Towards automated threat modeling for space systems via SPARTA matrix

Joonhyuk Park
School of Cybersecurity
Korea University
peter990527@korea.ac.kr

Jiwon Kwak
School of Cybersecurity
Korea University
jwkwak4031@korea.ac.kr

Geunwoo Baek
School of Cybersecurity
Korea University
sinse100@korea.ac.kr

Dohee Kang
School of Cybersecurity
Korea University
kangdohee1211@korea.ac.kr

Seungjoo Kim*
School of Cybersecurity
Korea University
skim71@korea.ac.kr

***Abstract*—The increasing significance of space-system cybersecurity in the space industry underscores the necessity of moving beyond development paradigms based on *security by obscurity*. Consequently, international standards such as ISO 20517 recommend the use of threat modeling to ensure security when developing space systems. Because manual threat modeling is time-consuming, it has motivated substantial research into the development of automated tools. Despite this interest, automated threat modeling tools specialized for the space domain remain scarce. Therefore, this paper proposes an automated threat modeling tool for the space domain by enhancing the Microsoft Threat Modeling Tool (MS-TMT). The tool was developed by integrating the Aerospace SPARTA matrix and the D3FEND knowledge base into MS-TMT. To evaluate its effectiveness, we conducted a case study involving four space-system security incidents, including the Viasat hacking. In the absence of existing satellite-specific threat modeling tools, we selected SecOpsTM as a comparative baseline because it is an automated threat modeling tool that identifies threats in a manner conceptually similar to our approach, enabling a fair and meaningful comparison. The quantitative evaluation demonstrated that our tool achieved an accuracy of 100%, whereas SecOpsTM achieved an average accuracy of 54%.**

I. Introduction

The number of space systems in Earth orbit has dramatically increased from tens in 2012 to thousands by 2023 [1]–[3]. In particular, the advent of Low Earth Orbit (LEO) mega-constellations, such as Starlink and OneWeb, is expected to further increase the importance of space systems, as these constellations provide essential services to modern society, including Earth observation, machine-to-machine communication, and Internet services [4]–[7]. Historically, the security of space systems was primarily addressed through *security by obscurity* [8]–[10]. This approach relied on hiding system information—such as requirements and system design—to prevent potential adversaries from launching cyberattacks. However, this strategy has become insufficient, as recent reports indicate that attackers increasingly exploit human-related *weak links*—such as system operators—rather than merely targeting technical vulnerabilities [11]–[13]. These attacks often leverage social engineering techniques to trick operators into divulging login credentials or to lead to mistakes that cause the unintentional disclosure of sensitive information.

Consequently, several nations have recently published security policies and guidelines for the cybersecurity of space systems. Notably, the United States released the Space Policy Directive–5(SPD–5) in 2020, which outlines cybersecurity principles for systems operating in space [14]. Furthermore, the Biden administration enacted the S.3511 legislation in 2022 to enhance the cybersecurity of commercial space systems and has issued various guidelines for the secure operation of space systems [15]–[17]. Standardization bodies have also proposed security policies and requirements that space systems—including both ground infrastructure and spacecraft—must satisfy. For example, ISO 20517 and ISO 22893 standards were developed to systematically manage cyber threats throughout the entire lifecycle of space systems and to ensure mission assurance [18]. Specifically, ISO 20517 recommends System Theoretic Process Analysis for Security(STPA-Sec), based on the System Theoretic Accident Model and Processes (STAMP), to analyze space systems from a safety and security perspective [19].

However, a fundamental challenge of threat modeling is that its goal—to identify every potential threat within the system under analysis—can be time-consuming and costly when performed manually by an analyst without automated support. For instance, guidelines published by Security Compass indicate that using automated threat modeling tools can reduce its cost to approximately 10% of the cost incurred manually [20].

*Corresponding author.

Workshop on Security of Space and Satellite Systems (SpaceSec) 2026
23 February 2026, San Diego, CA, USA
ISBN 978-1-970672-02-2
https://dx.doi.org/10.14722/spacesec.2026.23067
www.ndss-symposium.org

Consequently, extensive research is actively being conducted on the threat modeling [21], [22], [24], [42], [43]. However, there is currently no dedicated automated threat modeling tool specifically tailored for systems operating in the space domain.

In this paper, we propose an automated threat modeling tool for space systems. We develop our tool by integrating Aerospace's SPARTA [29]—which catalogs threats relevant to space systems—and additional knowledge bases into MS-TMT, one of the most widely used automated threat modeling tools [25]. The remainder of this paper is organized as follows. Section 2 introduces the background and preliminaries necessary to understand our work. Section 3 reviews related work. Section 4 presents our method for specializing MS-TMT for space systems, detailing how the SPARTA and D3FEND databases [36] were analyzed and integrated. Section 5 describes the tool we developed in detail. Section 6 shows the results of a case study on two known security breach incidents in the space domain using our tool. Section 7 presents the evaluation results, focusing on the tool's accuracy in identifying threats from actual space-system incidents. Finally, Section 8 summarizes our research.

II. Background / Preliminaries

In this section, we provide the background and preliminaries necessary to understand the STRIDE threat modeling methodology and its automated threat modeling tool, MS-TMT.

A. STRIDE threat modeling

STRIDE is a threat modeling methodology originally proposed by Loren Kohnfelder and Praerit Garg at Microsoft [23]. Although it was initially developed for desktop software such as Windows and Office, STRIDE has since been widely applied across a broad range of systems. The methodology typically consists of the following five steps:

1) **Creating a Data Flow Diagram**(DFD): Modeling the target system by abstracting it into data flows and their interactions.
2) **Constructing an attack library**: Gathering threat information relevant to the target system.
3) **Analyzing threats**: Identifying threats applicable to each element of the DFD.
4) **Deriving an attack tree**: Developing attack scenarios based on the identified threats.
5) **Deriving mitigations**: Proposing mitigations to address the threats represented in the attack scenarios.

This methodology classifies potential threats into six distinct types. These six threats are defined as follows:

1) **Spoofing** (S): Impersonating an authorized entity.
2) **Tampering** (T): Modifying data stored on disk, transmitted across a network, or held in memory.
3) **Repudiation** (R): The ability of an actor to deny performing an action.
4) **Information disclosure**(I): Exposing information to unauthorized parties.
5) **Denial of service** (D): Actions that prevent the system from providing services to legitimate users.
6) **Elevation of privileges** (E): Gaining unauthorized privileges or capabilities beyond those legitimately granted.

B. MS-TMT

MS-TMT is Microsoft's threat modeling tool used within the Security Development Lifecycle (SDL) [25]. It enables users to construct DFDs and automatically identify potential threats based on the STRIDE methodology. To support these functionalities, MS-TMT defines templates consisting of the following components:

1) **Stencils**: The architectural elements of the system (i.e., DFD components).
2) **Threats**: A factor or condition that can negatively affect the stencils themselves or their interactions.

For threats, MS-TMT adopts predefined threat definitions based on the STRIDE methodology. For stencils, MS-TMT provides five stencil types, as shown below:

1) **Process**: Any executable code or function that handles data, such as executables, assemblies, or COM components.
2) **External entity**: A user, organization, or system outside the modeled boundary, beyond the control of the target system.
3) **Data flow**: The transfer of information between DFD elements, such as network messages or function calls.
4) **Data store**: A location where data is stored, such as files, databases, or registry keys.
5) **Trust boundary**: A boundary separating areas with differing privilege levels. Any data or execution flow crossing this boundary indicates a need for security review.

III. Related work

In this section, we summarize the recent researches related to our research topic. When selecting the papers to analyze, we followed a structured approach rather than simply searching academic databases.

A. Paper selection strategy

We collected and analyzed recent papers similar to our research topic following the three steps below.

1) **Keyword-based collection**: We queried across major academic databases—including IEEE, Elsevier, and Google Scholar— using a combined set of keywords. The first keyword group included terms such as "space system", "space segment", and "satellite", while the second group included "threat modeling tool" and "automated threat modeling". We restricted the results to papers published within the last five years, yielding **84 papers** in the initial stage.
2) **Screening based on abstracts and conclusions**: We examined the abstract, introduction, and conclusion of each paper to determine its relevance. Papers not directly related to either cybersecurity for space systems or threat

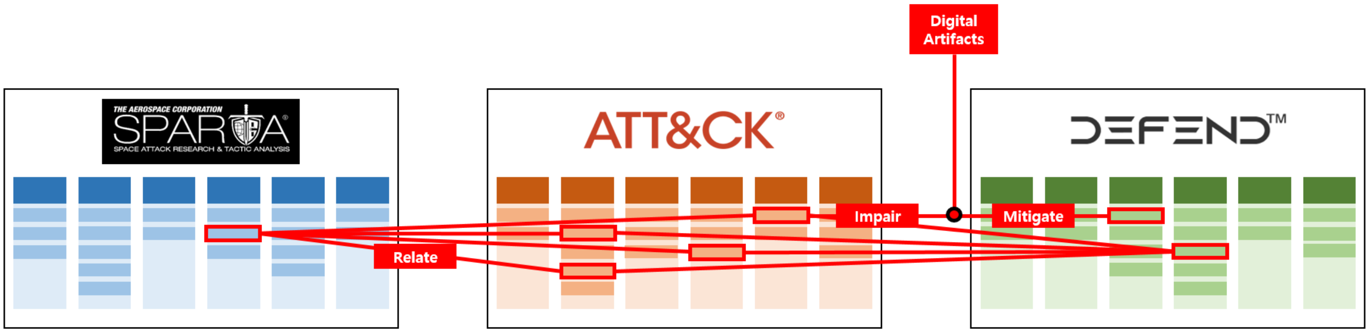

Digital Artifacts
THE AEROSPACE CORPORATION
SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS
ATT&CK®
D3FEND™
Relate
Impair
Mitigate

Fig. 1: Bridging SPARTA TTPs to D3FEND artifacts using MITRE ATT&CK TTPs

modeling methodologies/tools were excluded. Through this screening process, **25 papers** remained.

3) **Reviewing full-text**: We reviewed the main body of these 25 papers and ultimately selected **3 papers** that were most relevant to our research scope.

B. Analyzing related works

In 2024, Kvam Frøseth conducted threat modeling for maritime operations that rely on LEO satellite communications [22]. This research applied STRIDE threat modeling to the Starlink LEO communication system and used the SPARTA matrix to assess attack scenarios affecting maritime ground infrastructure. The results demonstrated that STRIDE provides a holistic view of threats—identifying Spoofing, Information Disclosure, and Denial of Service (DoS) as high-risk threats—while SPARTA validated that an SDR-based GPS spoofing attack could lead to severe maritime incidents. A key limitation, however, is the absence of a threat modeling framework tailored for maritime-satellite environments, requiring manual refinement of both STRIDE and SPARTA to match domain-specific assets and threat characteristics.

In 2025, Tran et al. introduced TerrARA, an automated threat modeling tool for IaC (Infrastructure as Code)-based cloud environments, focusing primarily on AWS Terraform configurations [21]. TerrARA automatically extracts DFDs from Terraform files and applies the SPARTA engine with 72 cloud-specific threat patterns. Strengths include: (1) accurate reflection of the deployed system through IaC parsing, enabling CI/CD integration, and (2) fine-grained cloud-oriented threat detection. However, TerrARA faces challenges such as difficulty detecting implicit communication paths not represented in IaC files, complexity in node consolidation, and the absence of threat prioritization.

In 2025, Amro et al. proposed ThreatSpider, a Cyber Threat Intelligence(CTI) driven semi-automated threat modeling framework [24]. ThreatSpider integrates multiple CTI sources—including ATT&CK, ATLAS, EMB3D, SPARTA, and NVD—to automate threat identification, mitigation generation, and cybersecurity requirement derivation throughout the system development lifecycle. The framework offers advantages such as reduced manual effort, improved model currency through CTI-based updates, and support for compliance with standards like IEC 62443 and ISO 27001. Its limitations include insufficient domain support (e.g., for space systems), the need for more refined system property mappings across CTI sources, and the lack of integrated threat prioritization or risk assessment features.

IV. Our Approach: Integrating SPARTA and D3FEND into MS-TMT for Space Domain

In this section, we describe the process by which we integrate the SPARTA and D3FEND databases into MS-TMT to specialize the tool for the space domain. To define stencils for modeling space system as well as ground infrastructures to identify potential threats for each stencil, we relied on the D3FEND and SPARTA cybersecurity databases, respectively. To determine whether threats documented in SPARTA can be applied to the artifacts defined in D3FEND, we leveraged MITRE ATT&CK as a common logical intermediary referenced by both sources. This forms the logical chain: **SPARTA TTP - MITRE ATT&CK TTP - D3FEND artifact**, as shown in figure 1.

A D3FEND artifact represents any object that can be compromised by a threat actor. The D3FEND Artifact Ontology (DAO) structurally links ATT&CK-defined threats to the artifacts they can target [36]. This structural relationship enables us to both refine system component definitions and systematically determine which threats can affect each component.

A. Defining stencils in template with D3FEND

We began by investigating the components and objects of spacecraft and ground infrastructure. We analyzed documentation such as open-source software requirements, NASA design reports, and cybersecurity guidelines for ground infrastructure issued by international governmental and organizational bodies. However, this information was inconsistent across documents, varying widely depending on mission profiles and security policies.

To address this inconsistency, we systematically categorized space system components into domain-specific segments and subsystems. We referenced classifications in prior literature [30]–[35], including recent survey papers and NASA technical documents that distinguish bus and payload components.

Through this process, we identified the initial set of stencils for the threat modeling template, including subsystems from the space segment (e.g., attitude control, power, communication subsystems) and from the ground segment (e.g., mission analysis, tracking, and transmission modules).

However, we recognized that these initial components were insufficient for comprehensive threat modeling for 2 reasons.

[Identified Issues]

- Not all digital objects and relevant data that comprise the space system were fully identified (e.g., OS process, file system, network flow type). The general IT components that make up the ground infrastructure were not sufficiently considered in the initial set.
- Lacking clear rationales or logical links for determining which specific threats could apply to each component.

To resolve these issues, we used the DAO from D3FEND to extend the component set in two ways:

[Our actions]

- We incorporated general IT infrastructure elements that are also present in space systems (e.g., file systems, databases, OS processes) into the stencil definitions.
- We derived properties and constraints for each stencil based on DAO semantics to enable precise threat identification.

For example, D3FEND defines a configuration file as a subtype of file. Accordingly, we assigned the property "stores configuration data" to the file-system stencil. During modeling, whether a DFD element represents a configuration-file storage determines which threats are applicable, based on DAO-derived constraints. After defining stencil properties, we mapped all components to the fundamental DFD elements: process, data store, external entity, data flow, and trust boundary. For instance, the communication subsystem of the space segment was mapped to a Process element as "Space-Segment Communication Module Process."

B. Defining threats in template with SPARTA

To identify threats specific to the space domain, we redefined the general STRIDE-based threats provided by MS-TMT. This was necessary because the default STRIDE definitions focus primarily on general-purpose software systems and do not adequately capture the threat landscape of the space environment. Thus, we derived space-specific threats by referring to the Tactics, Techniques, and Procedures (TTPs) documented in Aerospace's SPARTA framework, which extends MITRE ATT&CK for space systems.

After redefining these threats, we used the previously described MITRE ATT&CK–DAO structural relationship to determine the system components to which each threat may apply. This allowed us to map SPARTA TTPs to components aligned with D3FEND artifacts, using ATT&CK techniques as the logical bridge.

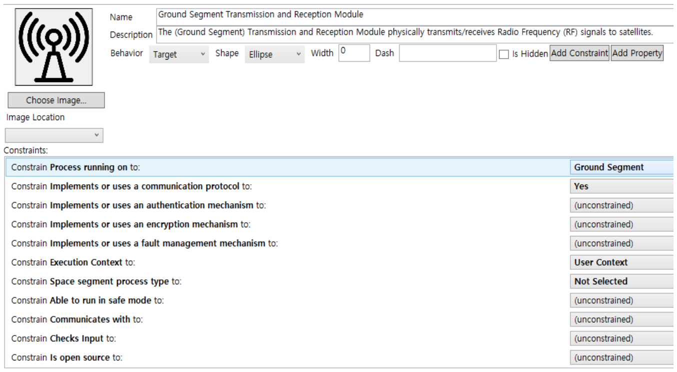
During this process, we identified several missing links in which specific SPARTA threat elements lacked corresponding D3FEND artifacts. These gaps limited the ability to determine which system components could be affected by particular attack techniques. To resolve this issue, we analyzed officially reported satellite-related security incidents [26] to identify which components were targeted in real attacks. This analysis enabled us to manually establish the missing associations between SPARTA threats and D3FEND artifacts, ultimately ensuring completeness in the specialized threat model.

V. Tool Implementation

Building upon the approach described earlier, this section explains the implementation details of our space-system template for MS-TMT. We developed a template that defines stencils for modeling space systems across the ground, user, space, and link segments. In addition, we specified the conditions under which threats are instantiated within the template. The complete version of the template, including usage examples, is publicly available in our repository [37].

A. Stencils in template

The stencil set consists of 34 elements, comprising 6 basic stencil types and 28 specific stencils that represent concrete instances of those types. Each stencil is defined with properties such as its operational environment, privilege level, and so on.



Name: Ground Segment Transmission and Reception Module

Description: The (Ground Segment) Transmission and Reception Module physically transmits/receives Radio Frequency (RF) signals to satellites.

Behavior: Target Shape: Ellipse Width: 0 Dash: □ Is Hidden

Constraints:

Constrain **Process running on** to:	**Ground Segment**
Constrain **Implements or uses a communication protocol** to:	**Yes**
Constrain **Implements or uses an authentication mechanism** to:	(unconstrained)
Constrain **Implements or uses an encryption mechanism** to:	(unconstrained)
Constrain **Implements or uses a fault management mechanism** to:	(unconstrained)
Constrain **Execution Context** to:	**User Context**
Constrain **Space segment process type** to:	**Not Selected**
Constrain **Able to run in safe mode** to:	(unconstrained)
Constrain **Communicates with** to:	(unconstrained)
Constrain **Checks Input** to:	(unconstrained)
Constrain **Is open source** to:	(unconstrained)

Fig. 2: Stencils for transmission and reception module in our template

As shown in figure 2, we defined the transmission and reception module of the ground segment as an MS-TMT stencil. This module is responsible for the physical transmission and reception of Radio Frequency (RF) signals exchanged between the ground segment and the space segment. In its configuration, *Process running on* is set to ground segment, and the property *Implements or uses a communication protocol* is set to *Yes*, indicating that the module either implements or utilizes a communication protocol. This stencil is used to analyze threats targeting the communication boundary between the space and ground segments—one of the most critical interfaces in space systems.

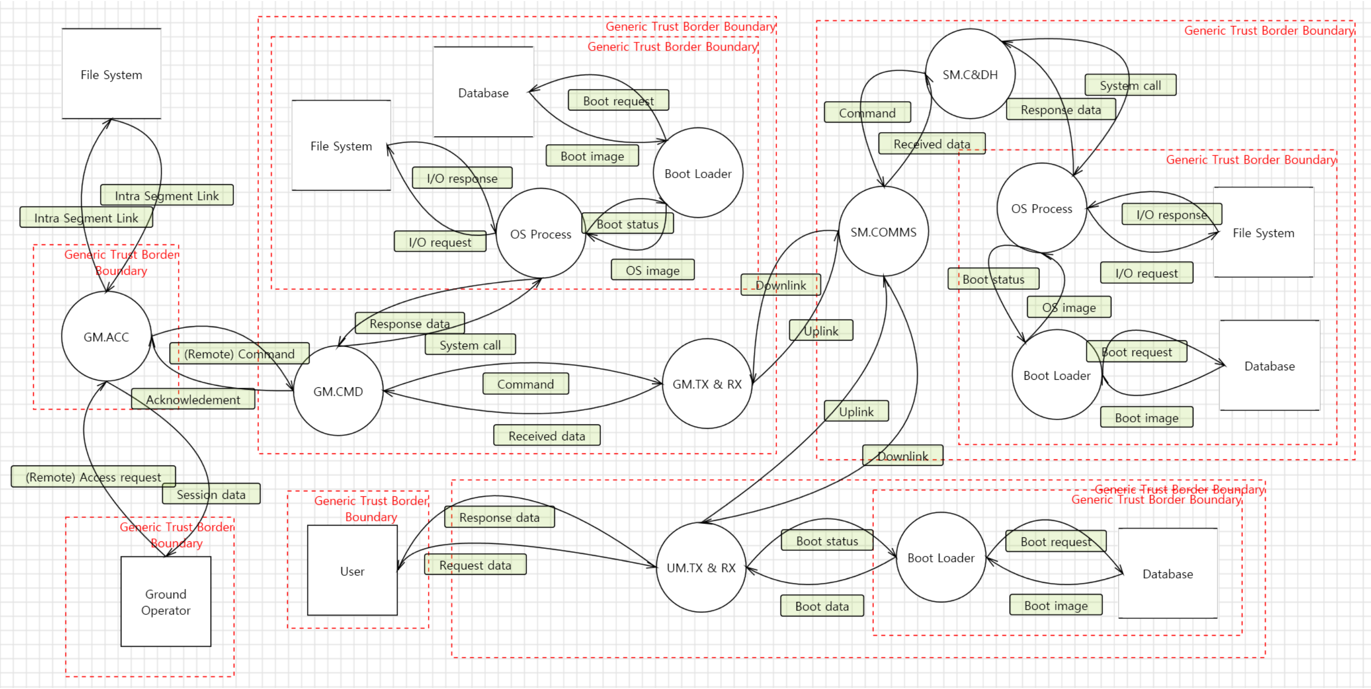


Fig. 3: Modeling KA-SAT system in MS-TMT

B. Threats in template

We defined a total of 131 unique threat types. To determine the threat-instantiation conditions, we referred to the SPARTA TTP - space component mapping established from the previous process. We also analyzed real-world space-domain security incidents and extracted common patterns indicating where threats could arise within a DFD. These patterns were then formalized into threat-identification rules in MS-TMT.

> **Title**: [IA-0002] Compromise Software Defined Radio
> **Threat identification rule**: Conditions under which this threat is instantiated in the model.
>
> - **Include**: (target is [Space Segment Communication Module Process] or target is [Ground Segment Transmission and Reception Module]) and (flow crosses [Generic Trust Line Boundary] or flow crosses [Generic Trust Border Boundary])
> - **Exclude**: -

One of the threats included in our template is *Compromise Software-Defined Radio*. This threat targets the module responsible for RF signal transmission and reception. It can be triggered not only when an attacker gains physical access to a ground station but also when a malicious spacecraft attacks another spacecraft in orbit. Accordingly, we defined its target as either the *Space Segment Communication Module Process* or the *Ground Segment Transmission and Reception Module Process*. Furthermore, the threat is instantiated only when a data flow crosses a *trust boundary*—for example, when a privileged attacker bypasses security mechanisms and gains unauthorized access across the Generic Trust Boundary. This reflects the escalation conditions under which a compromise of the RF chain becomes a significant security threat.

VI. Case Study

In this section, we present the results of the threat modeling conducted for two known security breach scenarios in the space domain—using our specialized MS-TMT template. Specifically, the Viasat hacking incident and the memory manipulation attack demonstrated in DefCon 2022.

A. Viasat hacking incident

The Viasat hacking is a well-known security breach incident in the space domain. The attack occurred on February 24, 2022, when the Viasat KA-SAT network experienced a widespread service outage [27]. The attacker initially gained access through a misconfigured VPN device and subsequently executed malicious commands on modems communicating with the space segment. By overwriting the flash memory in the modem, the attacker rendered it inoperable, ultimately causing a denial-of-service of the satellite Internet service.

We first created the DFD of the KA-SAT system using the stencils defined in our tool, as illustrated in figure 3. The diagram is divided into four segments—ground, space, user, and link—separated by trust boundaries. Each segment contains components representing functional elements of the space system. The ground segment includes operational components such as the *command module*, the *transmission and reception module*, and the *network management module*, where the external entity, the *human operator*, interacts with the system via remote access. The space segment is composed of the *communication module* and the *C&DH module*, while the user segment consists of the *modem* and the *bootloader*, representing the end-user terminal infrastructure. The link segment is expressed as data flows connecting these components, capturing all communication channels, including intra segment links, ground to ground communication, and uplink/downlink communication.

By applying our template's threat-identification rules to the KA-SAT DFD, we identified 435 potential threats across the system. These threats encompass all SPARTA tactic categories and cover diverse attack surfaces spanning the ground, space, user, and link segments. Table 1 illustrates a representative subset of threats that were exploited in the actual Viasat incident.

ID	Threat Title
1	[REC-0005.04] Eavesdropping: Active Scanning (RF/Optical)
2	[RD-0002.01] Compromise Infrastructure: Mission-Operated Ground System
3	[IA-0008.01] Rogue External Entity: Rogue Ground Station Compromise Infrastructure: Mission-Operated Ground System
4	[EX-0010.02] Malicious Code: Wiper Malware
5	[PER-0005] Credentialed Persistence
6	[DE-0011] Credentialed Evasion
⋮	
434	[LM-0007] Credentialed Traversal
435	[IMP-0002] Disruption

TABLE I: Identified threats for the Viasat hacking

Specifically, many of the automatically generated threats align with those reported in the actual Viasat incident—such as threats involving unauthorized access to ground infrastructure, manipulation of modem firmware, and exploitation of misconfigured authentication mechanisms [28]. Additionally, several threats that were not explicitly mentioned in official reports but are plausible within the KA-SAT architecture were also identified, highlighting the tool's ability to reveal latent or overlooked attack paths.

B. DefCon 2022 - memory manipulation attack

This case was demonstrated by Brawner at DefCon 2022 [38]. He demonstrated how an adversary with knowledge of the target spacecraft's memory architecture can exploit the inherent trust relationship between ground and space segments. The attacker conducts extensive reconnaissance to understand memory maps and the interaction between the VxWorks operating system and PowerPC processor. With this knowledge, a single command packet transmitted through authenticated and encrypted channels can overwrite critical memory locations, causing the spacecraft's kernel to crash and rendering it unable to process information. This scenario was subsequently analyzed using the SPARTA framework, with a detailed kill chain published by Aerospace at DefCon 2023 [46].

Since the specific space system architecture used in this demonstration was not publicly disclosed, we applied our threat modeling approach to the same generic DFD used for the KA-SAT analysis. This DFD represents a representative space system model encompassing ground segment operational modules, space segment processing components, and the communication links between them. By utilizing this generalized architecture, we demonstrate the template's applicability across different space system scenarios without requiring detailed proprietary system information.

Applying our MS-TMT template to this scenario generated the same comprehensive set of 435 threats identified in the KA-SAT analysis. Table 2 presents the specific subset of threats that correspond to the attack techniques.

ID	Threat Title
1	[REC-0001.09] Gather Spacecraft Design Information: Fault Management
2	[REC-0003.02] Gather Spacecraft Communications Information: Commanding Details
3	[IA-0007.02] Compromise Ground System: Malicious Commanding via Valid GS
4	[IA-0009.03] Trusted Relationship: User Segment
5	[EX-0006] Disable/Bypass Encryption
6	[EX-0009.01] Exploit Code Flaws: Flight Software
7	[EX-0012.03] Modify On-Board Values: Memory Write/Loads
⋮	
432	[IMP-0002] Disruption
433	[IMP-0003] Denial
434	[IMP-0004] Degradation
435	[IMP-0005] Destruction

TABLE II: Identified threats for the memory manipulation scenario

Notably, our automated threat identification successfully captured all TTPs outlined in the kill chain presented by [46], including reconnaissance activities, exploitation of trusted command pathways, and direct memory access vulnerabilities. This demonstrates the template's capability to systematically identify relevant threats across diverse attack scenarios within the space domain.

VII. Evaluation

In this section, we describe the data and procedures used to evaluate the accuracy of our enhanced MS-TMT template in identifying threats relevant to space systems. We also present the results of comparing our template with an existing automated threat-modeling tool. The detailed threat-modeling results for each attack case used in the evaluation are available in our GitHub repository [37].

Because no automated threat-modeling tool specifically designed for space systems currently exists, we selected a comparative tool that produces threat information in a form most analogous to ours. The tool chosen was SecOpsTM, a Python-based automated threat-modeling tool that identifies potential threats from a DFD and outputs them in the MITRE ATT&CK TTP format. To ensure a fair comparison, we configured the tool with attribute values that mirror the conditions used in our own template.

A. Collecting data for evaluation

To evaluate the accuracy of our template, we first collected space system-relevant attack cases reported in academic research over the past five years. The following criteria were applied:

1) The selected cases had to include information at a level of detail comparable to threats documented in MITRE ATT&CK, rather than providing only Proof-of-Concept (PoC) vulnerabilities.
2) To ensure an objective evaluation, attack cases referenced during the development of our template were excluded.

Using these criteria, we collected four satellite-related security incidents. These included the well-known Viasat hacking incident and three additional cases reported at prominent security conferences such as DEFCON.

B. Evaluation process

We evaluated accuracy using the collected cases according to the following procedure:

1) Each attack case was documented and mapped to the corresponding MITRE ATT&CK TTPs. This mapping was based on expert analyses that reference MITRE ATT&CK. The manually identified ATT&CK TTPs were used as the ground truth for evaluation.
2) For each case, we constructed a DFD of the target system and analyzed it with the SecOpsTM tool to identify threats. SecOpsTM was selected as a comparative baseline because it similarly derives MITRE ATT&CK-based TTPs from a DFD [41].
3) Since our tool identifies SPARTA TTPs while SecOpsTM identifies MITRE ATT&CK TTPs, we converted our tool's results into ATT&CK TTPs using established mappings between the two frameworks. This conversion was necessary to enable a direct, consistent comparison.

C. Overall evaluation results

Using the evaluation dataset, we conducted threat modeling with both our enhanced MS-TMT tool and SecOpsTM. The threats identified by each tool were then compared against the ground truth to quantitatively assess accuracy. Across all four attack cases, our tool demonstrated significantly higher accuracy than SecOpsTM as shown in table 3.

TABLE III: Results of evaluation

Case No.	**Accuracy rate of identified threat**	
	SecOpsTM	**Ours**
[28]	26%	100%
[38]	66%	100%
[39]	64%	100%
[40]	58%	100%
Total	**54%**	**100%**

For example, in the Viasat hacking incident, experts identified 27 security threats. Our tool successfully identified all of them, while SecOpsTM identified only 7 threats. In table 4, a detailed comparison of identified threats for the Viasat case shows this contrast clearly.

TABLE IV: Comparison of identified threats for the Viasat hacking

MITRE ATT&CK Technique	**SecOps TM**	**Our Template**
[T1595] Active Scanning	X	O
[T1593] Search Open Websites/Domains	X	O
[T1589] Gather Victim Identity Info.	O	O
[T1650] Acquire Access	X	O
[T1586] Compromise Accounts	X	O
[T1588] Obtain Capabilities	X	O
[T1190] Exploit Public-Facing App.	X	O
[T1133] External Remote Services	X	O
[T1078] Valid Accounts	X	O
[T1195] Supply Chain Compromise	O	O
[T1059] Command and Scripting Interpreter	X	O
[T1072] Software Deployment Tools	O	O
[T1542] Pre-OS Boot	O	O
[T1068] Exploit for Privilege Escalation	X	O
[T1562] Impair Defenses	O	O
[T1070] Indicator Removal	O	O
[T1049] System Network Connections Disc.	X	O
[T1082] System Information Discovery	X	O
[T1021] Remote Services	X	O
[T1570] Lateral Tool Transfer	X	O
[T1529] System Shutdown/Reboot	X	O
[T1485] Data Destruction	X	O
[T1495] Firmware Corruption	O	O
[T1561] Disk Wipe	X	O
[T1531] Account Access Removal	X	O
[T1498] Network Denial of Service	X	O
[T1489] Service Stop	X	O

Beyond quantitative accuracy, qualitative differences also emerged. Our tool not only matched the ground truth but also provided precise contextual information about where each threat could manifest—for example, the specific system component or data flow affected. In contrast, SecOpsTM produced several incorrect results, such as indicating that space-specific threats could occur within ground-segment components, and failed to detect most threats unique to the space domain.

D. Limitations and future work

Because detailed architectural information was not publicly available for most incidents, we modeled the target systems using a generalized space-system architecture represented as a DFD, except in the Viasat case. As a result, both tools may have identified threats that might not arise in the actual system architectures, since the generalized model includes more components and data flows than some real systems.

Although our evaluation demonstrates strong coverage of space-relevant threats, we acknowledge that not all space-related threats are covered by our template. The uncovered threats were mostly extremely specific ground-segment misconfigurations that fall outside the SPARTA framework's space-focused scope. Our template also focuses on cyber threats representable within a DFD-based model; thus, physical threats—such as kinetic anti-satellite (ASAT) attacks—cannot be meaningfully modeled.

For future work, we plan to incorporate relevant parts of MITRE ATT&CK for Enterprise to achieve more comprehensive coverage of ground-segment threats while retaining our focus on the space domain. This integration will enable the identification of threats across a broader range of components within space systems.

Additionally, emerging advances in artificial intelligence present promising opportunities for enhancing the threat modeling process. Recent studies have demonstrated the potential of Large Language Models (LLMs) in automating traditional threat modeling tasks. Elsharef et al. proposed an LLM-based system that automatically extracts design characteristics from documentation and answers threat modeling questions, achieving over 75% accuracy in meeting human evaluation expectations [44]. Wu et al. introduced ThreatModeling-LLM for banking systems, demonstrating significant improvements in threat identification through prompt engineering and fine-tuning techniques, with accuracy increasing from 0.36 to 0.69 on the Llama-3.1-8B model [45].

While no studies have yet applied AI or LLMs to space domain threat modeling, the SPARTA-MITRE ATT&CK-D3FEND integration framework established in this research provides a foundation that could benefit from such automation in future iterations. Specifically, LLMs could assist in automating stencil derivation from space system architectures and formulating threat conditions by leveraging the logical interconnections between the listed databases, potentially reducing manual effort while maintaining domain-specific rigor.

VIII. Conclusion

In this paper, we addressed the absence of automated threat modeling tools specialized to space systems. We proposed a novel domain-specific template based on the MS-TMT, establishing a structured methodology that integrates three authoritative knowledge bases—Aerospace SPARTA, MITRE ATT&CK, and D3FEND—through the logical chain: **SPARTA TTP - MITRE ATT&CK TTP(s) - D3FEND artifact(s)**. This framework enabled the definition of DFD stencils grounded in D3FEND artifacts and the systematic linkage of these stencils to SPARTA threats, allowing for domain-specific and comprehensive threat identification for space systems.

We validated the proposed template using four real-world space-related security incidents, including the Viasat hacking. The quantitative evaluation showed that our tool significantly outperformed the existing SecOpsTM tool, achieving 100% coverage of the ground truth threats, compared to SecOpsTM's average of 54%. Moreover, qualitative analysis revealed that our tool provides precise contextual information regarding where threats can manifest within the system, enabling analysts to identify potential attack vectors from the early stages of system design without prior knowledge of specific incidents.

For future work, we plan to refine and expand the coverage of threats affecting the ground segment by incorporating relevant portions of MITRE ATT&CK for Enterprise, further enhancing the completeness and applicability of our domain-specific space-system threat modeling framework.

Acknowledgment

This work was supported by Korea Research Institute for Defense Technology Planning and Advancement(KRIT) grant funded by the Korea government(Defense Acquisition Program Administration), (KRIT-CT-24-001, Defense Space Security Research Lab, 2025).

References

[1] UCS, "UCS Satellite Database", 2023, accessed: 2025-09-19. [Online]. Available: https://www.ucs.org/resources/satellite-database

[2] United Nations Office for Outer Space Affairs, (UNOOSA), "Online Index of Objects Launched into Outer Space", 2022, accessed: 2025-10-01 [Online]. Available: https://www.unoosa.org/oosa/osoindex/

[3] N. Database, "Nanosats database: The comprehensive database of nanosatellites," 2023, accessed: 2024-11-16. [Online]. Available: https://www.nanosats.eu/

[4] R. L. Staehle, B. Anderson, B. Betts, D. Blaney, C. Chow, L. Fried man, H. Hemmati, D. Jones, A. Klesh, P. Liewer et al., "Interplanetary CubeSats: Opening the Solar System to a Broad Community at Lower Cost," NTRS- NASA Technical Reports Server, 2012.

[5] Camps Carmona, A. J. (2019). Nanosatellites and applications to commercial and scientific missions.

[6] Tsitas, S. R., and J. Kingston. "6U CubeSat commercial applications.", The Aeronautical Journal, vol 116, no 1176, pp189-198, 2012

[7] Sünter, I., Slavinskis, A., Kvell, U., Vahter, A., Kuuste, H., Noorma, M., ... & Ilves, T. "Firmware updating systems for nanosatellites", IEEE Aerospace and Electronic Systems Magazine, vol 31, no 5, pp36-44, 2016

[8] Willbold, J., Sciberras, F., Strohmeier, M., & Lenders, V. "Satellite cybersecurity reconnaissance: Strategies and their real-world evaluation", IEEE Aerospace Conference, pp1-13, 2024

[9] Willbold, J., Schloegel, M., Vögele, M., Gerhardt, M., Holz, T. and Abbasi, A., "Space odyssey: An experimental software security analysis of satellites", IEEE Symposium on Security and Privacy, pp. 1-19, 2023

[10] Bailey, B., "Cybersecurity protections for spacecraft: A threat based approach", The Aerospace Corporation, 2021.

[11] Lin, P., Abney, K., DeBruhl, B., Abercromby, K., Danielson, H. and Jenkins, R., "Outer Space Cyberattacks: Generating Novel Scenarios to Avoid Surprise", arXiv preprint, 2024

[12] Pavur, J. and Martinovic, I., "Building a launchpad for satellite cyber-security research: lessons from 60 years of spaceflight", Journal of Cybersecurity, vol 8, no 1, 2022

[13] NASA, "CYBERSECURITY MANAGEMENT AND OVERSIGHT AT THE JET PROPULSION LABORATORY", 2019, accessed: 2025-10-11. [Online]. Available: https://oig.nasa.gov/wp-content/uploads/2024/02/IG-19-022.pdf

[14] White House, "Space Policy Directive – 5", 2020, accessed: 2025-10-11. [Online]. Available: https://www.cisa.gov/resources-tools/resources/space-policy-directive-5

[15] 117th Congress, "S.3511 - Satellite Cybersecurity Act", 2022, accessed: 2025-10-11. [Online]. Available: https://www.congress.gov/bill/117th-congress/senate-bill/3511/committees

[16] NIST, "Satellite Ground Segment - Applying the Cybersecurity Framework to Satellite Command and Control", 2022, accessed: 2025-10-11. [Online]. Available: https://csrc.nist.gov/pubs/ir/8401/final

[17] NIST, "Introduction to Cybersecurity for Commercial Satellite Operations", 2023, accessed: 2025-10-11. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8270.pdf

[18] Carlos Lahoz, ”STAMP and New ISO Standard for Cybersecurity”, 2020, accessed: 2025-10-11. [Online]. Available: https://psas.scripts.mit.edu/home/wp-content/uploads/2020/07/STAMPNewISOCybersecV3.pdf

[19] Carlos Lahoz, ”STAMP and ISO 20517 Cybersecurity for Space Standard”, 2024, accessed: 2025- 10-12. [Online]. Available: https://psas.scripts.mit.edu/home/wp-content/uploads/2024/2024-06-06-1130__Carlos_Lahoz__PUB.pdf

[20] Security Compass, ”Understanding the Developer-centric Threat Modeling Process”, accessed: 2025-10-27. [Online]. Available: https://www.securitycompass.com/whitepapers/understanding-the-developer-centric-threat-modeling-process/

[21] Tran, A.D., Sion, L., Yskout, K. and Joosen, W., ”TerrARA: Automated Security Threat Modeling for Infrastructure as Code”, Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy, pp. 269-280, 2024

[22] Kvam Frøseth, E., Kavallieratos, G., & Katsikas, S., ”Threat Modeling in Satellite Communications for Maritime Operations”,European Symposium on Research in Computer Security, Springer Nature Switzerland, pp. 403-424, 2024.

[23] Loren Kohnfelder and Praerit Garg, ”The threats to our products,” Microsoft Interface, Microsoft Corporation Redmond, WA, pp 33-38, 1999.

[24] Amro, A., & Kavallieratos, G., ”ThreatSpider: CTI-Driven Semi-Automated Threat Modelling for Cybersecurity Certification”, In 2025 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 619-625, 2025

[25] Microsoft, ”Threat Modeling Tool feature overview”, 2023, accessed: 2025-11-11. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-feature-overview?source=recommendations

[26] Jason Fritz, Mark Manulis, R. Harrison, Venkkatesh Sekar, ”Space Attacks Open Database Project”, accessed: 2025-10-11. [Online]. Available: https://www.spacesecurity.info/space-attacks-open-database/

[27] Kazi, A., Kazi, S. and Bhosale, S., ”Invisible Battlefields: Analyzing the Viasat Attack and its Broader Implications”, Scientific Bulletin, Vol. 30, No. 1, pp. 59-67, 2025

[28] Space Security Info, ”An analysis of the Viasat cyber attack with the MITRE ATT&CK® framework”, 2023, accessed: 2025-11-04. [Online]. Available: https://www.spacesecurity.info/an-analysis-of-the-viasat-cyber-attack-with-the-mitre-attck-framework/

[29] The AeroSpace Corporation, ”Space Attack Research & Tactic Analysis (SPARTA)”, accessed: 2025-11-04. [Online]. Available: https://sparta.aerospace.org/

[30] Salim, S., Moustafa, N., and Reisslein, M., ”Cybersecurity of Satellite Communications Systems: A Comprehensive Survey of the Space, Ground, and Links Segments”, IEEE Communications Surveys & Tutorials, Vol. 27, No. 1, pp. 372-425, 2024

[31] Castanon Remy, J. L., Ear, E., Chang, C., Feffer, A., and Xu, S., ”SoK: Space Infrastructures Vulnerabilities, Attacks and Defenses”, In Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP), pp. 1-18, 2025

[32] NASA, ”The Core Flight System (cFS)”, accessed: 2025-10-01. [Online]. Available: https://github.com/nasa/cFS

[33] NASA, ”Spacecraft Subsystems”, accessed: 2025-10-01. [Online]. Available: https://www.nasa.gov/johnson/frontdoor/capabilities/spacecraft-subsystems/

[34] NASA, ”Basics of Spaceflight”, accessed: 2025-10-01. [Online]. Available: https://science.nasa.gov/learn/basics-of-space-flight/

[35] The Consultative Committee for Space Data Systems (CCSDS), ”Reference Architecture for Space Data Systems”, Magenta Books: Recommended Practices, 2024

[36] MITRE, ”D3FEND Artifact Ontology”, accessed: 2025-10-01. [Online]. Available: https://d3fend.mitre.org/dao/

[37] Joonhyuk Park, ”MS-TMT Space Template”, accessed: 2025-11-05. [Online]. Available: https://github.com/Hanjan-inSANE/MS-TMT-Space-Template

[38] Brawner, ”DEF CON 30 - Hunting for Spacecraft Zero Days using Digital Twins”, accessed: 2025-11-05. [Online]. Available: https://www.youtube.com/watch?v=t_efCpd2PbM

[39] Brandon Bailey, ”DEF CON 28 Aerospace Village: Exploiting Spacecraft”, accessed: 2025-11-05. [Online]. Available: https://www.youtube.com/watch?v=b8QWNiqTx1c

[40] E. Ear, J. L. C. Remy, A. Feffer, and S. Xu, ”Characterizing Cyber Attacks against Space Systems with Missing Data: Framework and Case Study,” *arXiv preprint arXiv:2309.04878v1*, 2023.

[41] Ellipse2v, ”STRIDE Threat Analysis Framework with MITRE ATT&CK Integration”, accessed: 2025-11-14. [Online]. Available: https://github.com/ellipse2v/SecOpsTM

[42] Katsikeas, S., Johnsson, P., Hacks, S. and Lagerström, R.,”VehicleLang: A probabilistic modeling and simulation language for modern vehicle IT infrastructures”, Computers & Security, vol 117, p.102705, 2022

[43] Da Silva, M., Puys, M., Thevenon, P.H., Mocanu, S. and Nkawa, N., ”Automated ICS template for STRIDE Microsoft threat modeling tool”, In Proceedings of the 18th International Conference on Availability, Reliability and Security, pp. 1-7, 2023

[44] I. Elsharef, Z. Zeng, and Z. Gu, ”Facilitating Threat Modeling by Leveraging Large Language Models,” In Workshop on AI Systems with Confidential Computing (AISCC), Network and Distributed System Security (NDSS) Symposium 2024, 2024.

[45] T. Wu, S. Yang, S. Liu, D. Nguyen, S. Jang, and A. Abuadbba, ”ThreatModeling-LLM: Automating Threat Modeling using Large Language Models for Banking System,” arXiv:2411.17058, 2024.

[46] The Aerospace Corporation, ”Building Space Attack Chains using SPARTA”, accessed: 2025-11-05. [Online]. Available: https://sparta.aerospace.org/resources/OTR-2023-00989_SPARTA_DefCon2023.pdf

APPENDIX

This section introduces all stencils incorporated in our template. Both stencils and their derived stencils are denoted in boldface, with descriptions provided for each.

1) **Generic Space System Process**: A representation of a generic space system process.
   - **OS Process**: An operating system process, or system process, is a process running to perform operating system functions.
   - **Boot Loader**: A bootloader is software that is responsible for booting a computer. When a computer is turned off, its software—including operating systems, application code, and data—remains stored on non-volatile memory. When the computer is powered on, it typically does not have an operating system or its loader in random-access memory (RAM). The computer first executes a relatively small program stored in read-only memory (ROM, and later EEPROM, NOR flash) along with some needed data, to initialize RAM (especially on x86 systems) to access the nonvolatile device (usually block device, eg NAND flash) or devices from which the operating system programs and data can be loaded into RAM.
   - **Space Segment Communication Module Process**: The Communication Module is the ears and mouth for the Observatory. The system receives instructions (commands) from the space/ground segment and sends (transmits) the science and status data back.
   - **Space Segment Attitude Control Module Process**: The Attitude Control Module senses the orientation of the Observatory, maintains the Observatory in a stable orbit, and provides the coarse pointing of the Observatory to the area on the sky that the Science Instruments want to observe.

- **Space Segment Propulsion Module Process**: The Propulsion Module contains the fuel tanks and the rockets that, when directed by the Attitude Control System, are fired to maintain the orbit.
- **Space Segment Electrical Power Module Process**: The Electrical Power Module converts sunlight shining on the solar array panels into the power needed to operate the other subsystems in the bus as well as the Science Instrument Payload.
- **Space Segment RPOD Module Process**: Rendezvous, Proximity Operations, and Docking (RPOD) Module enables a spacecraft to approach and connect with another in orbit.
- **Space Segment Command and Data Handling Module Process**: The Command and Data Handling (C&DH) Module is the brain of the spacecraft bus. The system has a computer, the Command Telemetry Processor (CTP) that takes in the commands from the Communications System and directs them to the appropriate recipient. The C&DH also has the memory/data storage device for the Observatory, the Solid State Recorder (SSR). The CTP controls the interaction between the Science Instruments, the SSR and the Communications System.
- **Space Segment PNT Payload Module Process**: The PNT (Positioning, Navigation, and Timing) Payload Module is the component on a satellite responsible for broadcasting signals that allow users to determine their location and precise time.
- **Space Segment Scientific Payload Module Process**: The Scientific Payload Module enables scientific and remote sensing missions (e.g., sensors to monitor space weather, or zerogravity experiment and testing).
- **Space Segment Thermal Control Module Process**: The Thermal Control Module Process functions to keep all the spacecraft's component systems within acceptable temperature ranges during all mission phases.
- **Ground Segment Tracking and Ranging Module Process**: The Tracking and Ranging Module monitors satellites' orbits such as position, orientation, and trajectory and attains measurements of the distance and direction from the ground station to satellites and other space objects (e.g., debris).
- **Ground Segment Transmission and Reception Module**: The (Ground Segment) Transmission and Reception Module physically transmits/receives Radio Frequency (RF) signals to satellites.
- **Ground Segment Command Module Process**: The Command Module produces commands to control the space segment.
- **Ground Segment Payload Process Module Process**: The Payload Process Module processes data directly generated by the payload component of the satellite (e.g., processing images from the Earth's atmosphere taken by the payload component of the satellite).
- **Ground Segment Mission Analysis Module Process**: The Mission Analysis Module processes data related to the mission of a satellite (e.g., processing data related to the orbit of a satellite to assess whether the current orbit lets the satellite achieve the mission).
- **Ground Segment Access Terminal Module Process**: The Access Terminal Module provides operators with (remote) accesses to network enclaves in the ground segment, such as the command module. It is also in charge of managing access to the software services in the ground segment, such as flight and orbit simulation in the mission analysis module.
- **User Segment Transmission and Reception Module**: The (User Segment) Transmission and Reception Module sends/receives data from/to the user to/from a satellite or a ground relay, such as voice transmission over satellite communications.

2) **Generic Space System Data Store**: A representation of a generic space system data store.
   - **File System**: A representation of a file system.
   - **Database**: A representation of a database.
3) **Generic External Entity**: A representation of an external entity.
   - **User**: A representation of a human user.
   - **Space Environment**: A representation of space environment. This includes all non-artificial aspects of the space including space debris and solar radiation. Space segment processes may collect data from space environment using sensors for analysis.
4) **Generic Space System Data Flow**: A unidirectional representation of the flow of data between space system elements.
   - **User I/O**: A representation of human user input and output.
   - **Crosslink Communication**: A representation of transmission between space segments.
   - **Uplink or Downlink Communication**: A representation of transmission between a space segment and a ground segment.
   - **Space Environment Signals**: A representation of signals coming in from or going out to space environments.
   - **Ground to Ground Communication**: A representation of transmission between ground segments.
   - **Intra Segment Link**: A representation of communication within a single segment.
5) **Generic Trust Border Boundary**: A border representation of a trust boundary.
6) **Generic Trust Line Boundary**: An arc representation of a trust boundary.