SIEM4GS: Security Information and Event Management for a Virtual Ground Station Testbed

Yee Wei Law[1] and Jill Slay[1,2]
[1]UniSA STEM, University of South Australia, Mawson Lakes, Australia
[2]SmartSat Cooperative Research Centre, Adelaide, Australia
YeeWei.Law@unisa.edu.au
Jill.Slay@unisa.edu.au

Abstract: As the space sector continues to grow, so do the cybersecurity risks. As large as the attack surface of a space system is, the ground segment remains an attractive source of intrusion points, not only because of its relative accessibility but also because the ground system is often viewed as little more than a conventional IT system. Thus, a representative security assessment of a space system cannot avoid addressing the vulnerabilities of the associated ground system and the relevant threats. This motivates the construction of a virtual ground station testbed, as part of larger reference platform, to support our ongoing research on the cybersecurity of space systems. Presented here is a discussion of the preliminary work being undertaken at the University of South Australia node of the SmartSat Cooperative Research Centre on such a testbed. A distinguishing feature of the testbed is the integration of a security information and event management (SIEM) system justifying the name of the testbed, "SIEM4GS". Based on the latest literature on ground stations, a logical architecture and an implementation plan involving only open-source software building blocks for SIEM4GS are proposed. Features of the ground station and SIEM services are discussed. A plan is provided on how to extend the SIEM system from a primarily "detect" role in the NIST Cybersecurity Framework to a "detect and respond" role.

Keywords: Ground station, ground system, mission operations service framework, security information and event management, extended detection and response, Elastic Stack.

1. Introduction

The space sector has been expanding rapidly in recent years. For example, between 2019 and 2020, the number of spacecrafts launched per year more than doubled, and 100,000 satellites are expected to be in orbit by the end of the decade (McDonald et al., 2021). Accompanying this growth is the mounting recognition of the pivotal role of space capabilities in national security. As such, securing space assets is a national priority. However, the inherently large attack surface of space systems poses immense challenges. Attacks — whether cyber, physical or both — can come from a wide variety of threat actors, through the supply chain or from the user end, to wreak havoc on the space segment, ground segment or user segment, before the aftermath spills over to other segments (Pavur and Martinovic, 2020). The research reported here is concerned with the *ground system*.

Definition 1. A *ground system* is an integrated set of ground functions used to support the preparation activities leading up to mission operations or the conduct of mission operations itself (ESA, 2008).

A ground system consists of ground stations, ground networks, control centres and remote terminal as its primary elements (Weston, 2020, Table 12-1). In the literature, the term "ground station" is used to refer to both the physical infrastructure and the ground system; it is for this reason "ground station" and "ground system" are used interchangeably in the ensuing discussion. In the terminology of NASA's Advanced Multi-Mission Operations System (AMMOS, see Benecken, 2020), a ground system is composed of a Mission Operations System (MO System) and associated earth-bound communications and data acquisition infrastructure (see Figure 1). A standard and precise definition of a MO System does not exist, but it can be understood as a set of implementation components that include a flight team and a *Ground Data System* (GDS) for ensuring the correct operations of a space mission. A GDS is a set of software, hardware and facilities as

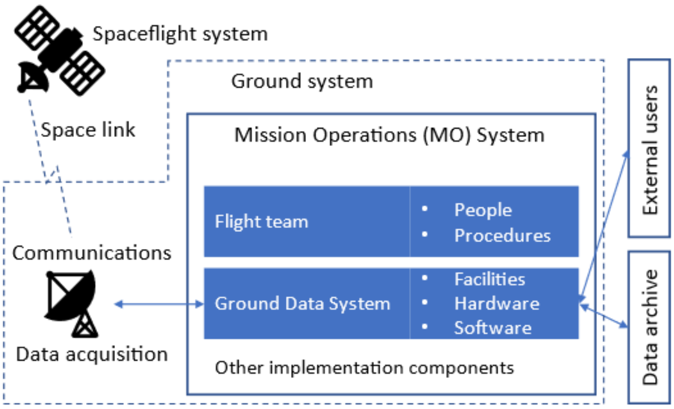

Figure 1: The components of a ground system in NASA's AMMOS terminology (Benecken, 2020).

well as support services (e.g., system administration support) for collecting and distributing mission data (Benecken, 2020). A rigorous discussion of the definitions of a MO System is deferred to Sec. 2.2.

Our focus on the security of ground systems through this work is based on the following rationales (Pavur and Martinovic, 2020, Sec. VI): (i) Coming from the ground-based Internet, ground systems represent an obvious point of entry into space systems. (ii) Ground stations are usually located in remote areas with limited physical security and are barely staffed, necessitating remote access to ground stations by operations centre staff located elsewhere, so ground stations can hardly be "air-gapped". (iii) A ground system typically serves multiple missions and multiple space agencies, so a pathway into a ground system can lead to compromise of multiple missions and agencies. In this scenario, a ground system represents a single point of failure. (iv) Ground system security has traditionally been treated as an extension of mainstream IT security, and as such, there has *not* been active research targeting the specific hardware, software, networks and architectures of ground systems.

The work presented here is concerned with a virtual ground station testbed, expanding our existing low-Earth-orbit satellite digital twin (Ormrod et al., 2021), to support our ongoing research into the cyberworthiness and cyberresilience of space systems. As our first step towards securing this testbed, a *security information and event management* (SIEM) system is used to monitor the network traffic of the testbed. SIEM is (i) the analysis of event data in real time for early detection of targeted attacks and data breaches, and (ii) the collection, storage, investigation and reporting on log data for incident response, forensics and regulatory compliance (Gartner, 2021). Our direction reflects industry best practice (Vera, 2016; Miller, 2021), and real-world needs to meet compliance requirements (Exabeam, 2021) such as the Sarbanes-Oxley Act.

Our contributions are as follows: (i) Our multivocal but rigorous literature review in Sec. 2 on ground systems, not only provides the theoretical basis for our virtual ground station testbed, but also clarifies space-related concepts and definitions that are not necessarily familiar to the security community, the primary audience of this paper. (ii) The proposed SIEM4GS architecture can serve as a reference architecture for future ground station security research. Once the source code for SIEM4GS is released, SIEM4GS can serve as a reference implementation too. The absence of a testbed similar to SIEM4GS testifies to the novelty of our work.

2. Literature review on ground systems

The primary ground system terminology is sourced from NASA, European Space Agency (ESA), and Consultative Committee for Space Data Systems (CCSDS). There is a large body of literature on ground system architectures and designs that can be sourced not only from the space agencies, but also from academia and industry. Traditional ground systems used a "chimney" or "stovepipe" architecture, catering to a single user and a single mission over a period of time. Current trends include:

- **Software-defined front-end processing**: A *front-end processor* (FEP) is a programmed-logic or stored-program device that interfaces data communication equipment with an input/output bus or memory of a data processing computer (Telecommunications Industry Association, 2021), and whose main functions are signal processing and encryption/decryption (Lowdermilk and Sethumadhavan, 2021). Fischer and Scholtz (2010) treat a "front end" as an assembly of antennas and rotators; and delegate signal processing functionality to a terminal node controller (TNC). However, since TNCs are only capable of VHF and UHF communications (Ahmad et al., 2016), *software-defined radios* (SDRs, i.e., radios in which the physical-layer functions are software-defined) are superseding TNCs to support communications in multiple frequency bands, especially the S band and X band (Weston, 2020). In other words, front-end processing is increasingly software-defined.
- **Adoption of SDR standard VITA 49:** The advent of SDRs created the need to ensure interoperability between diverse SDR components (Normoyle and Mesibov, 2008), especially within the signals intelligence community. To address this need, members of the defence industry and the VMEbus International Trade Association (VITA) initiated the analogue RF-digital standard called VITA 49 or VITA Radio Transport (Cooklev et al., 2012). This standard specifies a packet-based transport protocol for representing (i) digitised signal data, and (ii) metadata or context data about the radio (e.g., frequency, gain), the location of the radio and the processing done on the signal prior to the generation of the signal data packet. Besides interoperability, this standard enables accurate alignment of signal data and discrete events between multiple receivers that are either in the same location or separated by large distances. The many advantages of VITA 49 led to its quick adoption by the space sector.

- **Supporting multiple missions**: The proliferation of components ranging from field-programmable gate arrays (FPGAs) to SDRs has put deployment of low-cost satellites within the reach of organisations with modest budgets. However, barring educational projects like https://nyan-sat.com, the cost of licensing and setting up a dedicated ground station remains prohibitive for these organisations. By catering to multiple missions through technologies such as virtualisation and cloud computing, a ground station can offer its services to more organisations at a lower cost. After all, two consecutive passes of a satellite are typically separated by abundant idle time (Fischer and Scholtz, 2010).
- **Virtualisation**: This, in the context of ground stations, means reusing the same physical infrastructure for multiple simultaneous missions. To facilitate virtualisation, CCSDS (2010b) has defined a *Mission Operations Service Framework* (more details in Sec. 2.2) consisting of two interface layers that provide the patterns or templates of interaction between a consumer and a provider of a service. Through the standardised interfaces, any framework-aware mission operator can access any service that supports the framework. This allows any framework-compliant ground system to serve multiple missions. Besides standardised service interfaces, *scheduling* and *arbitration* are key to virtualisation, because a limited amount of hardware resources (e.g., antenna pool) means only a bounded number of operators can be given access to the resources at any given time. CCSDS (2018) has defined the XML-based Simple Schedule Format (SSF) for specifying scheduling information related to apertures at ground stations and/or relay satellites between operating entities. This allows mission operators to plan ahead to ensure there are sufficient aperture resources for their planned missions, and the ground station operator to ensure its resources are not overbooked. One way to achieve arbitration of resources is *software-defined networking* (Liu et al., 2020; Riffel and Gould, 2016). This mechanism enables a ground station operator to dynamically switch mission operators to their allocated antenna system at the scheduled time.
- **Cloud-based access and cloud-native implementation:** To be able to provide access and services that scale elastically, ground station operators increasingly resort to cloud computing technologies. As services and data are mirrored at redundant sites, cloud-based ground systems can achieve high reliability. An example of a “Ground Station as a Service” (GSaaS) offering is AWS Ground Station (see Sec. 2.3).
- **Network of ground stations:** Communications between a satellite and a ground station are limited to the portion of the satellite’s orbit during which the satellite is within line of sight from the ground station, i.e., less than ten minutes in a single pass. This means a latency of several hours to several days before the user receives satellite’s data. Furthermore, even when a satellite is in radio contact with a ground station, weather effects can cause more than 80% of packet loss (Vasisht and Chandra, 2020). Besides space-based relays, a network of geographically distributed ground stations can be used to provide all-time coverage and continuous access to communication and tracking services. *Virtualisation*, as discussed earlier, is one way to facilitate inclusion of a ground station into a ground station network. Examples of global ground station networks include Leaf Space, Satellite Network Open Ground Station (SatNOGS, see Sec. 2.1).

Below, discussion of ground system architectures is divided into (i) academia; (ii) space agencies; (iii) GSaaS.

2.1 Academia

The best-known architectures in academia include Standard University’s Mercury (Cutler and Fox, 2006), and SatNOGS. The former is defunct, so we focus on SaNOGS.

SatNOGS is an open-source software and open hardware project initiated by Hackerspace.gr and embraced by academia. SatNOGS aims to provide participants worldwide with crowd-sourced resources for building a global network of satellite ground stations (Surligas et al., 2021). SatNOGS consists of four major subprojects:

- **SatNOGS Ground Station** is a collection of hardware designs and specifications for antennas, rotators with 3D-printed parts for directional antennas, and front-end processors.
- **SatNOGS Client** is a software running on a computer controlling Ground Station hardware. The Client regularly polls the Network (discussed later) for observation jobs scheduled for the local Ground Station. Client functions include scheduling, rotator control, GNU Radio-enabled Doppler tuning and signal demodulation. Demodulated signal data, logs and other reports are queued for upload to the Network.
- **SatNOGS Network** is a web application hosted at https://network.satnogs.org, to which users can submit scheduled observation requests. Based on the requests, the Network calculates the observation windows from available Ground Stations. Once an observer accepts an observation job proposed by the Network, the job is inserted into the job queue of the observer’s Ground Station. Likewise, a Ground Station can query the Network for its list of scheduled jobs and uploads its data to the Network. When calculating

possible observations, SatNOGS Network extracts the relevant information from the SatNOGS Database (discussed next).

- **SatNOGS Database** is a web application hosted at https://db.satnogs.org, providing access to crowd-sourced satellite information and collected telemetry data. Satellites are identifiable by their North American Aerospace Defense Command (NORAD) space object catalogue number and their common name. Transponder records for each satellite are extracted from https://CelesTrak.com.

2.2 Space agencies

In NASA's literature (Weston, 2020), a group station comprises (i) a ground station terminal, (ii) a MO Centre, (iii) a Science Operations Centre, and (iv) data storage and network. While the MO Centre is responsible for ensuring the success of a mission, the Science Operations Centre is responsible for instructing the MO Centre what science operations to perform, besides generating and disseminating science data products. In NASA's AMMOS architecture depicted in Figure 1, the MO System serves as the system supporting the MO Centre, while the "External users" block plays the role of the Science Operations Centre.

ESA's (2008) ground system architecture in the standard ECSS-E-ST-70C differs from NASA's, but it also features a MO System. A ground segment in ESA's ECSS-E-ST-70C architecture has four top-level systems, namely (i) a ground station system, (ii) a ground communications system, (iii) a MO System, and (iv) a payload operations and data system. The ground communications system provides the interconnections between systems, to enable data distribution, voice and video communications, and system maintenance. In terms of their functions, ESA's version of MO System does not completely overlap with NASA's version of MO System.

Instead of the MO System, MO Services receive the focus in CCSDS' standards. CCSDS' *MO Service Framework* (CCSDS, 2010b) is based on the principles of a *service-oriented architecture*, and the framework defines (i) a model for interaction between two entities, and (ii) a model for common services providing functionality common to most uses of the service framework. From the perspective of the open systems interconnection model, the framework defines two layers between the application layer and the transport layer:

- The **MO Services Layer** sits right below the application layer. Two types of services are provided through this interface: *Functional Services* and *Common Services*. Functional Services are MO-specific services, such as monitoring and control (M&C), scheduling, etc., while Common Services are general services that even Functional Services might need, such as service directory, user login/authentication, etc. Both Functional Services and Common Services are defined as specialisations or extensions to the Common Object Model (COM). The COM provides a common information model for all MO Service objects. In this model, any change in the attributes of an MO Service object triggers an automation event. For each MO Service, the operations are defined as specialisations or extensions to the generic interaction patterns defined in the Message Abstraction Layer, to be discussed next.
- The **Messaging Abstraction Layer** (MAL, see CCSDS, 2013) sits below the MO Services Layer and above the transport layer. For ground systems to be interoperable, i.e., a user to be able to access any service that is compliant with the MO Service Framework, the consumer-provider communication protocol must be standardised. The MAL meets this standardisation need by specifying data types and structures (which the COM inherits), message format, and message exchange patterns (e.g., SEND, SUBMIT).

2.3 Ground Station as a Service (GSaaS)

The separation of the MO System from the ground station system by the ground communications system in ESA's architecture is mirrored by cloud-based ground system architectures. AWS Ground Station (Amazon Web Services, 2021) and Azure Orbital (Microsoft, 2021) are two examples of these architectures, and a significant commonality between them is observable. The representative cloud-based architecture supports (i) radio communications in the UHF, S and X bands; (ii) digitising RF signals and serving the digitised streams through the standardised packet-based protocol VITA 49; (iii) process automation and security. We note this architecture classifies *telemetry, tracking and command/control* (TT&C, see Definition 2) as a function of the customer's MO System, in a departure from ESA's ECSS-E-ST-70C architecture.

Definition 2. In TT&C (Guest, 2017), (i) *telemetry* is the collection of on-board measurements and instrument readings required to deduce the health and status of all subsystems of a satellite (e.g., propellant supply) and the transmission of this data to the command segment on the ground; (ii) *tracking* or more precisely, *carrier tracking*, refers to the process of locating and locking onto a satellite from a ground station; (iii) *command/control* refers to the execution of action sequence (e.g., toggling a relay) on the satellite and its payloads to meet mission objectives.

Considering the increasing popularity of GSaaS, our testbed SIEM4GS is based on this architecture, while the core services of SIEM4GS are informed by CCSDS' MO Service Framework. Sec. 3 has more details.

3. Architecting and implementing SIEM4GS

The preceding review provides the basis for the ground station part of SIEM4GS. The following subsection covers some preliminaries on the SIEM part of SIEM4GS, while Sec. 3.2 describes the architectures.

3.1 Preliminaries on SIEM

SIEM, as defined in Sec. 1, is an integral part of a modern IT infrastructure. A core component of a SIEM system is its *rule engine* (Exabeam, 2021), which contains rules that are run periodically. A SIEM rule is a set of conditions (including thresholds) expressed in Boolean logic, schedules and actions that enable notifications (Elasticsearch, 2021). SIEM rules are often called *correlation rules* because they are processed by means of *correlation*. An *alarm*/*alert* is triggered when the result of correlation is significant. The mechanism of *alarm correlation* or *alert correlation* is to ensure the frequency of alarms is manageable, and it refers to the conceptual interpretation of multiple alarms such that a new meaning is assigned to these alarms (Jakobson and Weissman, 1993). When an interpretation is consistent with an attack scenario, then the original multiple alarms can be merged into one; otherwise, no alarm should be triggered so that the number of false alarms is minimised. It is unclear how many of the latest advances in alert correlation (Salah et al., 2013), including those for detecting multi-stage attacks (Shin et al., 2019) have been implemented in commercial SIEM systems. By popular assessment (Barros, 2017; Manor, 2021), real-world SIEM rules remain simplistic to the degree that threat coverage and false positive rate remain undesirable. Thus, additional technologies are needed to complement the functionality of a traditional SIEM system, including:

- **User and Entity Behaviour Analytics (UEBA):** Also known as User Behaviour Analytics, this refers to application of data analytics to the discovery of abnormal/risky behaviour by users or entities such as endpoints, servers and routers, often in conjunction with a SIEM (Fortinet, 2021). UEBA can help detect anomalies that SIEM rules cannot, using for example specialised neural networks (Sharma et al., 2020), but it requires determination of user and entity baseline behaviour and painstaking feature engineering.
- **Security Orchestration, Automation and Response (SOAR):** This is the planning, integration, coordination and cooperation of the activities of security tools and experts to produce and automate required remediations in response to security incidents across multiple technology paradigms (Islam et al., 2019). Whereas SIEM automates the "detect" function defined in NIST's (2018) Cybersecurity Framework, SOAR aims to automate both the "detect" and "respond" functions of the framework, alleviating the "alert fatigue" problem of traditional SIEM.
- **Extended Detection and Response (XDR):** This refers to the extension of traditional "endpoint detection and response" to threat detection, investigation and response solutions that work across all threat vectors in an organisation's infrastructure, including endpoints, servers, networks and cloud, rather than just one piece thereof (Palo Alto Networks, 2022). Whereas SOAR serves as an upgrade to SIEM, XDR is often seen as comparable to SOAR but with a keener focus on threat detection and incident response (Miller, 2021). XDR started out as an industrial initiative for large vendors to integrate their security products and present a single pane-of-glass view of security information ("native XDR") but has evolved to embrace openness ("open XDR" or "hybrid XDR"). XDR aims to maximise situational awareness and coverage of the attack surface through enhanced data/intelligence ingestion and contextualisation. The success of both SOAR and XDR rests heavily on the synergistic leveraging of the latest software and hardware advances in big data analytics and artificial intelligence.

To implement the SIEM part of SIEM4GS, we use Elastic Stack because (i) its open source provides flexibility for customisation and experimentation; (ii) it has strong community support; (iii) it is widely used for cybersecurity research. Elastic Stack comprises four main components using the same Elastic Common Schema data format, namely (i) the distributed, RESTful search and analytics engine, Elasticsearch; (ii) the visualisation app, Kibana; (iii) the network and host data integrator consisting of endpoint kernel-level antimalware called Elastic Endpoint, distributed data shippers called Beats and server-side data processing pipeline called Logstash; and finally (iv) security content created by Elasticsearch and its user community. Elastic Stack supports open/hybrid XDR through its "Limitless XDR" feature, serving our research needs.

Our long-term plan is to push the envelope of XDR by exploring deep *semi-supervised*, *weakly-supervised* and *self-supervised* learning approaches for threat detection (Pang et al., 2022) because (i) manual data labelling for supervised learning is not scalable; (ii) unsupervised learning suffers from high false positive rates. Furthermore, learning will be done on *attributed networks*. An attributed network is a graph whose vertices have attributes (Liu et al., 2021). Their ability to model flows and multi-stage attacks, combined with a data-efficient learning paradigm with a low false positive rate, should pave way for fulfilling the promise of XDR and extending the "detect" function of SIEM's system to "detect and respond".

3.2 Architectures and implementation plan

Figure 2 and Figure 3 show the logical and physical architectures for SIEM4GS. For each of the non-SIEM logical blocks in Figure 2, a main reference implementation is identified with commentary, while alternative reference implementations are listed in Table 1 without commentary.

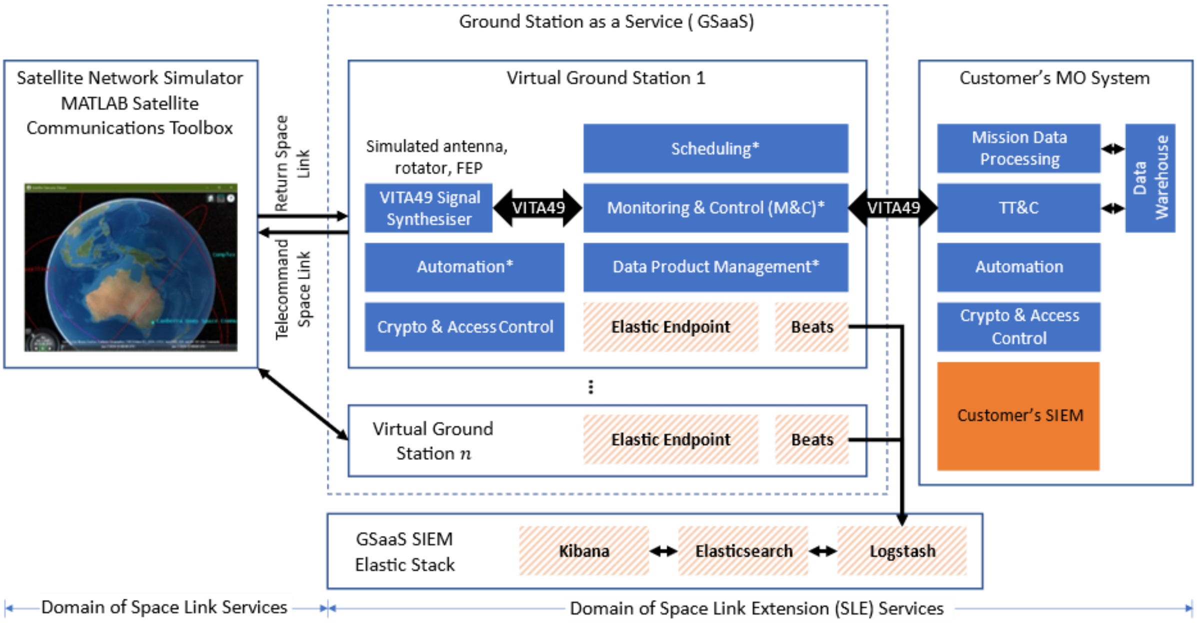

Figure 2: Logical architecture of the proposed virtual ground station testbed, SIEM4GS. Block names with an asterisk are associated with Functional Services identified by CCSDS (2010b). The hatch-filled orange blocks are part of the Elastic Stack SIEM.

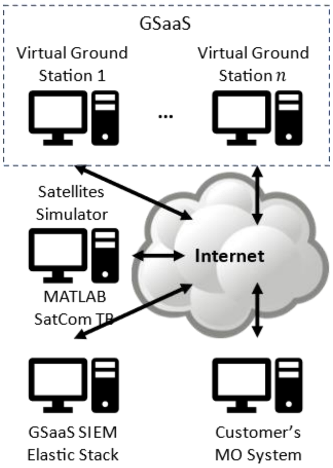

Figure 3: Physical architecture of SIEM4GS.

The logical blocks are discussed below:

- **Satellite Network Simulator**: This simulates satellite orbits and space-ground links; and interact with one or more Virtual Ground Stations that are part of the GSaaS serving the Customer's MO System. The simulator can be implemented using MATLAB's Satellite Communications Toolbox and Aerospace Blockset.
- **VITA 49 Signal Synthesiser:** As SIEM4GS is by design virtual, radio signals are synthesised and furthermore in the VITA 49 format to comply with the current standard. The VITA 49 signal stream serves as input to the Monitoring & Control block. This synthesiser is implemented using the open-source C++-based REDHAWK SDR framework (Robert et al., 2015). Building on the Common Object Request Broker Architecture (CORBA), REDHAWK can interoperate with software components written in any other language provided an object request broker exists for that language (e.g., Java).
- **Monitoring & Control (M&C):** Real-time M&C of a spacecraft (both pre-launch and post-launch) includes downlink telemetry processing and visualisation, as well as formulation and initiation of the transmission of spacecraft (tele)commands (Benecken, 2020). Telemetry data is formatted in the CCSDS-prescribed XML Telemetric and Command Exchange (XTCE) schema (OMG, 2019). By design, operationally significant events or anomalies trigger

alerts to be sent to subscribers of this service. XTCE-conformant M&C is built on Space Applications Services' Yamcs, an open-source Java-based software framework for M&C of spacecrafts, payloads and ground equipment (Schmitt et al., 2018).

- **Scheduling:** Upon generating a conflict-free schedule to automate MO (e.g., ground-space communications, data acquisition), a mission planning application sends the schedule to a scheduler via this Scheduling service. A schedule is a container for (i) predicted events, (ii) planned contacts (periods of ground-space connectivity), and (iii) scheduled tasks. The scheduler is responsible for (i) distributing the schedules to the applications responsible for executing the scheduled tasks, (ii) monitoring the status of the scheduled tasks, (iii) controlling the schedules, and (iv) feeding back the status of the scheduled tasks to the schedule-generating application. Scheduling is crucial because flight system tracking hours are limited by total user demand, as well as internal engineering and maintenance. There is currently no open-source implementation of the SSF (see Sec. 2), so SIEM4GS does not support SSF initially. The scheduler is built on that of the SatNOGS Client.
- **Automation:** Responses to events of interest are routinely automated. For example, AWS Ground Station enables automation of AWS services in response to satellite contact status via the mechanism of CloudWatch Events (Amazon Web Services, 2021). Implementation of this service can readily leverage the event handling mechanism or publisher-subscriber design pattern of a modern programming language, e.g., Java provides the Observable and Observer interfaces to support the observer design pattern.
- **Data Product Management:** This is the generation of scientific instrument data product, including processing, display and delivery, for use by instrument engineers, science planners and operators, as well as for public information releases. This service supports the transfer of data products in both directions, and alerts service subscribers about changes to the data product store, such as new product events. An archive pipeline typically includes metadata/label design, data format conversion, validation and delivery to archive storage, e.g., Open Archival Information System (CCSDS, 2012), Planetary Data System (Benecken, 2020). This service can be implemented on top of the CCSDS File Discovery Protocol (CFDP, see CCSDS, 2020), an implementation of which exists in Yamcs.
- **Cryptography & Access Control:** One of the six Common Services in CCSDS' MO reference model is Login (CCSDS, 2010a). Here, the Login service is generalised to cryptographic and access control services. As per CCSDS' (2019a) mandate, SHA-256 is used for collision-resistant hashing; AES-GCM with 256-bit keys is used for authenticated encryption with associated data; DSA/RSA/ECDSA is used for digital signatures. Additionally, as per CCSDS' (2019b) report, Space Data Link Security (SDLS) is used for link security; TLS is used for end-to-end security. SDLS is implemented using NASA's open-source C-based CryptoLib, while the rest is implemented using OpenSSL. A Java API for OpenSSL is available from Wildfly. A simple implementation of access control is available from Yamcs.
- **TT&C**: This in the representative GSaaS architecture (see Sec. 2.3) is the responsibility of the customer's MO system. The tracking part can be technically demanding because it involves the use of tones/pseudocode or sensing of Doppler frequency shifts to achieve ranging (Kinman, 2021). For the purpose of our virtual testbed, tracking is assumed to be unnecessary and hence not implemented.

Table 1: Alternative reference implementations for select logical blocks in Figure 2.

Logical block	Project name, main programming language, URLs
Satellite Network Simulator	SatSim, Python, https://gitlab.com/librecube/prototypes/python-satsim
	SNS3, C++, https://github.com/sns3/sns3-satellite
	OS3, C++, https://github.com/inet-framework/os3
M&C	ReatMetric, Java, https://github.com/dariol83/reatmetric, https://github.com/dariol83/ccsds
	Only for MAL (see Sec. 2.2): CCSDS MO services – ESA's Java implementation, Java, https://github.com/esa/mo-services-java
	Only for monitoring: SatNOGS Client (see Sec. 2.1), Python, https://gitlab.com/librespacefoundation/satnogs/satnogs-client
	Only for telemetry visualisation: OpenMCT, JavaScript, https://github.com/nasa/openmct
Data Product Management	CCSDS File Delivery Protocol, Python, https://gitlab.com/librecube/lib/python-cfdp
	core Flight System (cFS) CFDP Application (CF), C, https://github.com/nasa/CF
Access Control	CCSDS MO services – ESA's Java impl., Java, https://github.com/esa/mo-services-java

4. Conclusion and future work

Even while space assets continue to grow, the ground segment remains crucial. For our space cybersecurity research, a virtual ground station testbed called SIEM4GS has been under construction. A multivocal but rigorous review of ground station designs was performed, and an abridged version of the review is presented in Sec. 2, informing the design decisions underlying our reference architectures in Sec. 3. The proposed implementation plan can guide any testbed-building effort similar to ours as it involves mostly open-source software building blocks. An alternative clean-slate approach based on model-based systems engineering could have been used for our testbed design, but ours is a security-focused engineering trade-off that builds on open-source contributions. Our mission is to explore new ideas of cyberattacks and countermeasures for space systems starting with the ground segment and within a SIEM framework. Besides pushing the envelope of XDR to achieve both the "detect" and "respond" functions of NIST's Cybersecurity Framework, our long-term plan includes growing this testbed into a *digital twin* where a two-way communication link synchronises the states of a small-scale physical testbed with those of its virtual replica. A journal version of this paper containing an extended version of our literature reviews on ground stations and SIEM (up to UEBA, SOAR, XDR) as well as implementation results, is under development.

Acknowledgement

The authors thank David Culpin (Raytheon Australia), Dr Ronald Mulinde (UniSA), Dr David Ormrod (UniSA) for their initial input; and Brandon Klar (UniSA) for undertaking the implementation.

References

Ahmad, Y.A., Nazim, N.J. and Yuhaniz, S.S. (2016) "Design of a terminal node controller hardware for CubeSat tracking applications", *AEROTECH VI*, IOP Publishing, p. 012031, DOI: 10.1088/1757-899x/152/1/012031.

Amazon Web Services (2021) "AWS Ground Station User Guide", [online], accessed 27 December 2021, https://docs.aws.amazon.com/ground-station/latest/ug/groundstation-ug.pdf.

Barros, A. (2017) "SIEM Correlation Is Overrated", [online], Gartner Information Technology Blog, http://blogs.gartner.com/augusto-barros/2017/03/31/siem-correlation-is-overrated/.

Benecken, Z. (2020) "AMMOS Catalog Version 5.3", Multimission Ground System and Services (MGSS) Program, Interplanetary Network Directorate (IND) Office, NASA, https://ammos.nasa.gov.

CCSDS (2010a) "Mission Operations Reference Model", Recommended Practice 520.1-M-1.

CCSDS (2010b) "Mission Operations Services Concept", Informational Report 520.0-G-3.

CCSDS (2012) "Reference Model for an Open Archival Information System (OAIS)", Recommended Practice 650.0-M-2.

CCSDS (2013) "Mission Operations Message Abstraction Layer", Recommended Standard 521.0-B-2.

CCSDS (2018) "Cross Support Service Management — Simple Schedule Format Specification", Recommended Standard 902.1-B-1.

CCSDS (2019a) "CCSDS Cryptographic Algorithms", Recommended Standard 352.0-B-2.

CCSDS (2019b) "The Application of Security to CCSDS Protocols", Informational Report 350.0-G-3.

CCSDS (2020) "CCSDS File Discovery Protocol (CFDP)", Recommended Standard 727.0-B-5.

Cooklev, T., Normoyle, R. and Clendenen, D. (2012) "The VITA 49 Analog RF-Digital Interface", *IEEE Circuits and Systems Magazine*, vol. 12, no. 4, pp. 21–32, DOI: 10.1109/MCAS.2012.2221520.

Cutler, J.W. and Fox, A. (2006) "A framework for robust and flexible ground station networks", *Journal of Aerospace Computing, Information, and Communication*, vol. 3, no. 3, pp. 73–92, DOI: 10.2514/1.15464.

ESA (2008) "Space engineering: Ground systems and operations", Standard ECSS-E-ST-70C.

Elasticsearch (2021) "Rule types", [online], Kibana Guide, accessed 14 December 2021, https://www.elastic.co/guide/en/kibana/master/rule-types.html.

Exabeam (2021) "The essential guide to SIEM", [online], accessed 14 December 2021, https://www.exabeam.com/siem-guide/.

Fischer, M. and Scholtz, A.L. (2010) "Design of a Multi-mission Satellite Ground Station for Education and Research", *SPACOMM 2010*, pp. 58–63, DOI: 10.1109/SPACOMM.2010.13.

Fortinet (2021) "UEBA", [online], CyberGlossary, accessed 19 December 2021, https://www.fortinet.com/resources/cyberglossary/what-is-ueba.

Gartner (2021) "Security Information and Event Management", [online], accessed 12 December 2021, https://www.gartner.com/en/information-technology/glossary/security-information-event-management.

Guest, A.N. (2017) "Telemetry, Tracking, and Command (TT&C)", *Handbook of Satellite Applications*, 2nd edition, Springer International Publishing, pp. 1313–1324.

Husák, M., Komárková, J., et al. (2019) "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security", *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 640–660, DOI: 10.1109/COMST.2018.2871866.

Islam, C., Babar, M.A. and Nepal, S. (2019) "A Multi-Vocal Review of Security Orchestration", *ACM Comput. Surv.*, vol. 52, no. 2, DOI: 10.1145/3305268.

Jakobson, G. and Weissman, M. (1993) "Alarm correlation", *IEEE Network*, vol. 7, no. 6, pp. 52–59, DOI: 10.1109/65.244794.

Kinman, P. (2021) "Doppler Tracking", *DSN Telecommunications Link Design Handbook*, DSN No. 810-005, 202, Rev. D, Jet Propulsion Laboratory, California Institute of Technology.

Liu, Y., Chen, Y., Jiao, Y., et al. (2020) "A Shared Satellite Ground Station Using User-Oriented Virtualization Technology", *IEEE Access*, vol. 8, pp. 63923–63934, DOI: 10.1109/ACCESS.2020.

Liu, Y., Li, Z., Pan, S., et al. (2021) "Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning", *IEEE Trans. Neural Netw. Learn. Syst.*, early access, DOI: 10.1109/TNNLS.2021.3068344.

Lowdermilk, J. and Sethumadhavan, S. (2021) "The Gestalt: A Secure, High Performance, Low Cost Satellite Ground Station Architecture and its Implementation", *35th Annual Small Satellite Conference*, https://digitalcommons.usu.edu/smallsat/2021/all2021/108/.

Manor, Y. (2021) "Quantifying the Gap Between Perceived Security and Comprehensive MITRE ATT&CK Coverage", industry research report, https://www.cardinalops.com/siem-industry-research-report.

McDonald, G., Hacker, J., Dorame, T., et al. (2021) "Navigating space: A vision for space in defense", white paper from KPMG International and Space Foundation, https://home.kpmg/xx/en/home/insights/2021/08/navigating-space-a-vision-for-space-in-defense.html.

Microsoft (2021) "Azure Orbital: Satellite ground station and scheduling services for fast downlinking of Data", [online], accessed 12 September 2021, https://azure.microsoft.com/en-au/services/orbital/.

Miller, L. (2021) "The Gorilla Guide to Extended Detection and Response (XDR)", ActualTech Media in partnership with Fortinet, https://www.fortinet.com/resources/cyberglossary/what-is-XDR.

NIST (2018) "Framework for Improving Critical Infrastructure Cybersecurity", version 1.1, DOI: 10.6028/NIST.CSWP.04162018.

Normoyle, R. and Mesibov, P. (2008) "The VITA Radio Transport as a Framework for Software Definable Radio Architectures", *SDR 08 Technical Conference and Product Exposition*, SDR Forum.

OMG (2019) "XML Telemetric and Command Exchange Version 1.2", an OMG® XML Telemetric and Command Exchange™ Publication formal/18-10-04, Object Management Group.

Ormrod, D., Slay, J. and Ormrod, A. (2021) "Cyber-Worthiness and Cyber-Resilience to Secure Low Earth Orbit Satellites", *ICCWS 2021*, Academic Conferences International, pp. 257–266, DOI: 10.34190/IWS.21.044.

Palo Alto Networks (2022) "The Essential Guide to XDR", [online], accessed 12 January 2022, https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/ebooks/cortex-ebook_the-essential-guide-to-xdr.pdf.

Pang, G., Shen, C., Cao, L. and Van Den Hengel, A. (2022) "Deep Learning for Anomaly Detection: A Review", *ACM Comput. Surv.*, vol. 54, no. 2, article 38 , 38 pages, DOI: 10.1145/3439950.

Pavur, J. and Martinovic, I. (2020) "SOK: Building a Launchpad for Impactful Satellite Cyber-Security Research", arXiv preprint arXiv:2010.10872.

Riffel, F. and Gould, R. (2016) "Satellite ground station virtualization: Secure sharing of ground stations using software defined networking", *SysCon 2016*, pp. 1–8, DOI: 10.1109/SYSCON.2016.7490612.

Robert, M., Sun, Y., Goodwin, T., et al. (2015) "Software Frameworks for SDR", *Proceedings of the IEEE*, vol. 103, no. 3, pp. 452-475, DOI: 10.1109/JPROC.2015.2391176.

Salah, S., Maciá-Fernández, G., and Díaz-Verdejo, J.E. (2013) "A model-based survey of alert correlation techniques", *Computer Networks*, vol. 57, no. 5, pp. 1289–1317, DOI: 10.1016/j.comnet.2012.10.022.

Schmitt, M., Diet, F. and Mihalache, N. (2018) "Yamcs for lean Commercial Control Centres: The ICE Cubes Control Centre", *2018 SpaceOps Conference*, DOI: 10.2514/6.2018-2682.

Sharma, B., Pokharel, P. and Joshi, B. (2020) "User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder - Insider Threat Detection", *IAIT 2020*, ACM, DOI: 10.1145/3406601.3406610.

Shin, J., Choi, S.H., et al. (2019) "Unsupervised multi-stage attack detection framework without details on single-stage attacks", *Future Gener. Comput. Syst.*, vol. 100, pp. 811–825, DOI: 10.1016/j.future.2019.05.032.

Surligas, M., Papamatthaiou M., Daradimos, I., et al. (2021) "SatNOGS: Towards a Modern, Crowd Sourced and Open Network of Ground Stations", *GNU Radio Conference*, vol. 2, no. 1.

Telecommunications Industry Association (2021) "front-end processor (FEP)", [online], accessed 30 Dec 2021, http://standards.tiaonline.org/market_intelligence_/glossary/index.cfm?term=%26%23%24C%5BRR%3FN%0A.

Vasisht, D. and Chandra R. (2020) “A Distributed and Hybrid Ground Station Network for Low Earth Orbit Satellites”, *HotNets ’20*, ACM, pp. 190–196, DOI: 10.1145/3422604.3425926.

Vera, T. (2016) “Cyber Security Awareness for SmallSat Ground Networks”, *30th Annual AIAA/USU Small Satellite Conference*, https://digitalcommons.usu.edu/smallsat/2016/TS9GroundSystems/2/.

Weston, S. (2020) “Small Spacecraft Technology State of the Art”, Technical Publication NASA/TP-2020–5008734, Small Spacecraft Systems Virtual Institute, https://www.nasa.gov/smallsat-institute/sst-soa.