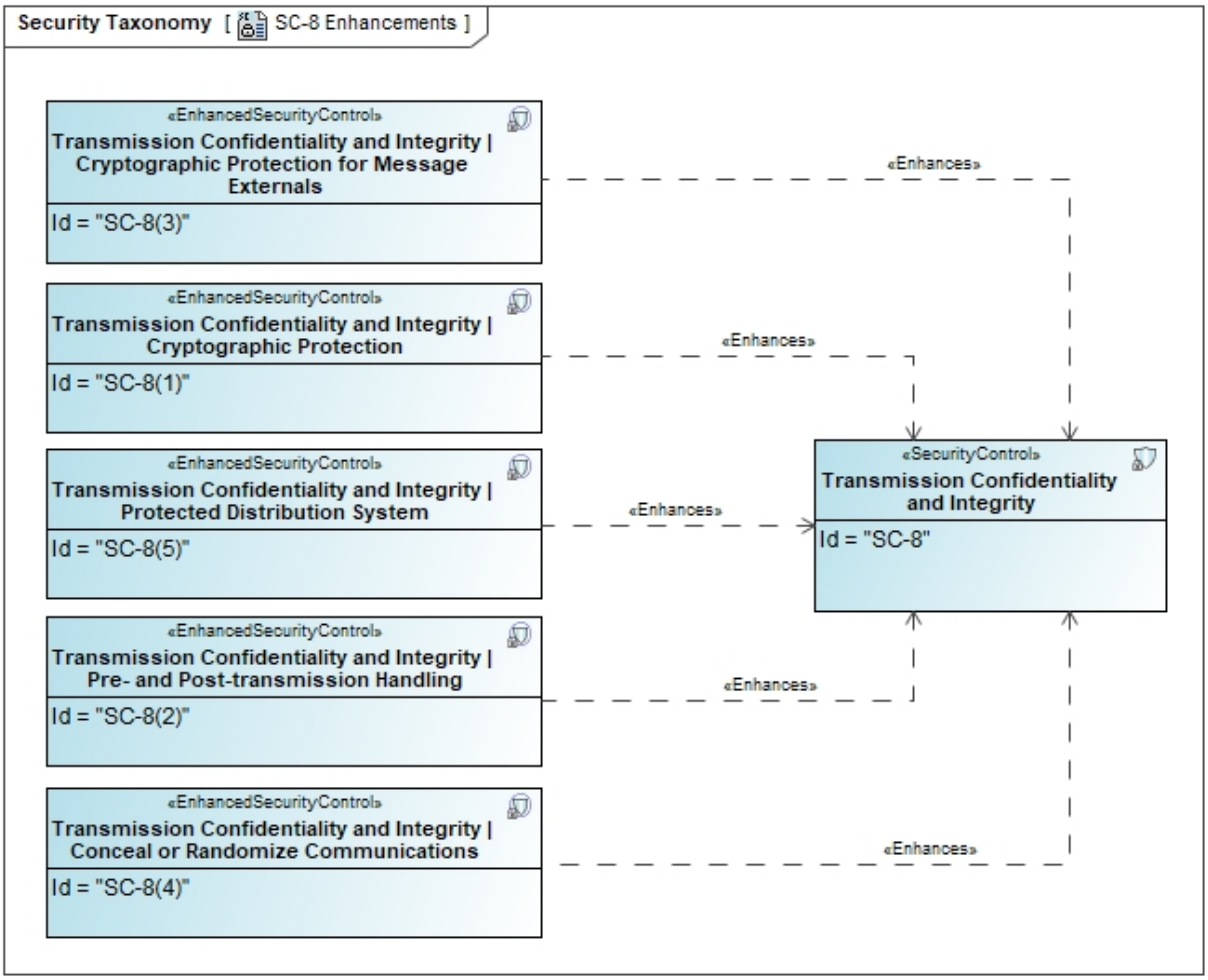

Figure 10: Security Enhancements For SC-8 Identified By UAF SC Library

Figure 11 shows a subset of security requirements with relations from the corresponding resource mitigation elements defined as a result of the security analysis. Traceability from mitigations back to requirements is defined and documented in the architecture model using SysML's *satisfy* relations.

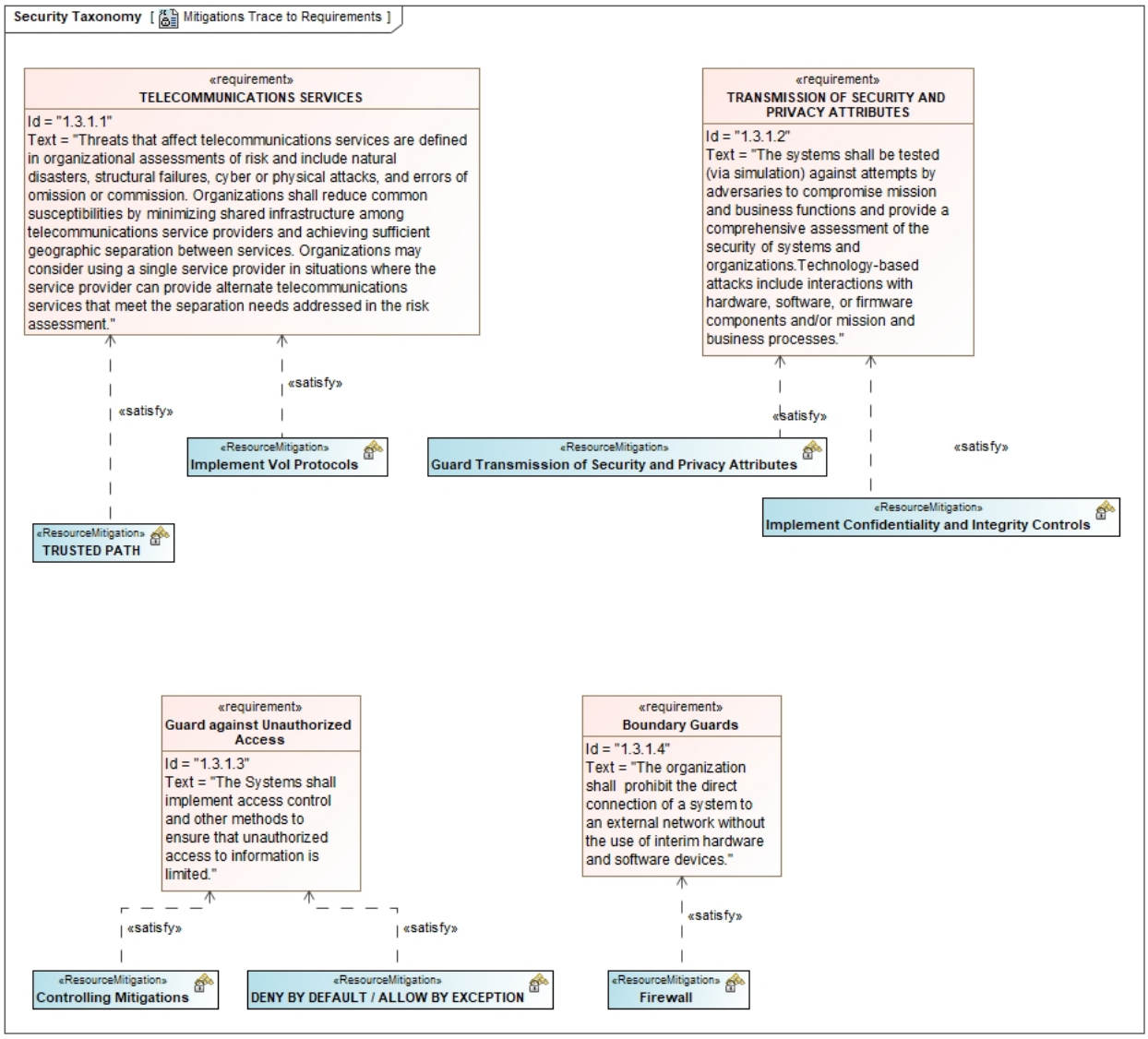

Figure 11: Traceability Of Risk Mitigations To Requirements Using UAF

7 Recommendations

Recommendation 1: Use the architecture model as the overarching model to maintain traceability from system design to cyber resiliency analytical models. Modeling security risks, mitigations, and SCs within the architecture model and relating those elements to other architecture model elements such as requirements and system design, enables a visualization of the essential cyber resiliency components. Visually modeling security risks, mitigations, and SCs within the architecture model moves the record of authority from documents to enable model-based traceability and analysis of the models.

Recommendation 2: Conduct rigorous cyber resiliency analysis using cyber analysis tools. One such tool referenced in this paper is MITRE's TRACE.

Recommendation 3: The architect is responsible for providing the overarching model. The need for additional analytical models may be identified by the architecture team in collaboration with

the teams working on various areas of specialization (e.g., cyber security community who gather daily logs). The architect is also responsible for validating that the downstream analytical analyses are interpreting the model as intended. There are cases where the architect has used a certain modeling convention and then the analyst assumed it meant something else. A feedback loop and continuous communication regarding the responsibility for interpretability of the model over its lifecycle goes hand-in-hand with the responsibility for generating the model.

Recommendation 4: The need to maintain traceability to the architecture and to validate that the various teams are working towards a consistent architecture model is the responsibility of the program manager in collaboration with the architecture team. The program manager's leadership can ensure that the architecture team and the cyber security team coordinate and maintain traceability across their models with a feedback loop across models (architectural and analytical). Empowered by the program manager's support, the architecture team ensures that the cyber resiliency analysis findings trace back to desired capabilities, and that the system design documents security risk mitigations.

8 Summary

This paper described how to define and model UAF Security Views and incorporate cyber resiliency analysis within an architecture model by applying an MBSE approach using UAF, RAAML, and SysML. A cyber resiliency analysis tool, MITRE's TRACE, was also used to describe how one might integrate cyber resiliency analysis with the architecture model. A sample model for SAR was utilized to illustrate how resiliency can be described in an architecture model, how cyber resiliency analysis findings trace back to desired capabilities, and how cyber resiliency, security risks, and security risk mitigations are documented in an architecture model with traceability to system components and to cyber resiliency requirements.

9 References

[1] Open Management Group, April 2020. Unified Architecture Framework Profile (UAFP) Version 1.1, OMG Document -- formal/19-11-07 https://www.omg.org/spec/UAF/1.1/UAFP/PDF

[2] Open Management Group, January 2021, Risk Analysis and Assessment Modeling Language (RAAML) v1.0 FTF, OMG Document -- ptc/21-01-01 (Risk Analysis and Assessment Modeling Language (RAAML), v1.0 beta 1)

[3] Friedenthal S., Moore A., Steiner S. (2014). A Practical Guide to SysML: The Systems Modeling Language, 3rd Edition. Ny, Ny: Elsevier.

[4] NIST Special Publication (SP) 800-160 Vol 2, "Developing cyber resilient systems: a systems security engineering approach," November 2019, https://doi.org/10.6028/NIST.SP.800-160v2

[5] NIST SP 800-53 Rev. 4, "Security and privacy controls for federal information systems and organizations," April 2013, https://doi.org/10.6028/NIST.SP.800-53r4

[6] TRACE, Traversal-driven Risk Assessment of Composite Effects GitHub - MITRE/trace: Traversal-driven Risk Assessment of Composite Effects

[7] Blade RiskManager (BRM) by KDM Analytics suite, KDM Analytics: automated risk assessment for cyber systems

[8] Open Management Group, June 2015, XML Metadata Interchange (XMI®) Specification, Version 2.5.1, OMG Document -- formal/2015-06-07 https://www.omg.org/spec/XMI/2.5.1/PDF

[9] MITRE ATT&CK® Matrix for Enterprise Matrix - Enterprise | MITRE ATT&CK®

Appendix A Security Structure (Sc-Sr) Views

A number of Security Risks and Mitigations (based on Security Controls [SCs] identified by Traversal-drive Risk Assessment of Composite Effects [TRACE]) for Search and Rescue (SAR) system components are created in Security Structure (Sc-Sr) Views, using the *Protects* Traceability Relationship to link applicable SCs to system components. Only one sample diagram was shown in the body of the paper. Below are the rest of the Sc-Sr Views created for the elements, the identified SCs and the applicable risks and mitigations identified by the security teams based on TRACE analysis. See Figures (12-21) below.

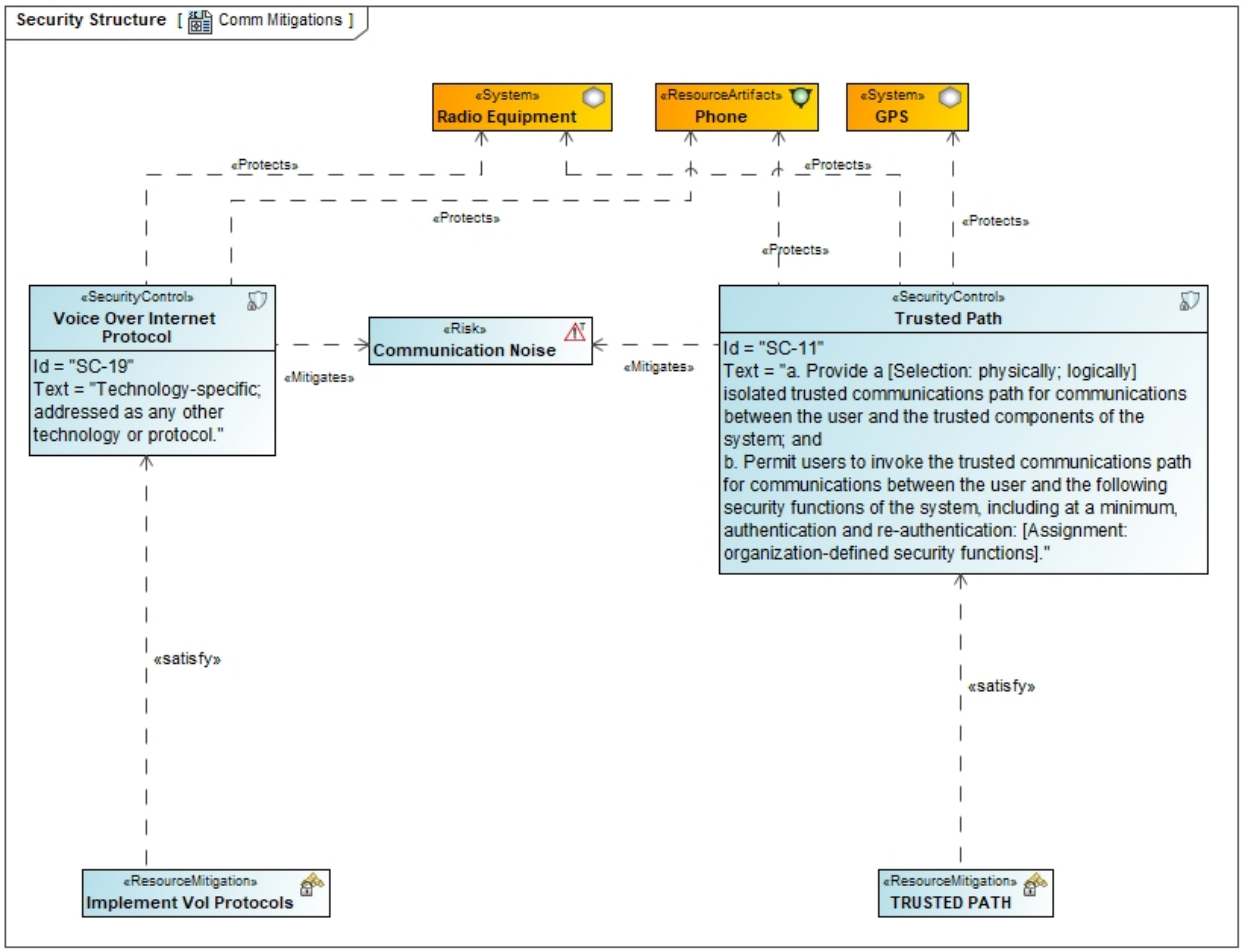

Figure 12: SCs for Radio, Phone, and Global Positioning System

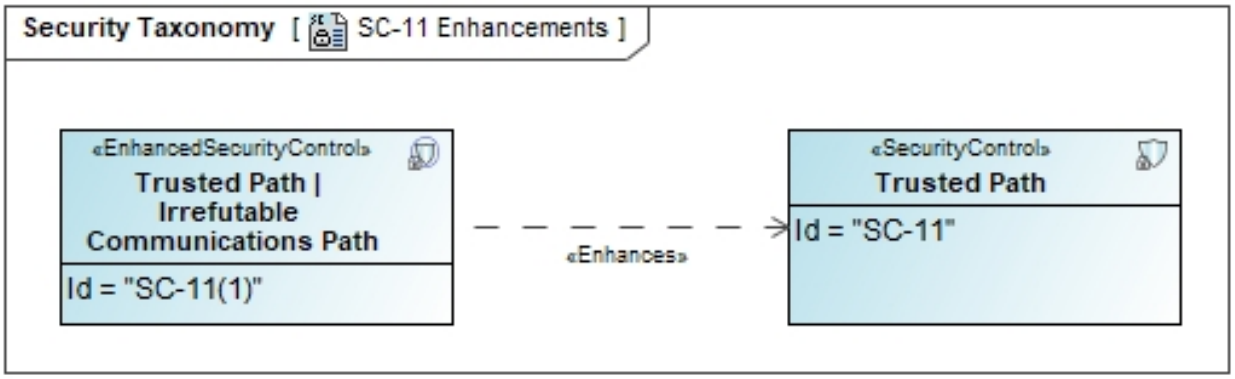

Figure 13: Security Enhancements for SC-11

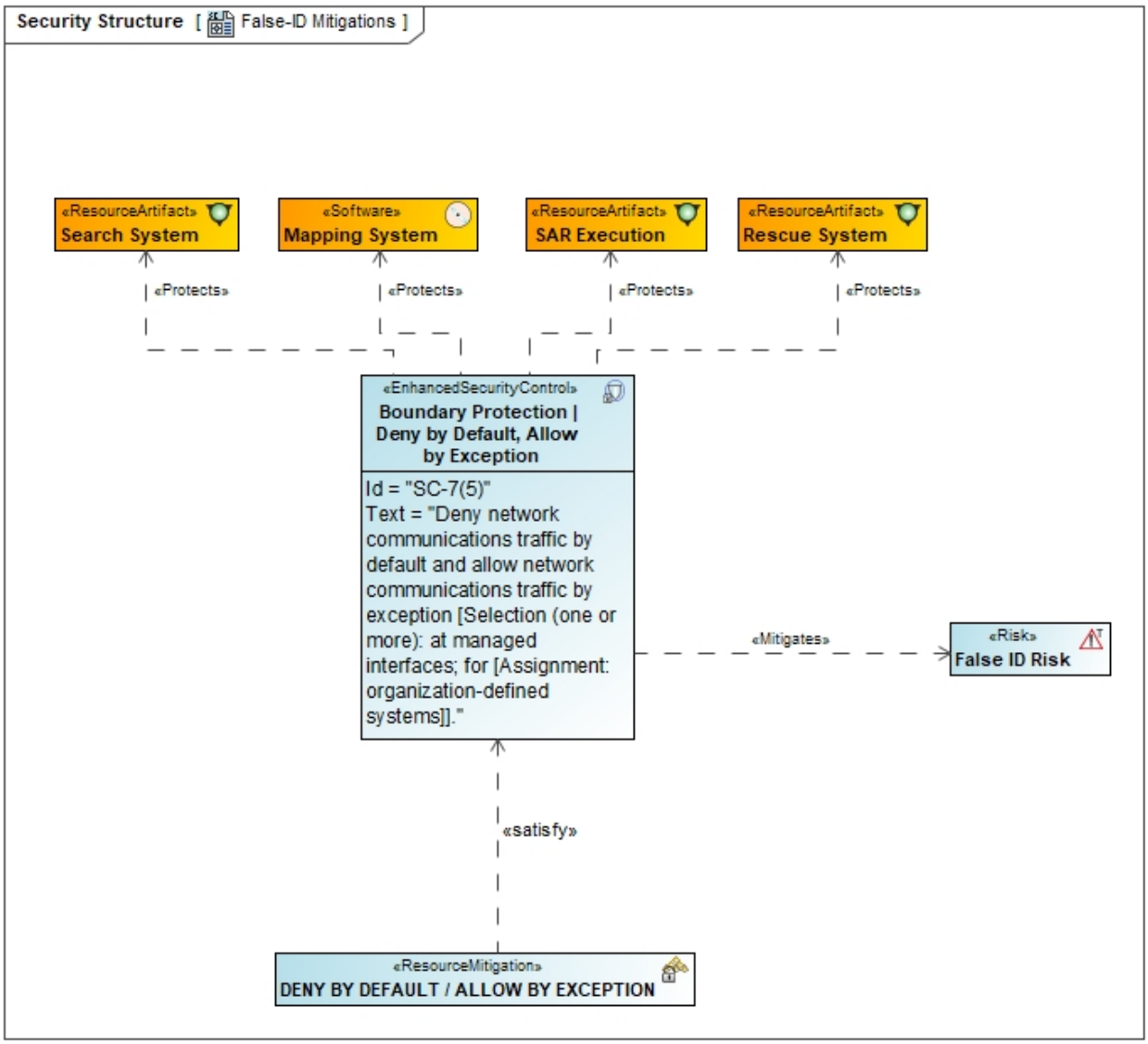

Figure 14: SCs For SAR Systems

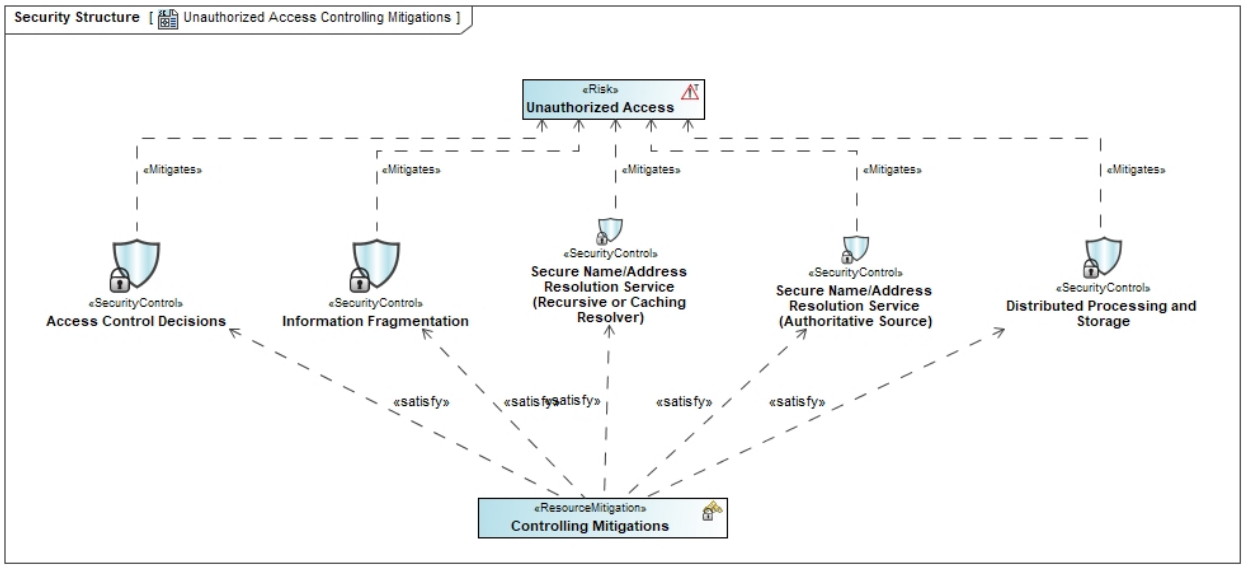

Figure 15: Risks And Mitigations Identified For SCs

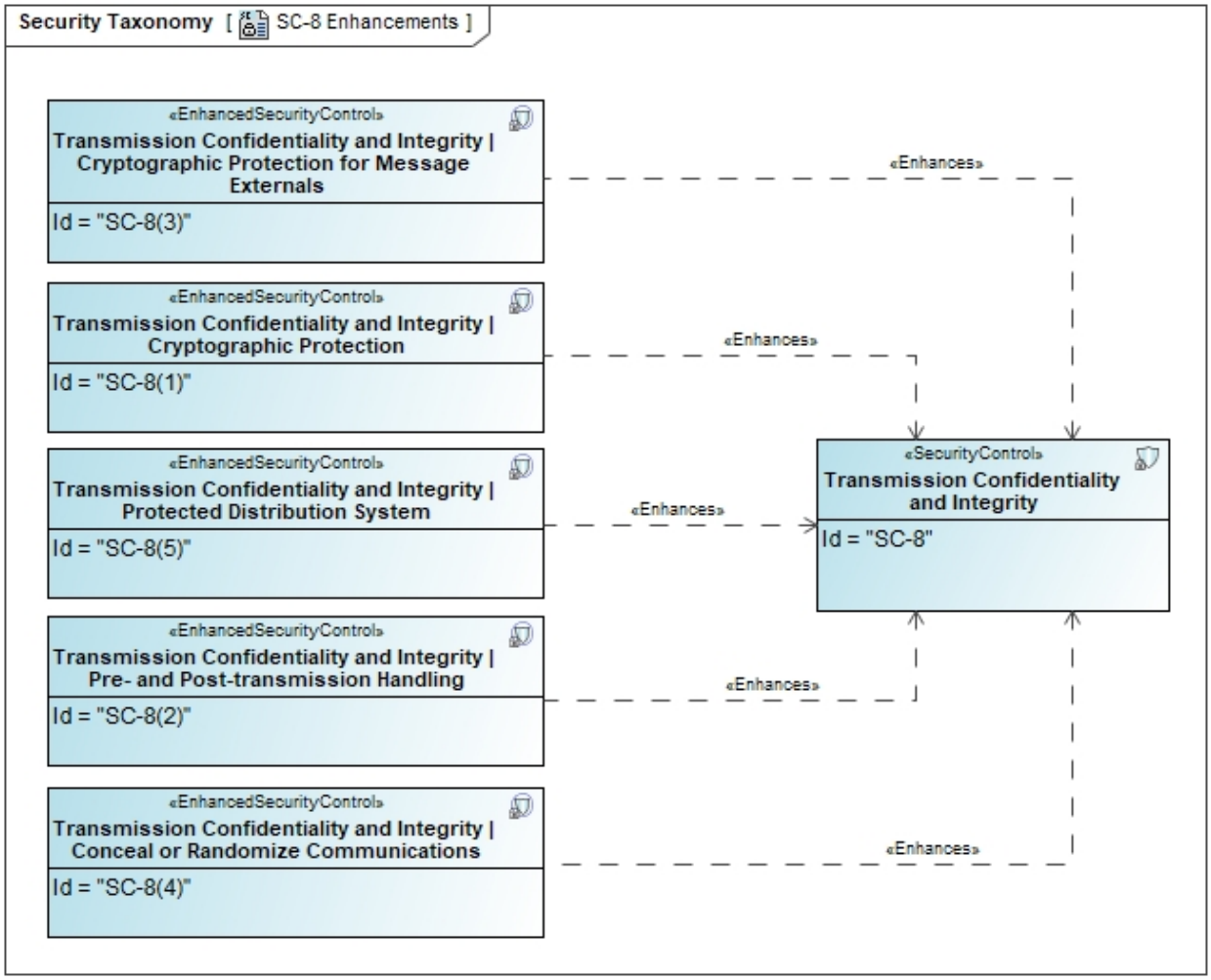

Figure 16: Enhancements For SC-8

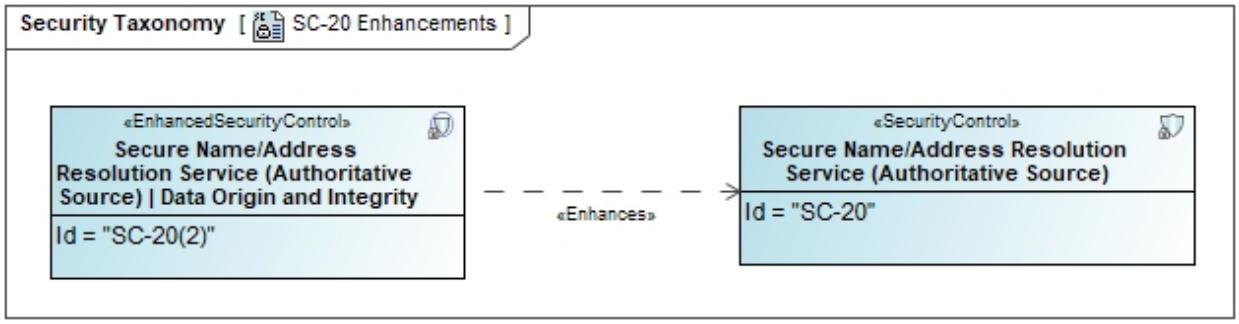

Figure 17: Enhancements For SC-20

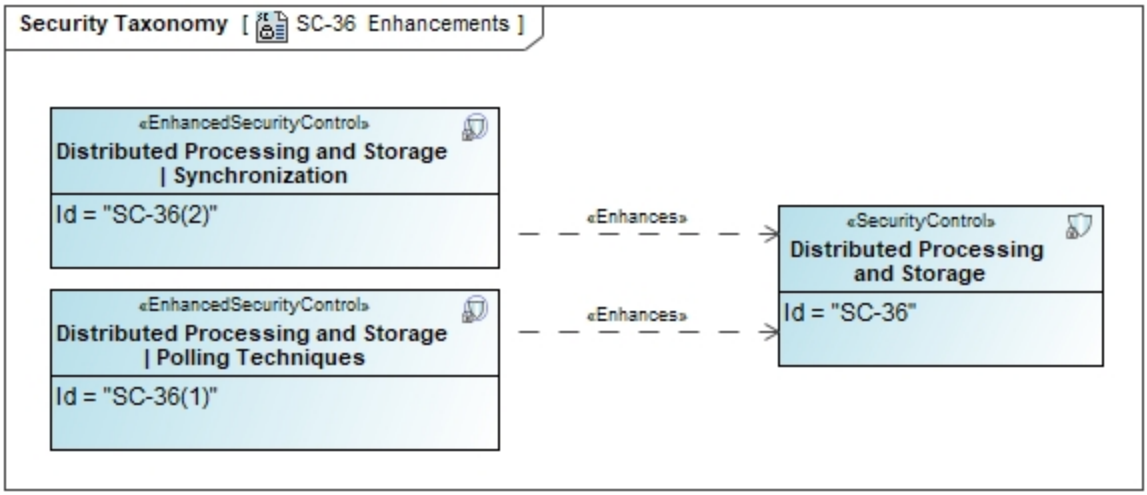

Figure 18: Enhancements For SC-36

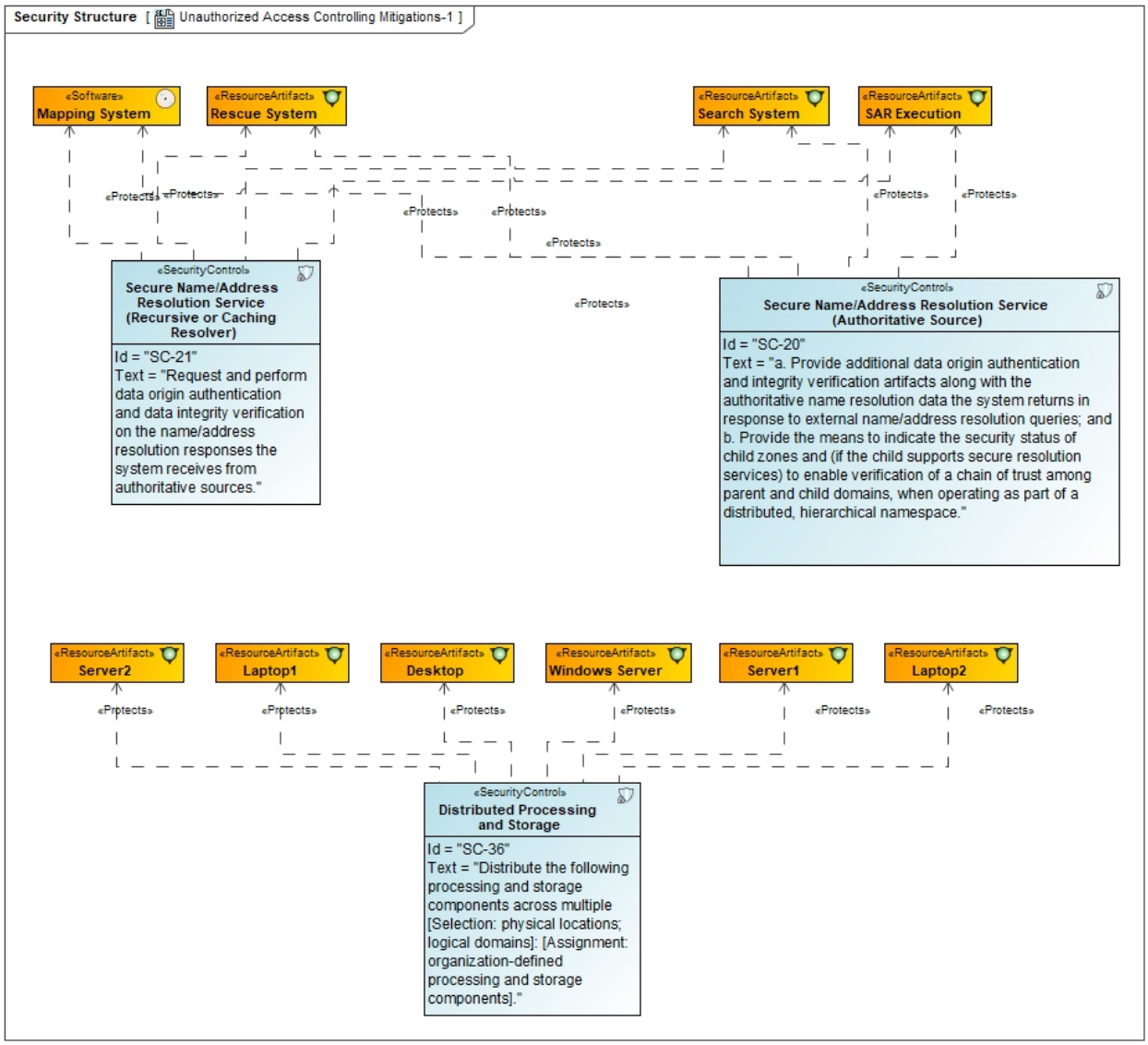

Figure 19: SCs For Various Systems Elements

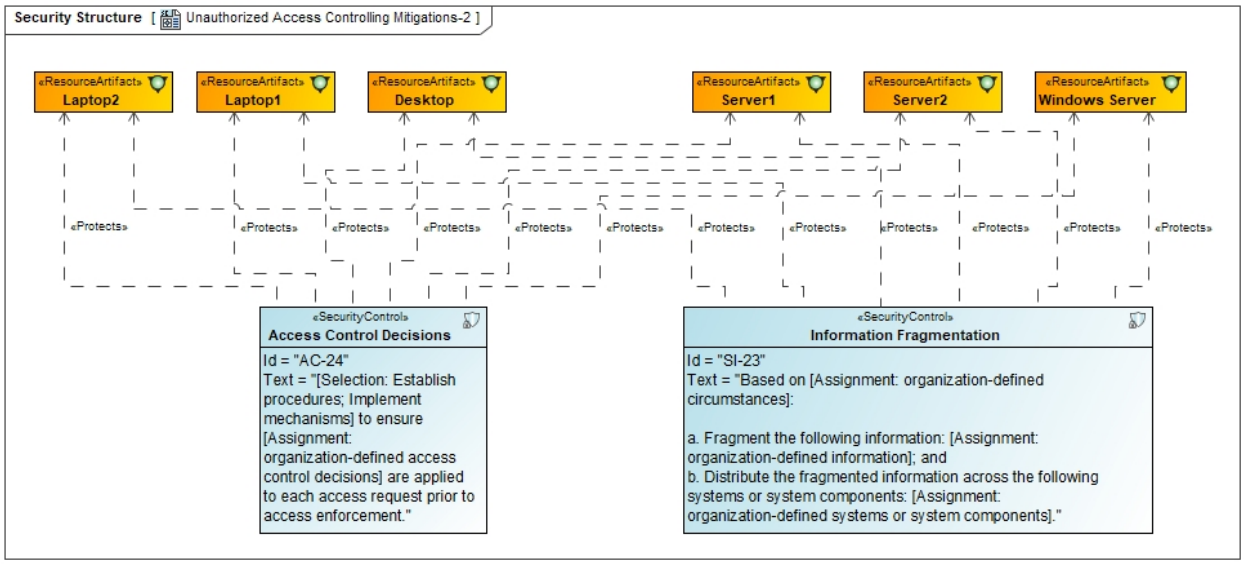

Figure 20: SCs For Various Systems Elements

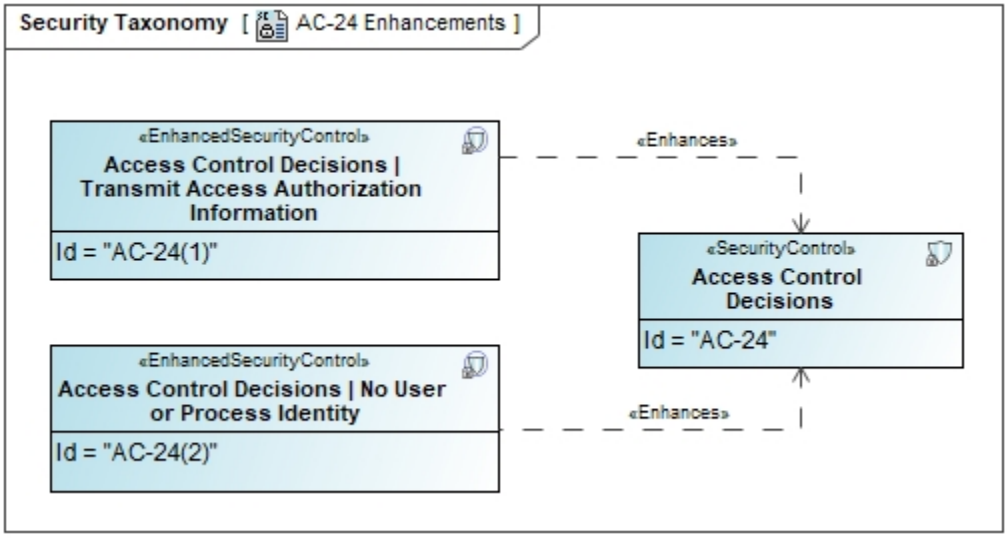

Figure 21: Enhancements For SC-24

Appendix B Abbreviations And Acronyms

Term	Definition
BRM	Blade Risk Management
C2	Command and Control
COTS	Commercial Off-The-Shelf
CSV	Comma-Separated Values
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
FT	Fault Tree
GOTS	Government Off-The-Shelf
ibd	Internal Block Diagram
MBSE	Model-Based Systems Engineering
NIST	National Institute of Standards and Technology
OMG	Object Management Group
RAAML	Risk Assessment and Analysis Modeling Language
RMF	Risk Management Framework
Rs-Sr	Resource Structure
SAR	Search and Rescue
SC	Security Control
Sc-Sr	Security Structure
SysML	Systems Modeling Language
TRACE	Traversal-drive Risk Assessment of Composite Effects
UAF	Unified Architecture Framework
XMI	XML Metadata Interchange
XML	Extensible Markup Language

NOTICE

This technical data was produced for the U. S. Government under Contract No. FA8702-22-C-0001, and is subject to the Rights in Technical Data-Noncommercial Items Clause DFARS 252.227-7013 (FEB 2014)