IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Space-Forensic Tool Coverage Derivation through Satellite System Profiling

Bae Geun Cho[1], Arpita Dinesh Sarang[2], Jun-Ho Hong[3], and Ki-Woong Park*, (Member, IEEE)

[1]SysCore Lab., Sejong University, Seoul, 05006, Republic of Korea (e-mail: 07534943@hanmail.net)
[2]SysCore Lab. (Convergence Engineering for Intelligent Drone), Sejong University, Seoul, 05006, Republic of Korea
[3]Sungshin Women's University, Seoul, 02844, Republic of Korea
*Department of Information Security, Sejong University, Seoul, 05006, Republic of Korea

Corresponding author: Ki-Woong Park (e-mail: woongbak@sejong.ac.kr).

ABSTRACT With the expanding commercialization of reusable launch vehicles, ventures into the space industry appear feasible, increasing the number of satellite systems. As a consequence, based on recent incidents of ground station compromise, satellite link disruption, and terminal vulnerabilities, these systems are facing a variety of cyber threats. This highlights the necessity for a satellite forensics system to determine and prevent the cause of incidents. According to the space environment resources constraint, the conventional information technology (IT) forensics for terrestrial IT environments is nearly inapplicable in a satellite environment. Due to long propagation delays, restricted visibility, and the use of space-specific protocols like CCSDS, they employ embedded RTOS-based architectures that require real-time control. This is implemented through a multi-layered architecture. This necessitates a cross-analysis system based on flow between segments, indirect state reconstruction, signal-based restoration, and protocols. Also, a combination of specialized and general IT tools and time synchronization-based correlation analysis. Therefore, we analyzed relevant research trends, hierarchically profiled, codified the artifacts generated, and moved according to the structural and operational characteristics of the Space-Link-Ground segment. We systematically linked the functionality of satellite signal tools and IT forensics tools with profiling code to propose a forensic tool coverage model for each artifact in each segment. We constructed and simulated scenarios to examine applicability and procedural validity. This significantly improves the visibility of evidence acquisition and appropriate tool selection in satellite systems. Furthermore, forensic procedures allow reconstruction of a spacecraft's internal states to counter security threats.

INDEX TERMS Cyber threat analysis, forensic tool coverage, satellite digital forensics, segment-based forensic structuring, space infrastructure security, space cybersecurity.

I. INTRODUCTION

The space industry has rapidly expanded from a government-aided industry to a non-public one, with reduced challenges facilitated by SpaceX's reusable launch vehicle technology [1]. The proliferation of micro-satellites and constellations established a new era of "new space," and satellite technology has revolutionized terrestrial communications and information transmission as a core foundation for the global economy and security. Whereas at the same time, satellite systems are becoming targets of cyberattacks. The 2007–2008 Landsat-7/Terra ground station breach [2], the 2022 Viasat KA-SAT ground infrastructure breach [3], the Starlink terminal security vulnerability study [4] around the same time, and the Iridium communications satellite vulnerability report [5] highlighted the vulnerability of the entire space-ground-link section and the need for digital forensics. In addition, Salim et al. [6] emphasized that it is challenging to distinguish between cyberattacks and hardware malfunctions based on ground observations alone, and preemptive forensic audit capabilities must be established and maintained even before orbital deployment due to the complexity of space-related responses. Ultimately, to minimize the destructive impact of a satellite system breach or internal error, systematic forensics is essential for timely identification of the incident cause and preventing recurrence.

Related research has focused on Unmanned Aerial Vehicle (UAV) forensics [7] [9] [10] [11] and attacks targeting satellite systems [6] [8]. However, from a practical response perspective, an operationally oriented reference model that systematically connects segment-specific (space-link-ground) forensic targets and competent applicable tools. This study aims to address this gap by presenting foren-

sic tool coverage across satellite systems. Specifically, the Space–Link–Ground segment is outlined top-down, with artifacts moving among the system's sub-elements bridged and codified through hierarchical profiling. Qualified tools are then mapped to each code. Furthermore, coverage of tools and analysis targets is established, providing a reference model that allows for immediate identification of which tools to use for each sub-analysis target in the event of an incident. Furthermore, the space segment is described with a conceptual satellite background study. As a consequence, the satellite field is faced with constraints such as the special nature of the physically difficult-to-access space environment, various Real-Time Operating System (RTOS)-based embedded CPUs, limited storage space and power, high latency, and distributed ground stations, and dedicated protocols based on the Consultative Committee for Space Data Systems (CCSDS). Accordingly, the types and methods of evidence acquisition differ eventually from traditional IT, and the absence of a standardized forensic tool coverage model further complicates evidence acquisition and tool selection. This study clarifies the necessity and complexity of satellite forensics and proposes a coverage estimation method based on satellite system profiling. The proposed forensic tool coverage is analyzed and applied through scenarios from the perspectives of both operators and investigators by discussing the applicability and validity of the proposed model. The objectives of this study are as follows.

1) Firstly, by profiling the structural and operational characteristics of the satellite system, we observe process for artifacts formulation and transfer by developing collection and analysis path into code.
2) Secondly, we systematically map the functions of representative forensic tools to the preceding profiling code, based on their intended purpose.
3) Thirdly, we systematically map the functions of representative forensic tools to the profiling code, proposing a forensic tool coverage model for artifacts by segment.
4) Lastly, we examined the applied forensic tool coverage through two scenarios: a Viasat ground station intrusion incident and a hypothetical scenario. From the perspectives of operators and investigators, we examined the response and verified the proposed method's performance based on real-world incidents and experimental simulation.

The paper is organized as follows. Section II reviews related research, whereas Section III presents the profiling and code structure for each segment. Section IV presents the current status of the tools and estimates their coverage. Section V conducts performance verification based on scenarios and real-world incidents. Finally, Chapter VI outlines the conclusions of this study.

II. RELATED WORK

By examining satellite and satellite communications security and attack analysis research alongside drone forensics research, we summarize common system profiling, artifact, and tool perspectives in both fields. Based on this, we discuss the requisite of this study, "Estimating Forensic Tool Coverage (FTC) through Satellite System Profiling," and its differences from existing research.

A. SATELLITE SYSTEM THREAT ANALYSIS

Representative studies on satellite system threat analysis include Salim et al. [6] (2025), which comprehensively surveyed the space, ground, and links segments. Peled et al. [8] (2023) and [6] evaluated the security of satellite systems. These studies systematically classified security threats and attack vectors for each segment, categorizing them into Space, Link, and Ground segments, based on Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE). Furthermore, they highlighted the importance of cross-segment chain attacks and supply chain attacks, presenting a comprehensive threat landscape for satellite security. However, these studies primarily focus on describing the threats that exist and lack a structured model for the types of artifacts, analysis procedures, or forensic tools that can be obtained in the event of an actual incident. [6] evaluated and [8] mapped the Viasat KA-SAT breach case and the Instrument Conditions Awareness Recognition and Understanding System (ICARUS) hypothetical scenario to the MITRE ATT&CK framework, analyzing the tactics, techniques, and procedures (TTPs) utilised in satellite attacks from the attacker's perspective. This study organizes satellite security research using a Study of Knowledge (SoK) approach, providing valuable information for understanding the attack spectrum. However, it fails to address actual incident response or forensic analysis procedures. In other words, it focuses on identifying security threats and attack techniques, and does not offer a forensic approach for understanding the state of data, and its analysis [8].

In contrast, this study investigates the threat-centric attack perspective of previous research and focuses on building an investigator and operator-centric forensic analysis model. This study profiles and encodes state data, commands, RF signals, and operational logs generated and transferred within a satellite system based on the Space–Link–Ground segment, thereby structurally representing the flow of artifacts. Furthermore, we proposed a segment-specific tool coverage model by connecting these code structures to practically usable forensic tools such as SatDump, gr-satellites, GNU Radio, Wireshark, and Zeek. Furthermore, we specified response procedures from both operator and investigator perspectives through reproducible scenarios in a practical environment.

In summary, while both prior studies make significant contributions to defining and categorizing security threats and attack techniques for satellite systems, they fail to present practical forensic procedures or tool application scopes. In contrast, this study provides a viable forensic framework that integrates segment-based artifact flow, tool mapping, and response procedures by exploring the shortcomings of prior studies and suggesting the potential for formalizing satellite forensics Table I.

TABLE I. Comparison of Satellite System Threat Analysis Studies

Division	Salim et al. (2025)	Peled et al. (2023)	Our Study
Study Purpose	Classification of threat factors and attack vectors by satellite system segment	Real-world incident-based attack mapping and attack TTP analysis	Establishing satellite system segment-based forensic profiling and tool coverage.
Approach	STRIDE-based threat classification, supply chain and chain attack analysis	Mapping Viasat and ICARUS Cases to MITRE ATT&CK	Top-Down Analysis Focused on Artifact Flow
Point of view	Focused on security analysis and threat models	Attacker Perspective	Operator and Investigator Perspectives
Analysis target	Vulnerability in the Space/Link/Ground segments	Attack Scenarios, TTPs, and Supply Chain Vulnerabilities	Segment-specific artifact and tool mapping
Result	Threat model, defining attack potential	Attack development/pattern	Codebook, tool coverage, and response scenarios
Tool perspective	No tool	No tools or forensic procedures	Clarifying the scope of tool use
Segment Linkage	Emphasizes the possibility of chain attacks	Attack flow description	Space–Link–Ground Correlation Analysis
Practical applicability	Low forensic utility	Lack of response procedures	Real response possible
Differences from this study	Threat definition-centric, no forensic model	Attack technology classification focus	Presentation of a practical forensic structure

B. DRONE FORENSICS

Previous research in the field of drone forensics has developed based on the premise that ground-based devices such as the aircraft, controller, mobile application, and sensors can be physically accessed.

Alotaibi et al. (2022) implemented comprehensive collection and analysis model for the drone forensics field. Formalization the collection, preservation, analysis, and post-investigation procedures of drone forensics (CCAFM-Comprehensive Collection and Analysis Forensic Model) and proposed a forensic workflow based on multiple artifacts such as drone bodies, apps, and sensors [7]. Furthermore, Al-Dhaqm et al. (2021) analyzed research from 2000 to 2020 and identified structural limitations in drone forensics, including a lack of standardization, model and firmware dependencies, and log format diversity. This study pointed out that drone forensics suffers from low reproducibility and a lack of consistent models due to analysis variations across models [9]. Bouafif et al. (2018) discussed challenges and new insights obtained on file system data through File Transfer Protocol (FTP) and serial access for real aircraft such as Parrot Augmented Reality (AR), Drone, and Bebop 2, and analyzed wireless signals and app logs to perform flight recovery and correlation analysis of pilot behavior [10]. Iqbal et al. (2019) also presented an experimental forensic procedure for analyzing aircraft-pilot-app correlations based on app logs, sensor data, and aircraft memory data [11]. While these empirical studies offer analysis methods optimized for specific aircraft or platforms, they also face limitations, such as variations in the scope and structure of the analysis due to differences in aircraft model and firmware. In contrast, this study, unlike drone forensics, reflects the characteristics of satellites operating in an orbital environment where physical access is impossible. Therefore, it structures the artifact flow based on the Space–Link–Ground segment and extends this to data generated throughout the entire satellite operation cycle, including Telemetry (TM), Tele Command (TC), Radio Frequency (RF) signals, and ground station operation logs. In particular, satellites are comprised of RTOS-based embedded systems and long-distance RF links, making direct extraction (FTP/Serial/JTAG) methods employed in drone forensics impossible. Accordingly, this study proposed an indirect state reconstruction model including RF-based TM restoration, TC packet analysis, and ground station log/DB-based correlation analysis. Furthermore, this study clearly demonstrates segment-based forensic tool coverage by linking satellite signal analysis tools such as SatDump, gr-satellites, GNU Radio, Wireshark, and Zeek, as well as existing IT forensics tools, with profile codes. Furthermore, by constructing scenarios and proposing reproducible satellite forensics procedures that can be performed from the perspectives of both operators and investigators, this study fundamentally differentiates itself from prior drone forensics research.

In summary, previous research on drone forensics has been based on the analysis of logs, sensors, and apps centered on physically accessible aircraft, and has limitations in standardization due to variations across aircraft types. This study, however, reflects the constraints of a satellite environment where physical access is impossible, constructs a segment-based, flow-oriented space forensics model, and proposes a coverage structure that integrates artifacts, tools, and procedures, thereby establishing a new forensic analysis paradigm.

III. SATELLITE SYSTEM PROFILING

A. ENVIRONMENTAL CHARACTERISTICS OF SEGMENTS

The space segment must perform in extreme environmental conditions. They include vacuum, cosmic radiation, microgravity, and thermal oscillations between -100°C to 100°C. These requirements have to include radiation-hard parts, operational spares, and error detection systems, and a shortage of solar-powered energy restricts system operations and communication time. Such conditions are huge in the way of data integrity maintenance [12] [13]. The atmospheric propagation properties and legal regulations control the link segment. The Kurz-under (Ku)/Kurz-above (Ka) bands experience precipitation attenuation, and the low-frequency bands experience ionospheric effects and multipath fading, which creates uncertainty in link stability. In addition, real-time forensic data collection and analysis are complicated due to encrypted transmissions within limited bandwidth and prop-

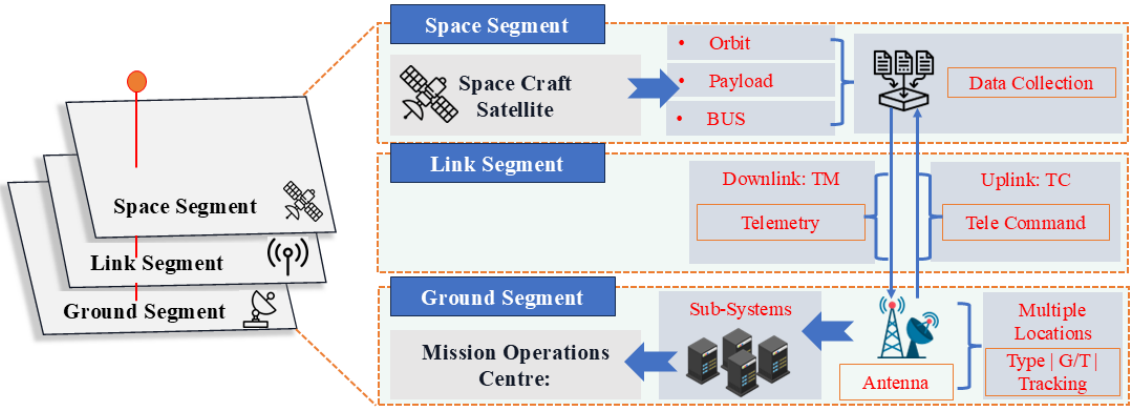

FIGURE 1. Satellite system structure.

agation delays [12] [13]. The ground segment consists of antennas, radio-frequency equipment, networks, and control systems that have been installed in various locations. Ground infrastructure is the operational link to external environments, which renders it a readily available attack surface that must be protected with comprehensive security measures and regular audits according to [12] [13].

B. SATELLITE SYSTEM STRUCTURE

The overall structure of the satellite system and its internal segments, considering environmental characteristics into account, are configured as shown in Fig. 1.

A detailed look at the structure of each segment is as follows:

1) Space Segment

Primarily, this study focuses on the space segment, targeting satellites. Fig. 2 illustrates the satellite's structural hierarchy. Communication within the satellite itself utilizes Controller Area Network (CAN), RS422, SpaceWire, and MIL-STD-1553B. The Operating System (OS), On-Board Computer (OBC) processor, memory, and storage vary in size, purpose, and other characteristics for each satellite. For optical satellites, the overall specifications are as follows:

The satellite's structure is divided into a payload (communications, optical, Synthetic Aperture Radar (SAR), etc.) according to the satellite's purpose and a BUS that handles the satellite's functions. It consists of a command and Data Handling System (CDHS), an Attitude and Orbit Control System (AOCS), an Electrical Power System (EPS), a Communication System (COMS), a Structure System (STS), a Propulsion System (PS), and each subsystem. Artifacts that can be verified based on this structure include status and event logs, satellite status information through TM, and file storage (dump files, camera images) [14].

2) Link Segment

The link segment located between the space segment and the ground station segment is broadly divided into TC for uplink from the ground to the satellite and TM for downlink from the satellite to the ground. Details are designed based on the protocol specified in CCSDS [14]. Radio waves mainly used for communication can be distinguished as VHF (Very High Frequency), UHF (Ultra-High Frequency), L/S/C/X/KU/Ka-band [12] [13].

Artifacts that can be identified include RF signals, network propagation signals (I/Q capture), and network packets.

3) Ground Segment

The ground station segment is an area that encompasses each system, including the antenna system and operating system, to the user's terminal, and its structural layer is as shown in Fig. 3.

The ground station structure consists of the Mission Operation Center (MOC), which is the largest flow, and the following subsystems exist under it: the Satellite Operating Subsystem (SOS), which transmits satellite commands and verifies received data; the Ground Station and antenna RF system, which transmit and receive data from the Ground Station Operation SOS to the satellite. The Ground Station may also use antennas located in various locations around the world. Depending on the scale of the satellite system, it is also configured within the SOS within the MOC. It consists of the Mission Planning Subsystem (MPS), which handles satellite scheduling and commands; the Flight Dynamics Subsystem (FDS), which predicts and determines the satellite's orbit; and the Image Reacquisition (IR), which handles image reception and processing. Finally, data is provided to users via the web or API [16]. Artifacts that can be verified through the ground station structure are generally RF information of the antenna, logs of the control and operation system (system, network, dedicated SW), database, user account and permission, storage, etc., in a general IT environment.

C. PROFILING-BASED CODING

The segments within each layer discussed above can be constitutionally connected according to the flow of artifacts, encompassing the entirety. Therefore, this study uses the Space–Link–Ground segment as the unit of analysis, but assumes that artifacts created and transferred in actual operations are not fixed to specific segments but intersect along

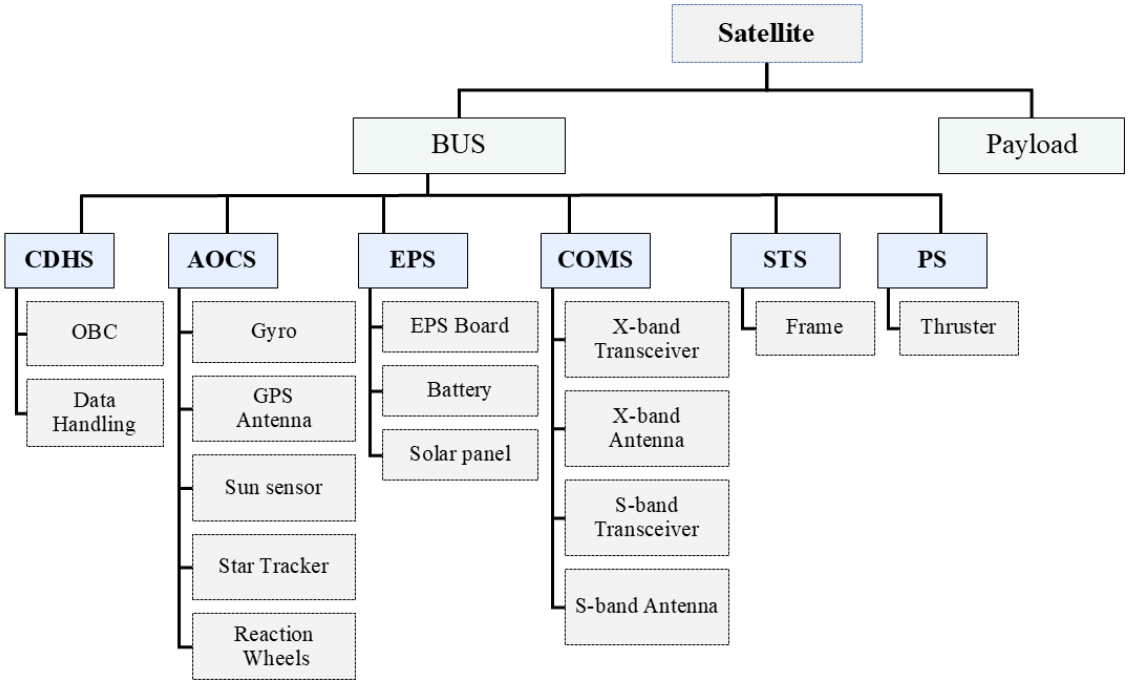

FIGURE 2. **Satellite structure. (this figure is taken from [14]).**

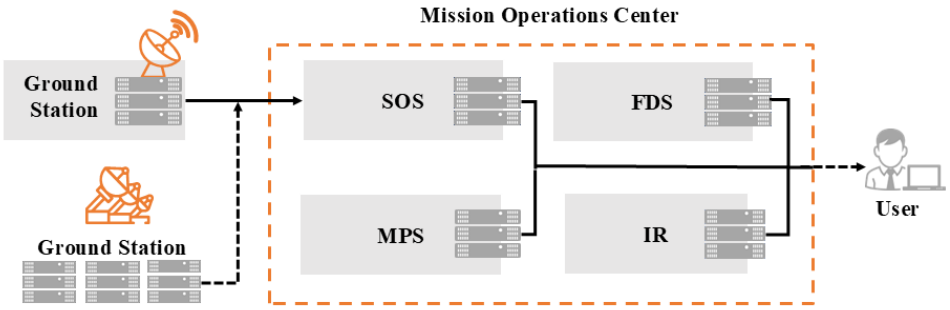

FIGURE 3. **Ground station segment structure.**

process paths. In other words, segments are not fixed but flow-oriented, intersecting layers, and subsequent coding and coverage analysis will follow this collaborative perspective. A diagram illustrating this is shown in Fig. 1. Therefore, we propose coding for profiling as shown in Fig. 4.

By reviewing the code, we determine which segments are being analyzed, which systems are involved, and what data flows are involved. To determine which interfaces will be utilised to analyze the files, we established the following coding rules. To facilitate the identification of which forensic tools can be used to analyze the profiling code files, we coded them as follows:

If you check Fig. 4, the code structure is composed of "Segment – Domain – Subsystem – Flow – Interface – Format _ Forensic Tool". Each identifier is separated by "-", and if there is an additional argument, it is added using a semicolon ";". Lastly, the forensic tool area clearly distinguishes the identifier and the forensic tool using "_". Also, in case of abbreviations, they are indicated as Downlink (DL), Uplink (UL), Packet (Pkt), Frame (Frm), SpaceWire (SpW), Transceiver (CVR), etc. For example, we're using the forensic tool Satdump to recover TM packets from the DL of the satellite OBC using S-band RF signals. This example demonstrates how to recover TM data from an RF-based location outside the ground station, thereby identifying whether the cause of TM failure is ground station failure or satellite malfunction. If we represent the cases of each segment of Space-Link-Ground presented above according to the coding rules, the Space Segment, the Link Segment, and the Ground Segment are as shown in Table. II.

The profile code proposed in this study is a core component for formalizing satellite forensic procedures. It is a representational system that rigorously describes the analysis target by combining segments (Space-Link-Ground), domains, subsys-

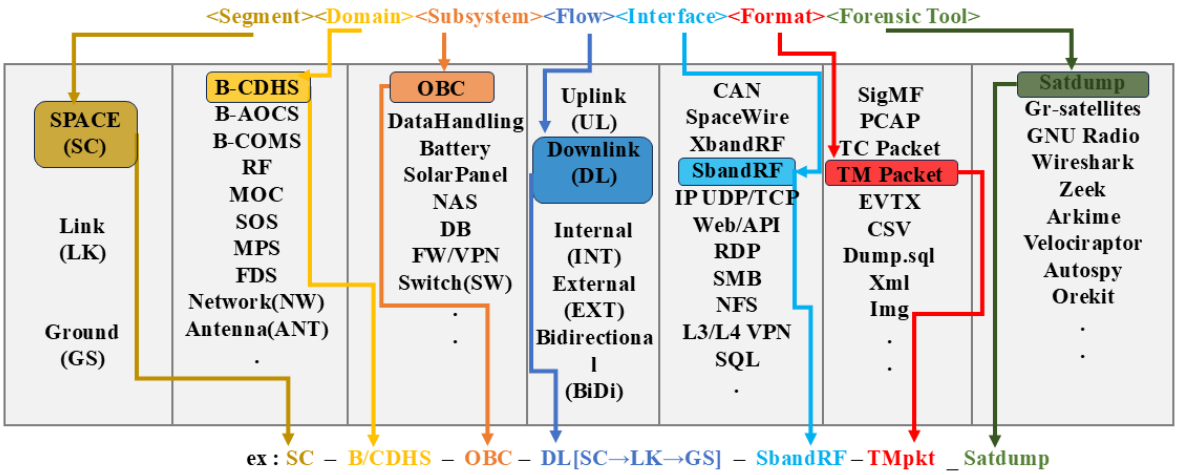

FIGURE 4. **Profiling Coding Rules.**

TABLE II. **Space segment Satellite System Profiling and Forensic Tool Combination Code**

Satellite System Profile Code	Forensic tools	Artifact
SC-B/CDHS-OBC-DL [SC→LK→GS]-SbandRF/CCSDS-TMpkt	gr-satellites, Kaitai Struct, Wireshark(CCSDS dissector)	TM packet
SC-B/CDHS-OBC-DL[SC→LK→GS]-SbandRF /CCSDS-TMpkt_gr-satellites;Kaitai Struct;Wireshark		
SC-B/CDHS-OBC-UL[GS→LK→SC]-SbandRF/CCSDS-TCpkt	gr-satellites, Kaitai Struct	TC packet
SC-B/CDHS-OBC-UL[GS→LK→SC]-SbandRF /CCSDS-TCpkt_gr-satellites;Kaitai Struct		
SC-B/AOCS-RW-DL[SC→LK→GS]-SbandRF/CCSDS-TMpkt	gr-satellites, Kaitai Struct, Wireshark(CCSDS dissector)	TM packet
SC-B/AOCS-RW-DL[SC→LK→GS]-SbandRF/CCSDS-TMpkt_gr-satellites;Kaitai Struct;Wireshark		

TABLE III. **Link and Ground segments Satellite System Profiling and Forensic Tool Combination Code**

Satellite System Profile Code	Forensic tools	Artifact
LK-RF-Sband-UL[GS→LK→SC]-SbandRF-SigMF	SatDump, GNU Radio, inspectrum	SigMF
LK-RF-Sband-UL[GS→LK→SC]-SbandRF-SigMF_SatDump; GNURadio; inspectrum		
LK–NW–FW/VPN–BiDi[GSLK] –L3/L4/VPN–PCAP	Wireshark, Zeek, Arkime	PCAP
LK-NW-FW/VPNBiDi[GS-LK]-L3/L4/VPNPCAP_Wireshark; Zeek;Arkime		
GS-MOC-MPS-UL[GS→LK→SC]-Web/API-SchPlan.xml	Pandas, Kibana	Schedule Plan.xml
GS-MOC-MPS-UL[GS→LK→SC]-Web/API-SchPlan.xml_Pandas;Kibana		
GS-MOC-DB-INT[GS]-SQL/PostgreSQL-dump.sql	PostgreSQL, SQLite(sqlite3)	dump.sql
GS-MOC-DB-INT[GS]-SQL/PostgreSQL-dump.sql_PostgreSQL; SQLite		

tems, data flows (UL/DL/INT), interfaces (RF/API/DB), and data formats (CCSDS/XML/SQL, etc.). This detailed code structurally represents the process by which satellite system artifacts are created and moved, and is utilized by forensic investigators to define the scope of analysis and precisely reconstruct evidence. However, in real-world operational environments, operators, not forensic experts, are often the first to recognize anomalies. It's practically impossible to immediately interpret detailed codes and determine a response. To address these limitations from an operator perspective, this study introduced short codes for operators in parallel with detailed codes. Short codes are a concise structure that extracts only three elements: segment (SC/LK/GS), data type (TM/TC/RF/LOG), and data flow (UL/DL/INT). For example, a downlink telemetry anomaly in the Space segment would be expressed as "SC-DL-TM," an uplink RF anomaly in the Link segment as "LK-UL-RF," and a suspected schedule tampering at the Ground would be expressed as "GS-INT-SCHED." When an anomaly occurs, operators can intuitively identify which segment and which information was affected through the short code, enabling them to immediately initiate a response without having to understand complex analysis structures.

In contrast, detailed codes are primarily used by forensic investigators or security analysts. Investigators, based on the problem areas identified by the short codes, reference the detailed codes to specify the scope of the analysis and the evidence path. They utilize the segment-subsystem-interface information contained in the codes to systematically determine which artifacts should be analyzed using which tools. Furthermore, the detailed code plays a crucial role for engineers and automation specialists. It defines segments, data flows, and interfaces in a consistent structure, enabling it to serve as a formalized identifier for implementing future systemic integrations, such as forensic automation scripts, Security Information and Event Management (SIEM) integration, and

RF data collection pipelines. Thus, the code system of this study is not a single notation. Rather, it is a multi-layered utilization structure where operators use short codes to "quickly identify anomalies," investigators use detailed codes to "precisely structure the scope of analysis and evidence flow," and engineers use detailed codes to "automate and link tools." In addition to detailed code, the FTC system links this code-based forensic analysis to a practical, executable procedure. Code is a formalized address system ("What to Investigate") that defines the analysis target and scope, while tool coverage is a formalized execution system ("How to Investigate") that reaches that address and performs the analysis. By combining the two systems, operators can quickly identify response paths through short codes, investigators can select the right tools based on detailed codes and coverage information to perform consistent analysis procedures, and engineers can invoke analysis automation tools and script collection based on detailed codes.

Therefore, the combination of profile code and tool coverage ensures consistency across user layers in the formality and reproducibility of satellite forensic analysis, which serves as the basis for significantly improving the practical and technical response capabilities required in a satellite operating environment.

IV. DIGITAL FORENSIC TOOLS AND COVERAGE

A. DIGITAL FORENSIC TOOLS

1) Space Segment Forensic Target and Analysis System: In the space segment, telemetry-based artifacts such as state data, commands, and RF signals generated within the satellite are central to analysis. As satellites are physically inaccessible, direct acquisition of onboard data, such as memory and storage, is impossible. Consequently, indirect state recovery through TM/TC and RF signals becomes central to forensic analysis. This study profiled CCSDS-based TM packets generated from satellite subsystems (OBC, AOCS, EPS, etc.) and terrestrial TC packets, structuring them into a profile code and mapping them to tools such as gr-satellites, Kaitai Struct, and Wireshark (CCSDS dissector) to reconstruct state changes, command validity, and abnormal events. Furthermore, by analyzing S-band-based DL RF signals (IQ/SigMF) and AX.25-based beacon packets using SatDump, GNU Radio, and inspectrum, it was possible to identify the presence of transmissions, the possibility of RF attacks, and the causes of link disconnections. This artifact-to-tool mapping system provides an effective basis for reconstructing satellite status in real-time or retrospectively in the orbital environment.
2) Link Segment Forensic Target and Analysis System: Link segments are RF and communication sections connecting space and ground. As TM/TC packets are modulated and transmitted in RF form, they are prone to generating significant artifacts from a forensic perspective. This study profiled RF signals (SigMF/IQ) captured from the S-band uplink and Ka-band/S-band downlink, CCSDS TM/TC frames acquired through RF demodulation, and FW/VPN (Virtual Private Network)-based network traffic between the ground station and the MOC. RF-based analysis tools such as SatDump, GNU Radio, and inspectrum can be used to identify transmissions, modulation anomalies, and regenerated signals at the physical layer. Tools like Wireshark, Zeek, and Arkime are used to analyze unauthorized access, API call patterns, and command transmission flows at the L3/L4 network layer. Furthermore, comparative analysis of FW/VPN settings using RANCID and Oxidized can be used to identify unauthorized rules inserted by attackers into device configurations. This multi-layered analysis is essential for uncovering the true nature of link-based attacks, including satellite system disruption, spoofing, and jamming.
3) Ground Segment Forensic Target and Analysis System: The ground segment, where all information related to satellite operations is stored and processed, generates the most diverse forms of forensic artifacts. This study conducted profiling based on data available in real-world operating environments, including the MPS, ground station operation logs, time synchronization Network Time Protocol (NTP) logs, TM/TC storage databases, server and workstation OS logs, Network-Attached Storage (NAS) images, and network equipment configurations. Schedule and log visualizations using pandas and Kibana are used to track command tampering and Application Programming Interface (API) call patterns, while PostgreSQL and SQLite-based database analysis serve as the basis for recovering command-state correlations. Furthermore, server and terminal forensics using Kroll Artifact Parser and Extractor (KAPE), Velociraptor, and Autopsy contribute to identifying ground-level compromise traces, such as account hijacking, script execution, and malicious activity. RANCID and Oxidized detect ground-level infrastructure configuration changes, enabling the identification of settings manipulated by attackers to secure intrusion paths. In this way, the Ground Segment plays a crucial role in identifying the full-cycle cause of an accident through cross-referencing with data from the Space and Link segments.

   The following is a summary of the target readers, functions, analysis target extensions and outputs, tool names, and descriptions, organized through the profiling process described above.

The tools are categorized by their intended use: network forensics, log analysis, storage and firmware analysis, memory and endpoint forensics, binary analysis, satellite orbit and radio analysis, data analysis and visualization with version management, and vulnerability scanning. This classification clearly defines the functional role of each tool and serves as a basis for understanding the order and method of tool

integration for each analysis objective. The integration of tools through each part of the analysis pipeline is structured as follows. The digital forensics process is a hierarchical process that utilizes multiple tools sequentially based on various data sources and analysis objectives. From the collection stage to structural analysis, behavioral analysis, and integrated visualization, tools interact and complement each other throughout the process.

In network forensics, raw packets are first collected from network interfaces using tcpdump. These packets are then analyzed by the protocol unit in Wireshark or tshark, depending on the analysis purpose, or utilized in Scapy for packet reconstruction and modulation. When tracking long-term communication flows or identifying specific sessions is required, Packet Capture (PCAP) data is loaded into Arkime for session-based exploration. When behavior-based event extraction is required, communication logs are generated using Zeek. When the possibility of tampering with network equipment settings is the subject of analysis, RANCID and Oxidized collect equipment configuration information and manage its version with git, enabling analysis of the correlation between equipment settings and traffic changes.

The forensic analysis pipelines are designed specifically to be applied in cyber incident analysis. For log analysis, the tools applied are journalctl on the Linux platform and Windows Event Viewer on the Windows operating system to collect data, Elasticsearch to index, and Kibana to visualize the time-based patterns. In storage and firmware analysis, storage acquisition is performed with dd or Guymager, file system analysis is achieved using Autopsy and The Sleuth Kit, and the extraction and reconstruction of embedded firmware are done using binwalk, unsquashfs, and Firmware-Mod-Kit. In memory and endpoint forensics, artifacts are collected through KAPE and Velociraptor, and deep memory analysis is performed utilizing Volatility3. Binary analysis starts with tools such as strings and xxd, proceeds to structural analysis with Kaitai Struct, and another step using Ghidra, Interactive Disassembler (IDA), or Radare2. RF/I/Q information is derived through GQRX or Software-defined Radio (SDR)# for satellite signal and orbit analysis, and then implemented through GNU Radio, decoded by SatDump or gr-satellites, and orbit computations are carried out with tools such as pyorbital, GPredict, or Orekit. Results in all areas are aggregated inducing pandas, displayed using matplotlib, and version-controlled with Git to provide reproducibility. Also, Trivy is used to scan vulnerabilities of containerized environments.

B. PROFILING-BASED FORENSIC TOOL COVERAGE

The coverage of the analysis target and digital forensics tool combined through satellite system profiling in Chapter III. Table. IV presents a coverage matrix that allows practitioners to quickly identify which tools are available for analyzing each satellite system component. The criteria for judging "●" (analysis possible) and "○" (analysis impossible) in the coverage matrix are as follows. "●" means that the tool can directly parse the file format of the artifact, decode the protocol, or structurally analyze the data. On the other hand, "○" indicates that the tool cannot process the artifact natively or requires a separate plugin or conversion process. For example, Wireshark is marked with "●" for PCAP files and CCSDS packets, as it can analyze them directly through its built-in dissector. On the other hand, SigMF format is marked with "○", as Wireshark cannot directly process raw RF signal data. In order to analyze the items extracted from the satellite system profiling code presented in Chapter III, they were coded with forensic tools and organized as in Table. II and III.

By adding corresponding forensic tool codes based on the profile codes for the Space, Link, and Ground segments in the table, we can systematically identify what evidence can be obtained from the profiled code and where gaps in analysis exist. This is significant due to the same encoding scheme can be applied even if the satellite system structure or forensic environment changes, allowing for easy update of the coverage analysis model. The forensic tool coverage matrix for each segment is organized by segment in the Appendix A, B, and C.

V. SYSTEM INVASION SCENARIOS

The scenarios covered in this chapter are "1. Viasat KA-SAT Ground Station Intrusion Scenario" and "2. Hypothetical Scenario." These scenarios utilize the previously presented profiling and coverage matrix for tools to simulate potential intrusions and responses during actual satellite operations. The scenarios are presented from the perspectives of both the operator and the investigator.

A. VIASAT KA-SAT GROUND STATION INTRUSION SCENARIO

1) operator point of view

: The Network Operations Center (NOC) operator is responsible for KA-SAT satellite internet services in Europe. Where the incident occurred when the operator was handling the daily availability report [17] [18] [19].

1) First clue – modem OFFLINE alarm and shortcode note: While monitoring the dashboard, the operator noticed a graph showing an unusually rapid drop in the number of modem connections online in a specific beam. The link quality was normal, but the modem was repeatedly switching to OFFLINE. The operator noted the shortcode to track this situation later. (1) GS-DL-TM: Ground TM Archive (Modem Session/Status) and (2) GS-INT-DB: Operational DB (Modem/Firmware/Regional Information) The Operator concluded that the problem was more likely at the ground or terminal level than at the link level, and they expanded the Ground segment in the prepared coverage matrix. In the Ground / Operations / DB / "DB Transaction/Status" column, PostgreSQL, SQLite, and pandas were marked with ●. So, the operator decided to use PostgreSQL and pandas directly for GS-INT-DB. When the team aggregated the status by modem model, firmware, region/beam in the DB, we saw that certain

TABLE IV. SUMMARY OF FORENSIC TOOL COVERAGE BY SEGMENT

Segment	Category	Analysis Task	SatDump	gr-satellites	GNU Radio	Wireshark	Zeek	Kibana	PostgreSQL	pandas	Autopsy	KAPE	Volatility3	binwalk	Ghidra
Space	TM/TC Analysis	Telemetry packet decoding and command parsing	●	●	○	●	○	○	○	●	○	○	○	○	○
Space	Orbit Analysis	Satellite orbit and attitude calculation	●	●	○	○	○	○	○	●	○	○	○	○	○
Space	RF Signal	RF signal demodulation and analysis	●	○	●	○	○	○	○	○	○	○	○	○	○
Link	RF Processing	Signal demodulation and frame recovery	●	●	●	●	○	○	○	○	○	○	○	○	○
Link	Network Traffic	Protocol analysis and session tracking	○	○	○	●	●	○	○	○	○	○	○	○	○
Link	CCSDS Protocol	TM/TC packet and frame analysis	○	●	○	●	○	○	○	○	○	○	○	○	○
Ground	System Logs	Log collection and correlation analysis	○	○	○	○	○	●	○	●	○	○	○	○	○
Ground	Database	TM/TC archive and transaction analysis	○	○	○	○	○	○	●	●	○	○	○	○	○
Ground	Network Forensics	Traffic capture and anomaly detection	○	○	○	●	●	○	○	○	○	○	○	○	○
Ground	Disk Forensics	Storage acquisition and file system analysis	○	○	○	○	○	○	○	○	●	●	○	○	○
Ground	Memory /Binary	Memory dump and reverse engineering	○	○	○	○	○	○	○	○	○	○	●	●	●

models and certain firmware versions were all turned OFFLINE around the time of the accident.

2) Schedule and Management Commands – Review coverage and decide where to look next. "Did something go wrong during the mass firmware update, or did someone mess with the settings?" the operator wondered and decided to check the mission/management schedule and modem management log as the next step. First, the operator noted two more shortcodes: (1) GS-INT-SCHED: Mission/Provisioning Schedule and (2) GS-INT-MDM: Modem Management (TR-069/Dedicated Management) Log After debating where to start, the operator expanded the coverage matrix again and found the "Schedule/Admin Logs" column in the Ground/Operations/Servers section. Here, Kibana, Elasticsearch (log search/visualization), git (schedule file versioning), and pandas (time series aggregation) were marked with a "●". Based on this matrix, the operator planned work to use GS-INT-SCHED for version comparisons using git diff, GS-INT-MDM for searches using Kibana/Elasticsearch, and then pivot using pandas. Upon running the plan, confirmed that a large number of firmware and configuration update requests came from unfamiliar IP addresses and accounts in the short period immediately before the incident, and that this period coincided with the modem going offline.

3) Ground Network - At this point, the operator concluded that there was a high probability that the network traffic itself was abnormal. The operator wanted to view more of the ground station network segment, and in the coverage matrix, but found the Ground / Operations / Network (Packet) axis. The following shortcodes and tools were linked to this axis: GS-INT-VPN: VPN/Firewall Logs/Traffic - Tools (1): Zeek, Arkime, Wireshark/tshark, tcpdump Looking at the matrix, the operator checked the "tools that can be used immediately in our current environment." Wireshark and tcpdump were already installed by default, and Zeek and Arkime were ready for the analysis server. So, the operator decided to use Wireshark for real-time packet capture and Zeek for later session analysis. First, the operator started capturing data with tcpdump on the GS-INT-VPN section and simultaneously checked the stream in Wireshark. The operator noticed a pattern of intermittent, yet massive, HTTPS management traffic flowing in from an external IP range that was rarely seen before. Wanting to see when this pattern had started, the operator revisited the "Logs (System/App)" column in the same row of the coverage matrix. Kibana, Elasticsearch, journalctl, and Windows Event Viewer were marked with a "●". Using Kibana and Elasticsearch, which they had prepared, they checked the firewall/VPN logs in increments of 24 hours, 1 week, and 2 weeks. The results showed a concentrated increase in VPN login attempts and successful events using foreign IP addresses starting a few hours before the incident, and these sessions were creating internal connections to the MPS and modem management servers. At this point, the operator selected tools along the Ground/Operations/Network/Log/DB axes of the coverage matrix, gathering evidence centered on the GS-INT-VPN, GS-INT-MDM, GS-INT-SCHED, GS-INT-DB, and GS-DL-TM shortcodes. The operator then blocked suspicious accounts and IP addresses, disabled updates, preserved relevant logs, PCAPs, and DB/schedule dumps, and transferred the investigation to the next level.

2) Investigator's point of view

A few days later, the same person became the digital forensics investigator in charge of the post-mortem analysis of this incident. The investigator received the shortcode-based evidence bundle compiled by the operator and first opened the profile table and coverage matrix. The operator's memo listed the following shortcodes: - GS-INT-VPN - GS-INT-MDM - GS-INT-SCHED - GS-INT-DB - GS-DL-TM Referring to the profile table, each code was decoded into its full, detailed code form as follows:

1) GS-INT-VPN = GS-NW-FW/VPN-BiDi[GS-GS]-L3/L4/VPN-FWlog;set.cfg
2) GS-INT-MDM = GS-MOC-CPE-INT[GS]-TR069/Proprietary-MDM-log
3) GS-INT-SCHED = GS-MOC-MPS-UL[GS→LK→SC]-Web/API-SchPlan.xml
4) GS-INT-DB = GS-MOC-DB-INT[GS]-SQL/PostgreSQL-dump.sql
5) GS-DL-TM = GS-MOC-TM-DL[SC→LK→GS]-UDP-PCAP;CSV

If they organized these five codes into a single flow, the intrusion initiated by GS-INT-VPN leads to mass management commands through GS-INT-SCHED and GS-INT-MDM, and the result is It can be seen in GS-DL-TM and GS-INT-DB. Looking at the coverage matrix, the GS-INT-VPN row is marked with "●" for Ground / Operation / Network (Packet) and the Log column for Zeek, Arkime, RANCID, and Oxidized. Based on this information, the investigator applied a diff analysis to the GS-INT-VPN evidence, setting the session trace to Zeek/Arkime and the RANCID/Oxidized setting. The GS-INT-MDM and GS-INT-SCHED rows have Kibana, Elasticsearch, git, and pandas marked with a "●" in the Ground/Operation/Server/Log/Schedule column, so they used this combination to restore the management command and schedule change history in a timeline. The GS-DL-TM row has Wireshark/tshark, Zeek, and Arkime marked with a "●" in the Link/Downlink TM column, so they used these tools to analyze the TM/session changes at the time of the failure. For the rows, PostgreSQL, pandas, and matplotlib were used to quantify the extent of the damage. By following this sequence of short code → detailed code → coverage matrix, the investigator was able to assemble the entire data into a single timeline: "External VPN intrusion → Management server mass command issuance → Modem flash destruction → Service interruption."

B. ORBIT AND ATTITUDE CONTROL COMMAND MODULATION

1) operator point of view

The operator who controls several optical satellites in Low Earth Orbit (LEO). One day, the operator received a warning from FDS that the predicted and actual orbits were slightly misaligned. Even outside of regular maneuvering, something felt off.

1) TM Anomalies – Selecting which axis to watch from the coverage the operator decided to start by checking AOCS-related TMs and noted the following shortcodes: (1) SC-DL-TM: AOCS TM released from the satellite and (2) GS-DL-TM: Ground TM archive. The operator expanded the coverage matrix and located the Space / Bus-AOCS / Orbit and Attitude Control / TM axes. These axes were marked with "●" for SatDump, gr-satellites, Kaitai Struct, pyorbital, and Orekit. Therefore, based on the coverage, the operator decided to redecode the TM for SC-DL-TM using SatDump and gr-satellites, and then use pyorbital and Orekit for subsequent orbit/attitude calculations. The decoding results revealed AOCS-related maneuver signatures appearing at unscheduled points.
2) Mission Planner/MPS and Cloud – Ground Coverage Reference If a launch occurred but wasn't scheduled, there could be a problem with the MPS or cloud, or someone may have secretly added something. The operator added two shortcodes: (1) GS-INT-SCHED: Mission Planner Schedule/API and (2) GS-INT-DB: TM/TC and Operations DB. The operator rechecked the Ground/Operation/Server/Schedule and Ground/Operation/DB axes of the coverage matrix. The Schedule/API axis indicated git, Kibana, and pandas with a "●", while the DB axis indicated PostgreSQL, pandas, and matplotlib with a "●". Later compared the schedule file with the previous version using git diff for GS-INT-SCHED and viewed the API call log with Kibana. The operator used PostgreSQL and pandas to query the TC history to determine whether a TC related to the startup was actually issued during the problematic timeframe. While there was a record of the TC issuance, the approval flag was ambiguous, so the operator decided to check the RF and the TC itself.
3) Uplink RF/TC – Select tool by Link/Space coverage the operator added two more shortcodes: (1) LK-UL-RF: Uplink RF IQ (S-band) and (2) SC-UL-TC: TC packets actually transmitted by satellite In the coverage matrix, they located the Link / Uplink / RF IQ & TC axes, where GNU Radio, SatDump, inspectrum, gr-satellites, and Wireshark/tshark (CCSDS) were marked with a "●". Accordingly, they demodulated LK-UL-RF with GNU Radio and confirmed the signal pattern with inspectrum. Then, they parsed the demodulated data with gr-satellites, Kaitai Struct, and Wireshark (CCSDS) to determine SC-UL-TC. As a result, they confirmed that commands with TC codes and excessive ΔV parameters that were not in the schedule were indeed uplinked. They immediately switched to Safe Mode and blocked the account/transmission path. They organized the evidence around the "SC-DL-TM", "GS-DL-TM", "GS-INT-SCHED", "GS-INT-DB", "LK-UL-RF", and "SC-UL-TC" short codes and handed them over to the investigator.

2) Investigator's point of view

The forensic investigator for the same organization, began reconstructing the case, armed with the operator's shortcode bundle and coverage matrix. The operator's memo contained the following shortcodes: (1) SC-DL-TM, GS-DL-TM (2) GS-INT-SCHED, GS-INT-DB (3) LK-UL-RF, SC-UL-TC Decoding each code in the profile table into detailed codes is as follows:

1) SC-DL-TM = SC-B/AOCS-RW-DL[SC→LK→GS]-SbandRF/CCSDS-TMpkt
2) GS-DL-TM = GS-MOC-TM-DL[SC→LK→GS]-UDP-PCAP;CSV
3) GS-INT-SCHED = GS-MOC-MPS-UL[GS→LK→SC]-Web/API-SchPlan.xml
4) GS-INT-DB = GS-MOC-DB-INT[GS]-SQL/PostgreSQL-dump.sql
5) LK-UL-RF = LK-RF-Sband-UL[GS→LK→SC]- SbandRF-SigMF
6) SC-UL-TC = SC-B/CDHS-OBC-UL[GS→LK→SC]-SbandRF/CCSDS-TCpkt

By combining these into a single flow, the command generated in GS-INT-SCHED is uplinked as SC-UL-TC/LK-UL-RF, and the satellite's response appears in SC-DL-TM/GS-DL-TM, which is then ultimately stored in GS-INT-DB. Tools for each step were selected through a coverage matrix. The GS-INT-SCHED row, with git, Kibana, and pandas marked with a "•" in the Ground/Operation/Server/Schedule axis, was used for comparing schedule versions and analyzing API logs. The LK-UL-RF and SC-UL-TC rows, with GNU Radio, SatDump, gr-satellites, and Wireshark marked with a "•" in the Link/Uplink/RF IQ·TC axis, were directly applied to uplink signal demodulation and TC parsing. The SC-DL-TM and GS-DL-TM rows were used to reconstruct the TM pattern immediately after startup, as SatDump, gr-satellites, Zeek, and Wireshark were "•" on the Downlink TM axis. Finally, the GS-INT-DB row was used to quantify the predicted and actual trajectories, fuel consumption, and mission impact, as PostgreSQL, pandas, and matplotlib were "•" on the DB axis.

Starting from the short code left by the operator, the investigator followed the detailed code and coverage matrix to restore the data flow "GS-INT-SCHED → SC-UL-TC/LK-UL-RF → SC-DL-TM/GS-DL-TM → GS-INT-DB", and was able to prove that the unauthorized TC was an event that led to an actual operation and affected the mission.

In conclusion, the two scenarios specifically demonstrate how the proposed profile code and tool coverage matrix can be utilized in the satellite system intrusion incident response process. Through this, we confirmed that this model can be used in real-world operational environments.

VI. PERFORMANCE EVALUATION

A. VERIFICATION METHODOLOGY

To validate the effectiveness of the segment-based profiling code and forensic tool coverage model proposed in this study, the Viasat KA-SAT incident (2022) [3] [18] and the Landsat-7/Terra ground station intrusion incident (2007-2008) [2], which were actual satellite system intrusion incidents, were selected as analysis subjects. However, it should be noted that this study calculated the improvement effects through theoretical estimates based on publicly available incident reports, not through actual measured data in an actual operating environment. Therefore, the quantitative figures presented are not actual performance measurement results, but rather the expected improvement rate that can be achieved when the proposed model is applied.

The improvement effect was estimated using a Work Breakdown Structure (WBS)-based method in software engineering [22]. Each response step was broken down into detailed tasks, and the impact of the proposed model's improvement factors on these tasks was analyzed to calculate the time reduction rate. The effectiveness of each major improvement factor was calculated based on prior research. Shortcode-based evidence identification was applied to a study in software maintenance that found that using documented identifiers reduced search time by 80-90% [23]. The selection of the coverage matrix tool was based on the experimental results of Iyengar & Lepper's (2000) study on choice overload, which showed that providing predefined options reduced decision-making time by 70-85% [26]. The predefined queries and scripts were based on research on the effectiveness of automation in the DevOps field [24], and the parallel analysis system was based on Amdahl's law and project management theory [25]. This estimate assumes a prebuilt coverage model, intermediate or higher operator proficiency, and typical system performance. It excludes actual environmental variables, technical obstacles, and organizational procedures. Therefore, the presented improvement rate represents a theoretical upper limit, and may be lower in real-world environments.

B. ANALYSIS BASED ON THE VIASAT KA-SAT INCIDENT (2022)

According to an official Viasat report, [3], the operations team recognized the anomaly within hours, but it took approximately four days to confirm the breach and approximately 35 days to complete a detailed analysis. A step-by-step reconstruction of the actual response process and an analysis of the expected improvements resulting from the application of the proposed model. In the initial detection phase, operators, after recognizing a large number of modem offline alarms, relied on their experience to determine whether it was a link quality problem, equipment malfunction, or a cyberattack. This process of sequentially inspecting multiple systems took approximately two to five hours [3]. When applying the proposed model, the shortcodes "GS-DL-TM" and "GS-INT-DB" are immediately identified upon alarm occurrence, PostgreSQL and pandas are verified in the coverage matrix, and damage aggregation is performed using predefined SQL queries. The detailed tasks of the existing method are broken down into failure type determination (30-60 minutes), system identification (20-40 minutes), query writing and execution (30-60 minutes), and result interpretation (30-60 minutes), taking a total of 110-220 minutes. With the proposed model,

applying an effectiveness coefficient of 0.15-0.40 to each task reduces the total time to 35-70 minutes, representing a potential time reduction of approximately 60% or more, with a maximum potential of 70%. In the root cause identification stage, VPN logs, management server logs, and firmware analysis were sequentially performed, and it took approximately 96 to 125 hours due to delays in information sharing between each team [3]. In the proposed model, after expanding short codes into detailed codes, the network team uses Zeek and Arkime to track abnormal VPN sessions based on the coverage matrix, the system team uses Kibana to visualize abnormal management command patterns, and the control team performs TM/TC correlation analysis in parallel with PostgreSQL. Based on the profile code, the data flow "GS-INT-VPN → GS-INT-MDM → GS-DL-TM" is tracked. Conventional methods require a total of 40-80 hours, including VPN log analysis (8-16 hours), management server log analysis (12-24 hours), and information sharing between teams (8-16 hours). The proposed model, which applies Zeek/Arkime automation (EF=0.25), a Kibana dashboard (EF=0.20), and a parallel analysis system (EF=0.35), reduces the total time to 12-25 hours, representing an expected improvement of approximately 60-70%. During the detailed forensic analysis phase, approximately 720 hours, or 30 days, were required from the time of breach confirmation to the official report release [3]. The proposed model, with its predefined firmware analysis tool pipeline (binwalk → Firmware-Mod-Kit → Ghidra), automated timeline restoration scripts (pandas, matplotlib), and coverage checklists to prevent missing evidence, is estimated to shorten this time from approximately 168 hours to 216 hours, or 7 to 9 days. This represents a time reduction of approximately 70-75%.

C. ANALYSIS BASED ON THE LANDSAT-7/TERRA EVENT (2007-2008)

From 2007 to 2008, the ground stations of Landsat-7 and Terra, US Earth observation satellites, were compromised, and the attackers gained the ability to execute satellite commands [20]. The US Congressional report [20] did not disclose the specific time of detection or response process, but considering that the average detection period for a typical APT attack is approximately 197 days [21], it is estimated that the intrusion was discovered several months to several years after the intrusion. If the proposed model had been implemented, Zeek and Arkime would have been able to detect abnormal access patterns in real time or within hours based on the GS-INT-VPN shortcode, and the Wireshark CCSDS dissector would have been able to verify unauthorized TC packets in real time using the SC-UL-TC shortcode and block them before satellite transmission. Furthermore, it is estimated that the impact scope assessment could have been completed within days through segment-to-segment correlation analysis based on the GS-INT-DB, SC-DL-TM, and LK-UL-RF codes. However, since the specific detection time at the time was not disclosed, quantitative improvement rates cannot be calculated. Therefore, this study uses this case as a supplementary example, not for quantitative performance verification, but to demonstrate the proposed model's potential for preventive detection and blocking. It also suggests the potential for shortening retention times from months to years to detection times within hours or days. The confidence level for this approach is rated medium-low (∗∗...).

D. COMPREHENSIVE ANALYSIS AND IMPLICATIONS

Through an analysis of two real-world cases, we confirmed key improvements in the proposed model. First, systematic evidence identification through shortcodes eliminates the initial confusion of "Where to start?" Second, the immediate selection of appropriate tools through a coverage matrix eliminates manual searches and trial and error. Third, correlation analysis between segments using profile codes minimizes delays in information sharing between teams. Fourth, coverage checklists prevent evidence omissions, reducing the need for reanalysis.

The experimental implementation of this model was conducted to validate reproducibility. This model was simulated using two automated pipelines based on invasion scenarios. Pipeline A (Baseline) simulates the unstructured sequential response as documented in the Viasat incident report. For this pipeline, operators execute the available tools without profiling codes or coverage matrix guidance. Pipeline B (Proposed Model) simulates shortcode-based target identification, coverage matrix-based selective tool execution, and profile code-based data flow tracking as proposed in this paper. This approach ensures validation by quantitative evaluation of our proposed model. This simulation does not execute the actual forensic tools (PostgreSQL, Zeek, etc.) against live data. This is due to the real-world constraints that restrict the access and implementation of satellite data. Non-compliance towards security policies and the application of software that is not certified or pre-approved are unethical practices for utilizing satellite data. Therefore, we utilised the Python scenario generator (seed=42). It produces 20 randomized data instances based on the Viasat KA-SAT incident and other incident scenarios. These instances include varying attacker Internet Protocol(IPs) ranging 30-50, timing, affected modem count (500-8500), target firmware versions (2-4), and target beam IDs (1-3). The invasion scenario complexity influences pipeline behavior due to naturally occurring variance across instances. A reference Docker-composed environment (PostgreSQL, Elasticsearch, Kibana, Zeek, Gitea) is provided for data structure and query pattern verification. The core model program imitates pipeline B by executing the main modules. These modules are, namely, shortcode_identifier.py, coverage_resolver.py, tool_executor.py and evidence_correlator.py. The module shortcode_identifier.py scans data tables obtained from the incident scenarios and produces an ordered shortcode list. Later, this ordered shortcode list is processed and requests coverage_resolver.py to look up from the external coverage_matrix.json. This coverage matrix is loaded from an external JSON file that maps the tools from the collection. Further, the tool_executor.py implements coverage tools to filter,

TABLE V. Pipeline B implementation outcome for one incident

Shortcode	Forensic Tools	Target table	Operation
GS-INT-VPN	Zeek, Wireshark	vpn_auth_log, fw_traffic_log	Filter external IPs, trace lateral movement
GS-INT-MDM	Elasticsearch, Kibana	mgmt_ commands	Filter firmware_update commands, timeline
GS-INT-DB	PostgreSQL	modem_status	GROUP BY firmware/beam/status aggregation
GS-DL-TM	Elasticsearch/ Kibana	modem_status	Time-series OFFLINE event aggregation
GS-INT-SCHED	Gitea/Git	schedule_history	Filter unauthorized/suspicious changes

aggregate, and time-series on actual data records and output artifacts as A1-A5. The evidence_correlator.py then performs key matching to define the root cause of the attack incident. The outcome of this model implementation as pipeline B for one instance is recorded in Table V. In comparison, Pipeline A does not include Modules 1 and 2. All available tools are executed in 6 phases: full traffic scan with manual browsing, full database scans on all tables, broad log grep across all sources, manual schedule review with complexity-dependent miss probability (10-40%), manual cross-referencing with 55-75% per-link success rate, and extra exploratory scans driven by data volume. Root cause identification depends on the completeness of the evidence chain assembled through manual correlation—it is not guaranteed, unlike the previous version, where Pipeline A always succeeded.

The results from both pipeline executions were evaluated based on the proposed quantitative Performance Metrics (PMs). PMs are comprehended as 6 independent PMs, as PM1 to PM6 mentioned in Table VI. Therefore, PMs were recorded by implementing the model through two pipelines for comparing the effectiveness of pipeline B over pipeline A. The experimental results for pipeline A and pipeline B are mentioned in Table. VII. PM1, PM2, PM3, and PM6 convey statistically significant improvements. Particularly, PM6 for Pipeline A identifies the root cause in only 45% of instances (9/20), while proposed Pipeline B achieves 85% (17/20). Pipeline A's PM6 is affected by schedule tampering scenarios where manual correlation cannot establish the schedule-command-impact chain without shortcode guidance. PM4 and PM5 show no significant difference between pipelines. PM4 is constrained by attack type, and PM5 is an automated correlation in Pipeline B that adds a 10% failure rate. Pipeline A follows procedures documented in the actual Viasat incident report and has complexity-driven variability. Also, Pipeline A's root cause identification is data-driven rather than guaranteed, requiring actual evidence chain completeness. Whereas Pipeline B includes a 10% correlation failure rate for temporal ambiguity. Its coverage matrix may also require extension for distinguished ground station configurations. Additionally, injecting noise into the generated data degrades the shortcode identification accuracy.

In the Viasat incident, the overall response time is estimated to have been reduced from approximately 35 days to

TABLE VI. Six quantitative metrics for the simulated pipeline execution

ID	Definition	Unit
PM1	**Evidence Collection Time:** cumulative simulated duration from scenario start to collection of all five key artifacts	seconds (simulated)
PM2	**Total Tool Invocations:** count of all simulated tool executions until root cause identification	count
PM3	**Unnecessary Tool Ratio:** proportion of invocations not contributing to the final conclusion	ratio(0-1)
PM4	**Evidence Completeness:** number of five predefined key artifacts successfully collected	count (/5)
PM5	**Cross-Segment Correlations:** successfully established evidence-to-evidence timeline links	count
PM6	**Root Cause Identified:** binary indicator for correct identification of VPN intrusion as root cause	binary (0/1)

TABLE VII. Model Analysis Results (n=20, seed=42)

Metric	Pipeline A	Pipeline B	Improv.(%)	Cohen's d	p-value
PM1 (sec)	219.7 (SD=89.5)	84.6 (SD=31.5)	0.615	2.01	<.001
PM2 (count)	13.9 (SD=1.0)	9.65 (SD=1.3)	0.306	3.55	<.001
PM3 (ratio)	0.595 (SD=0.10)	0.033 (SD=0.06)	0.944	6.69	<.001
PM4 (/5)	3.95 (SD=1.00)	4.0 (SD=1.03)	-	-0.05	0.33
PM5 (count)	2.5 (SD=1.05)	2.75 (SD=0.97)	-	-0.25	0.449
PM6 Root Cause	0.45 (SD=0.51)	0.85 (SD=0.37)	0.889	-0.9	0.017

approximately 7.5 to 9.5 days, representing an overall improvement of approximately 70%. In particular, some detailed tasks, such as log analysis and eliminating inter-team waiting time during the root cause investigation phase, theoretically demonstrated significant time savings. The Landsat incident demonstrated the potential for a proactive response, enabling early detection and real-time interception of unauthorized commands to halt incidents before they impact satellite missions. However, this validation is a theoretical estimation based on task decomposition. In real-world environments, the improvement rate may be lower due to factors such as large data sets, system performance, and internal organizational processes. Nevertheless, the four key improvement factors presented in this study are logically valid, suggesting that the proposed model can serve as a practical reference model applicable in real-world operational environments.

VII. CONCLUSION

This study explores a satellite system by analyzing its Space–Link–Ground segments, using a combination of profiling code and forensic tools to model artifact flow within a code-based framework. It introduces the concept of "coverage derivation through segment profiling" and shows that the

types of evidence differ across segments. By synchronizing ground-link evidence and applying hierarchical correlation analysis, the research suggests a method to reconstruct a spacecraft's internal state indirectly, offering a viable approach for space forensics, especially where direct access is limited by environmental and technological challenges. The proposed structure can be readily deployed in actual incident response, including acquisition priorities, procedures, and mapping tables. Furthermore, the coverage matrix can be utilized to proactively reveal evidence gaps based on the satellite system profile, improving forensic preparation before orbital deployment. The experimental simulation ensured reproducibility of the proposed model with approximately 70% overall improvement. A limitation of this paper is that the coverage matrix and improvement effects presented in this study are experimentally simulated based on publicly available documents and prior research. This is due to field trials with actual satellite operational data, which is constrained by security policy. Empirical validation in an actual satellite operating environment was not conducted. Future research should quantitatively verify the effectiveness of the proposed model through the establishment of a simulation environment or expert evaluation. However, unauthorized collection or decryption of RF signals can violate relevant laws and policies.

APPENDIX A
SPACE:

In Table. VIII, the space segment, "●" indicates that the tool supports the specific analysis task, while "○" indicates no support.

TABLE VIII. SPACE SEGMENT FORENSIC TOOL COVERAGE

Domain	Sub-system	Analysis Task	Description	SatDump	gr-satellites	GNU Radio	inspectrum	GQRX	SDR#	pyorbital	Orekit	GPredict	Wireshark	Kaitai Struct	binwalk	Firmware-Mod-Kit	unsquashfs	Autopsy	The Sleuth Kit	Ghidra	Radare2	strings	xxd	pandas	matplotlib
Bus-CDHS	OBC	TM Packet Decoding	Decode CCSDS telemetry packets from spacecraft	●	●	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○
Bus-CDHS	OBC	TC Packet Parsing	Parse and validate telecommand packets	○	●	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○
Bus-CDHS	Data Handling	Onboard Log Analysis	Extract and analyze spacecraft logs	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
Bus-AOCS	Gyro	Attitude Data Recovery	Recover attitude sensor telemetry	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
Bus-AOCS	GPS	Orbit Calculation	Calculate satellite orbit from GPS data	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	●	○
Bus-AOCS	Reaction Wheel	Maneuver Detection	Detect unauthorized maneuvers	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
Bus-EPS	Battery	Power TM Analysis	Monitor power subsystem telemetry	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
Bus-EPS	Solar Panel	Energy Generation Tracking	Track solar panel performance	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
Bus-COMS	S-band Transceiver	RF Signal Analysis	Analyze S-band RF signals	●	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Bus-COMS	S-band Antenna	Beacon Signal Decoding	Decode AX.25 beacon packets	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
Payload	Camera	Image Data Extraction	Extract image files from spacecraft	○	○	○	○	○	○	○	○	○	○	○	●	○	○	●	●	○	○	○	○	○	○
Payload	Camera	Image Metadata Analysis	Analyze EXIF and image metadata	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○
Payload	Sensor	Binary Data Parsing	Parse custom sensor data formats	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●	●	○	○
Payload	Firmware	Firmware Extraction	Extract firmware from dump files	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○
Payload	Software	Binary Reverse Engineering	Analyze onboard software binaries	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○

APPENDIX B
LINK:

In Table. IX, the Link segment, "●" indicates that the tool supports the specific analysis task, while "○" indicates no support.

TABLE IX. LINK SEGMENT FORENSIC TOOL COVERAGE

Domain	Sub-system	Analysis Task	Description	GNU Radio	SatDump	gr-satellites	inspectrum	GQRX	Wireshark	tshark	Zeek	Arkime	tcpdump	Scapy	Kaitai Struct	RANCID	Oxidized	pandas	matplotlib
Downlink	S-band RF	Signal Demodulation	Demodulate S-band downlink signals	●	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○
Downlink	Ka-band RF	High-Freq Signal Analysis	Analyze Ka-band RF signals	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
Downlink	TM Stream	Frame Recovery	Recover CCSDS TM frames from RF	○	●	●	○	○	●	○	○	○	○	○	○	○	○	○	○
Downlink	TM Packet	Packet Decoding	Decode telemetry packets	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	○
Downlink	Beacon	AX.25 Decoding	Decode amateur radio beacons	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○
Downlink	IQ Data	Signal Pattern Analysis	Analyze IQ/SigMF signal patterns	●	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○
Uplink	S-band RF	Signal Verification	Verify uplink signal integrity	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
Uplink	TC Stream	Command Packet Analysis	Analyze telecommand packets	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	○
Uplink	RF IQ	Uplink Pattern Detection	Detect unauthorized uplink patterns	●	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○
Network	VPN/FW	Traffic Analysis	Analyze VPN/firewall traffic	○	○	○	○	○	●	●	●	●	●	○	○	○	○	○	○
Network	L3/L4	Session Tracking	Track network sessions	○	○	○	○	○	●	○	●	●	○	○	○	○	○	○	○
Network	UDP/TCP	Protocol Analysis	Analyze transport layer protocols	○	○	○	○	○	●	●	○	○	○	●	○	○	○	○	○
Network	Router Config	Configuration Tracking	Track router/switch config changes	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○
Link Quality	RF Metrics	Link Quality Analysis	Analyze link quality metrics	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●

APPENDIX C
GROUND:

In Table.X and Table.XI, the Ground segment, "●" indicates that the tool supports the specific analysis task, while "○" indicates no support.

TABLE X. GROUND SEGMENT FORENSIC TOOL COVERAGE (A)

Domain	Sub-system	Analysis Task	Description	Wireshark	Zeek	Arkime	tcpdump	Scapy	RANCID	Oxidized	Kibana	Elasticsearch	journalctl	Windows Event Viewer	PostgreSQL	SQLite	pandas	matplotlib	dd	Guymager	Autopsy	The Sleuth Kit	KAPE	Velociraptor	Volatility3	binwalk	Firmware-Mod-Kit	unsquashfs	mtd-utils	Ghidra	IDA Pro	Radare2	strings	xxd	git	Trivy
RFAntenna	Antenna System	RF Quality Monitoring	Monitor antenna RF quality metrics	○	○	○	○	○	○	○	●	●	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
RF/Antenna	Antenna Log	Log Analysis	Analyze antenna system logs	○	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Network	Firewall	FW Log Analysis	Analyze firewall logs for intrusions	○	●	●	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Network	VPN	VPN Session Analysis	Analyze VPN session data	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Network	Router /Switch	Config Change Tracking	Track network device config changes	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
Network	Packet Capture	Network Forensics	Capture and analyze network packets	●	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Server	Linux	System Log Analysis	Analyze Linux system logs	○	○	○	○	○	○	○	●	●	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Server	Windows	Event Log Analysis	Analyze Windows event logs	○	○	○	○	○	○	○	●	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Server	Application	App Log Correlation	Correlate application logs	○	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Operations	MPS	Schedule Tracking	Track mission planning schedule changes	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
Operations	Database	TM/TC Archive Query	Query historical TM/TC data	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Operations	Database	Transaction Log Analysis	Analyze database transaction logs	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Operations	API	API Call Analysis	Analyze API call patterns	○	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Storage	NAS/ Disk	Disk Acquisition	Acquire forensic disk images	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Storage	Disk Image	File System Analysis	Analyze file system structure	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Storage	Timeline	Timeline Reconstruction	Reconstruct activity timeline	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Storage	Firmware	Firmware Extraction	Extract firmware from devices	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	○	○	○	○	○	○	○

TABLE XI. GROUND SEGMENT FORENSIC TOOL COVERAGE (B)

Domain	Sub-system	Analysis Task	Description	Wireshark	Zeek	Arkime	tcpdump	Scapy	RANCID	Oxidized	Kibana	Elasticsearch	journalctl	Windows Event Viewer	PostgreSQL	SQLite	pandas	matplotlib	dd	Guymager	Autopsy	The Sleuth Kit	KAPE	Velociraptor	Volatility3	binwalk	Firmware-Mod-Kit	unsquashfs	mtd-utils	Ghidra	IDA Pro	Radare2	strings	xxd	git	Trivy
Storage	NAS	Network Storage Analysis	Analyze network-attached storage	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Endpoint	Workstation	Artifact Collection	Collect forensic artifacts	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○
Endpoint	Memory	Memory Dump Analysis	Analyze memory dumps	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○
Endpoint	Binary	Reverse Engineering	Reverse engineer binaries	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○
Endpoint	Binary	String Extraction	Extract strings from binaries	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○
Endpoint	Container	Vulnerability Scanning	Scan containers for vulnerabilities	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
Cloud	Audit Log	Cloud Audit Analysis	Analyze cloud service audit logs	○	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) (Project No. 2022-11220701, 30%, RS-2023-00228996, 20%; RS-2024-00438551, 10%, IITP-2025-RS-2021-II211816, 10%), the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2025 (Project Name: Training Global Talent for Copyright Protection and Management of On-Device AI Models, Project Number: RS-2025-02221620, 20%), and the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Project No. RS-2023-00208460, 10%).

REFERENCES

[1] Jones HW. The recent large reduction in space launch cost. In: 48th International Conference on Environmental Systems (ICES); 2018 Jul 8-12; Albuquerque, NM, USA. Report No. ICES-2018-81.
[2] KBS News. "US Satellite Hacking Damage… Suspected to be from China" [Internet]. 2011 Oct 27 [cited 2025 Dec 19]. Available from: https://news.kbs.co.kr/news/pc/view/view.do?ncd=2378797
[3] Viasat. Ka-Sat network cyber attack overview [Internet]. Carlsbad (CA): Viasat; 2022. Available from: https://www.viasat.com/perspectives/corporate/2022/ka-sat-network-cyber-attack-overview/. Accessed 2025 Dec 20.
[4] Temple J. Iridium satellite constellation. IEEE Spectrum [Internet]. 2017. Available from: https://spectrum.ieee.org/iridium-satellite. Accessed 2025 Dec 20.
[5] Wouters L. Glitched on Earth by humans: A black-box security evaluation of the SpaceX Starlink user terminal. In: Proceedings of Black Hat USA 2022; 2022 Aug; Las Vegas, NV, USA.
[6] Salim S, Moustafa N, Reisslein M. Cybersecurity of satellite communications systems: A comprehensive survey of the space, ground, and links segments. IEEE Commun Surv Tutor. 2025;27(1):372–425.
[7] Alotaibi FM, Al-Dhaqm A, Al-Otaibi YD, Alsewari AA. A comprehensive collection and analysis model for the drone forensics field. Sensors (Basel). 2022;22(17):6486.
[8] Peled R, Aizikovich E, Habler E, Elovici Y, Shabtai A. SoK: Evaluating the security of satellite systems. arXiv. 2023; arXiv:2312.01330.
[9] Al-Dhaqm A, Ikuesan RA, Kebande VR, Razak S, Ghabban FM. Research challenges and opportunities in drone forensics models. Electronics (Basel). 2021;10(13):1519.
[10] Bouafif H, Kamoun F, Iqbal F, Marrington A. Drone forensics: Challenges and new insights. In: Proc Cybercrime Investigation and Digital Forensics Workshop (CID 2018). IEEE; 2018.
[11] Iqbal F, Yankson B, AlYammahi MA, AlMansoori N, Qayed SM, Shah B, et al. Drone forensics: Examination and analysis. Int J Electron Secur Digit Forensics. 2019;11(3):245–264.
[12] Wertz JR, Everett DF, Puschell JJ, editors. Space mission engineering: the new SMAD. Torrance (CA): Microcosm Press; 2011.
[13] National Aeronautics and Space Administration. State-of-the-art of small spacecraft technology. Washington (DC): NASA; 2024.
[14] Kim HD, Choi WS, Kim MK, Kim JH, Kim KD, Kim JS, et al. Design and development of the SNIPE bus system. J Space Technol Appl. 2022;2(2):81–103.
[15] Consultative Committee for Space Data Systems. CCSDS recommended standards [Internet]. Reston (VA): CCSDS. Available from: https://ccsds.org/. Accessed 2025 Dec 20.
[16] Lee HH, Kim BY, Park BK, Yang KH, Baek MJ, Chun YS. Utilization and effect of satellite simulator for COMS operation preparation. Aerospace Eng Technol. 2010;9(1):84–92.
[17] Mura A. An analysis of the cyberattack against ViaSat of February 2022: from technical details to the overall relevance for cybersecurity of critical infrastructures. Bologna: University of Bologna; 2024. Unpublished manuscript.
[18] UK Government. Russia behind cyber attack with Europe-wide impact an hour before Ukraine invasion [Internet]. London: GOV.UK; 2022. Available from: https://www.gov.uk/government/news/russia-behind-cyber-attack-with-europe- wide-impact-an-hour -before-ukraine-invasion. Accessed 2025 Dec 20.
[19] SentinelOne. AcidRain: a modem wiper rains down on Europe. SentinelOne Labs; 2022.
[20] U.S.-China Economic and Security Review Commission. China's cyber activities. Washington (DC): USCC; 2011.
[21] Mandiant. M-Trends 2013: attack the security gap. Alexandria (VA): Mandiant; 2013.
[22] Project Management Institute. A guide to the project management body of knowledge (PMBOK guide). 6th ed. Newtown Square (PA): PMI; 2017.
[23] IEEE. IEEE Std 1219-1998: IEEE standard for software maintenance. New York: IEEE Computer Society; 1998.
[24] Humble J, Farley D. Continuous delivery: reliable software releases through build, test, and deployment automation. Boston (MA): Addison-Wesley; 2010.
[25] Kerzner H. Project management: a systems approach to planning, scheduling, and controlling. 12th ed. Hoboken (NJ): Wiley; 2017.
[26] Iyengar SS, Lepper MR. When choice is demotivating: can one desire too much of a good thing? J Pers Soc Psychol. 2000;79(6):995–1006.

BAE GEUN CHO received the B.S. degree in computer engineering and information and communication engineering from Hanyang Cyber University, Seoul, South Korea, in 2014. He is currently pursuing the M.S. degree in information security at Sejong University, Seoul, South Korea, with an expected graduation in 2026.

From 2019 to 2024, he worked as a Satellite Operations Engineer at iOPS. Since 2024, he has been with CES, a satellite imagery provider. His research interests include satellite systems, remote sensing, and information security.

He significantly contributed to the Korean Society for Aeronautical and Space Sciences with the concept and design of a general-purpose simulator, cloud-based ground station service, and multi-satellite operation platform.

ARPITA DINESH SARANG received the B.S. and M.S. degrees in Computer Science from the University of Mumbai, India, in 2022. She is pursuing a full-time Ph.D. degree in Computer and Information Security from Sejong University, Seoul, South Korea, from 2023.

She is highly passionate about cyber threat analysis for innovative platforms. Her main focus is building Cyber Threat Intelligence (CTI) for software infrastructures. Her contributions range from malware reverse engineering to system security. Her previous experience involves working on the development of the Access Control Management (ACM) software products and the Delivery Assurance Tools. She worked on a system penetration project by exploiting Windows system vulnerabilities. She is a member of the Indo-Pacific-European Hub for Digital Partnerships (INPACE).

JUN-HO HONG is an assistant professor in the Department of Convergence Security Engineering, Sungshin Women's University. He received the B.S., M.S., and Ph.D. degrees in Law from Dankook University. From 2014 to 2024, he was director of the Korea Information Security Education Institute and, Korea Information Security Industry Association.

He is interested in working towards personal information protection. He was awarded for Meritorious Service to the development of the information security industry and the Korea Law Association Achievement. He co-authored and published books on the Personal Information Protection Field, Practical Techniques, and Introduction to Personal Information Protection and Utilization.

KI-WOONG PARK is a full professor in the Department of Computer and Information Security at Sejong University. He received the B.S. degree in computer science from Yonsei University, and the M.S. and Ph.D. degrees in computer science and electrical engineering from KAIST. In 2008-2009, he was a graduate researcher at Microsoft Research Asia and Microsoft Research Redmond, and a researcher at the National Security Research Institute (NSRI) in 2012.

He has a passionate interest in designing, building, and analyzing secure systems, especially for cloud computing systems, networked systems, and embedded systems. He often takes the approach of reevaluating existing security mechanisms as well as the actual system implementation and subsequent evaluation in real computing systems. His work was supported by a Microsoft Research Fellowship from 2009 to 2010. He is a member of the IEEE, the IEEE Computer Society, and ACM.