Defending the Cosmos: Honeypot-based Adversary Detection and Emulation System (HADES)

Arnold Chan[1], Sam Seo[2] and Ady James[2]
[1]University of South Australia, Australia
[2]Adelaide University, Australia

arnold@xcomspace.com
sam.seo@adelaide.edu.au
ady.james@adelaide.edu.au

Abstract: This paper presents HADES (Honeypot-based Adversary Detection and Emulation System), a modular CubeHoneypot that is a ground-deployable cubesat-scale test bed honeypot architecture for cybersecurity research within space systems. Built using open-source technologies, including NASA's Operational Simulator for Small Satellites (NOS3), NASA's core Flight System (cFS), NASA's Open Mission Control Technologies (Open MCT) and the T-Pot honeypot platform, HADES emulates telemetry and command functions of real satellite subsystems. Unlike existing satellite honeypots, HADES is designed for both educational and adversary behaviour research applications, enabling cost-effective experimentation and threat analysis in controlled, realistic environments. The paper outlines the design methodology, implementation stages, system components and early results from simulated attacks and telemetry operations.

Keywords: Adversary emulation, CubeSat security, Satellite honeypot, Satellite security, Space cybersecurity, Ground-based satellite testbed

1. Introduction

The rising dependence on small satellites, especially CubeSats, in both civilian and defence sectors has created an urgent need to explore their cybersecurity vulnerabilities. Despite this, space systems often remain underrepresented in mainstream cyber research due to their perceived inaccessibility and high cost. The need to detect and analyse adversarial behaviour in these domains is more pressing than ever (Pavur and Martinovic 2022; Boschetti, Gordon and Falco 2022; Sharmin et al. 2025). Several studies have documented the massive growth and evolution of cybersecurity threats that have impacted space industry organisations, resulting in operational and financial losses (Boschetti et al. 2022). Figure 1 summarises key cybersecurity vulnerabilities in satellite systems, outlining major attack vectors and their potential impacts on satellite operations and infrastructure, adapted from Pavur and Martinovic (2022).

Vulnerability	Description
Satellite Uplink Interception	Intercepting/manipulating satellite uplink signals can allow attackers to gain unauthorised access to satellite systems.
GPS Spoofing	Spoofed GNSS signals can mislead satellites or receivers about their location, causing navigational errors and potential system failures.
Denial of Service (DoS) Attacks	Flooding satellite links or exploiting protocol weaknesses disrupts operations and prevents legitimate users from accessing services.
Satellite Firmware Vulnerabilities	Outdated or poorly secured firmware can be exploited to gain control of satellite functions and access sensitive data.
Ground Station Compromise	Attackers targeting ground stations (control/management nodes) can pivot to the wider satellite network and its services.

Figure 1: Vulnerabilities in Satellite Technology

Figure 2 shows the major real-world cyber incidents affecting space systems, highlighting the growing vulnerability of space infrastructure to cyber threats.

Case Study	Description
2014 Cyberattack on USA Weather Satellites (Peters 2014)	This attack disrupted data transmission and highlighted vulnerabilities in environmental satellite systems, reinforcing the need for stronger cybersecurity in national weather infrastructure.
2017 GPS Spoofing in the Black Sea (Goward 2017)	Multiple vessels in the Black Sea reported GPS positions, believed to be caused by deliberate spoofing. The incident raised concerns about satellite navigation vulnerabilities in conflict zones.
2018 NASA JPL Cyberattack (Paganini 2019)	A cyberattack on NASA's Jet Propulsion Laboratory exposed vulnerabilities in network segmentation and access control, allowing attackers to access mission systems and sensitive data.
2021 Attack on US Military Satellite Systems (Trevithick 2021)	Demonstrated the consequences of unauthorised access to satellite control networks, emphasising the importance of encryption, authentication and multi-layered defence for military space assets.
2022 Viasat Satellite Internet Disruption (Boschetti et al. 2022)	A cyberattack during the early phase of the Russia–Ukraine conflict disabled thousands of Viasat KA-SAT modems across Europe, disrupting satellite internet services and critical infrastructure.

Figure 2: Case Studies and Real-World Cyber Incidents in Space Systems

Honeypot is a deception solution that diverts cybercriminals from real targets by creating a fake attack target. It also obtains information about the identities, attack tactics, techniques and procedures to learn more about the intentions of their adversaries. Therefore, building and deploying the honeypot solution is necessary as a defensive strategy (Moric et al. 2025). Prior efforts, such as "HoneySat", have laid the groundwork by exploring software-based satellite honeypots (López-Morales et al. 2025). However, HADES aims to take a step further: creating a realistic ground-deployable CubeSat-sized system that not only simulates telemetry (TM) and telecommand (TC) operations, but also captures and interprets real attack behaviour through exposed services. This dual simulation and emulation approach supports education, security research and future mission planning. This paper identified a lack of consolidated and easily accessible academic literature on cybersecurity within the space sector, making it challenging for researchers and organisations to locate relevant and domain-specific analyses. Furthermore, the characteristics of the next-generation honeypot for space assets still need to be understood and developed (López-Morales et al. 2025). Thus, this research project seeks to determine, assess, and analyse critical research materials in honeypots. This area is vital for leveraging the maximum possibility of a honeypot in the future. Therefore, this study analysed the obstacles to a successful honeypot and how its adaptability can be improved. Furthermore, this project involves researching honeypot simulation with open-source tools to integrate with space assets for learning and their potential reactive capability to security threats.

1.1 Purpose and Justification

Satellite ground stations can collect information from space assets that are in orbit around the Earth. Commercial space infrastructure, including ground systems, is increasing dual-use, serving both civilian and military functions. Because space assets often serve both civilian and military functions, they may become targets during military conflict (Boschetti et al. 2022). Due to this, commercial operators need to carefully plan where to put ground stations to get the most use out of them while also taking into account the risk to their security. In addition to these physical and geopolitical threats, the space sector is heavily dependent on technology, which makes it an ideal target for cyber threats and attacks. As satellite systems and ground control networks become more integrated, the need for effective security safeguards has never been greater. One good idea that has emerged is to use honeypots, which are decoy systems that are intended to lure in, analyse and study malicious activities. This project looks at honeypots usage in the space sector, including their pros, cons and possible usages. While HADES demonstrated the feasibility of satellite honeypots, it remains primarily a software-based emulation. It bridges this gap by combining physical modularity, subsystem simulation and open-source mission software in a compact CubeSat form factor.

1.2 Scope

Honeypots are an established cybersecurity tool that is designed to attract, detect and analyse malicious activity by mimicking real systems. While widely applied to traditional IT networks, Supervisory Control and Data Acquisition (SCADA)/Industrial Control System (ICS) and Internet of Things (IoT) systems, their adoption in the space domain remains limited (Guan et al. 2023). The HoneySat project demonstrated that a virtual satellite system could be constructed with emulated TM and TC links. However, HoneySat lacked hardware interaction and relied heavily on abstracted network behaviour. HADES builds upon this concept by incorporating physical hardware (e.g., Raspberry Pi 5), integrating flight software (cFS) and visualising TM with Open MCT. The inclusion of real service protocols (e.g. SSH, Telnet, HTTP) through T-Pot enhances the system's

interaction fidelity. Recent updates to cFS 2.0 introduced plug-and-play capabilities and improved security handling, offering an ideal platform to simulate modular spacecraft behaviour and command injection scenarios. This project aims at researching and developing an adaptable honeypot solution that uses open-source technologies to simulate genuine TM and TC operations. This experimental project is to simulate part of Stefunko's (2018) idea on the usage of Long Range Wide Area Network (LoRaWAN) gateway as a long-range network communication, which this project will utilise two small, low-cost Raspberry Pi versatile computers. It was simulate, observe and evaluate how attackers act through open services and send parsed intrusion data to a satellite subsystem simulator like NOS3 and flight software like cFS and Open MCT dashboard that shows the system's TM updates and overall state. Also, the simulation was to adapt NASA's (Interplanetary Overlay Network Additional Information - NASA 2023) Interplanetary Overlay Network (ION) Delay/Disruption Tolerant Networking (DTN) to simulate both ground and space segments.

1.3 HADES System Overview

This research proposes HADES to simulate the cyber-physical behaviour of a CubeSat space segment and its mission control centre ground segment. The system supports space cyber security research, including cyber threat emulation, TM monitoring and adversary analysis. HADES enables researchers to simulate satellite-like operations, attract real adversarial behaviour, and observe how cyberattacks impact spacecraft TM and functions without using real on-orbit assets. It bridges traditional honeypot concepts with space mission emulation. Unlike orbital missions, the simulation is entirely ground-based. However, it mirrors real satellite operations, including TM/TC flows, mission control dashboards (NOS3 or Open MCT), cFS spacecraft software, delay-tolerant networking and exposure to real-world cyber threats through honeypot gateways.

1.4 HADES Concept of Operations (CONOPS)

The operations of HADES are divided into mission-style phases and system modes, similar to space mission lifecycle planning. These phases provide structure from system assembly to live threat monitoring and eventual shutdown or upgrade. The purpose of this CONOPS is to describe how HADES is deployed, operated and utilised. Figure 3 outlines operational goals, end-user perspectives, system interactions, constraints and expected outcomes. This structured approach ensured that the system's performance could be validated in a controlled environment before exposure to live threats (T3) and provided a clear path for data analysis (T4).

Phase	Description	Key Objectives
T1 (Phase 1)	System Calibration & Health Checks	The system enters Health Check Mode where: • Telemetry packets are generated • Subsystems report status (simulated battery voltage, temperature, CPU load). • NASA Open MCT connects to the telemetry stream. • Log ingestion to T-Pot is verified. • Intrusion detection logs are tested via simulated benign SSH attempts. Exit Criteria: All subsystems report nominal, Open MCT displays live telemetry and logs are timestamped and stored correctly.
T2 (Phase 2)	Controlled Simulation & Attack Injection	The system enters Simulation Mode. • No real attackers; simulation cyberattacks are performed. • These commands are translated into telecommands (TC) and injected into cFS. • Telemetry responses are visualised in Open MCT (e.g., temperature spike, battery fault).
T3 (Phase 3)	Live Threat Exposure (Operational Phase)	Equivalent to the Operational Phase. The system enters Honeypot Live Mode. • Attackers begin interacting unknowingly with the CubeSat honeypot system. • Telemetry anomalies are linked to attacker commands in real time.
T4 (Phase 4)	Data Analysis and Logging	System enters Analysis Mode and Logging Mode. • MITRE ATT&CK / SPARTA for Space techniques are mapped. • Graphs of attack timing vs. subsystem telemetry deviations are generated.
T5 (Phase 5)	Shutdown, Maintenance or Upgrade	System enters Shutdown Mode. • Services are powered down safely. • cFS flight software is backed up. • Logs stored ≥ 30 days. • Hardware inspected for thermal, power or board degradation.

Figure 3: HADES ConOps Phases

1.5 HADES System Architecture

Figure 4 shows that the HADES system architecture is designed to emulate a realistic satellite-ground system using modular and low-cost components to support cybersecurity research in space systems. At the core of the system is the computing core, which consists of two Raspberry Pi 5 units.

Subsystem	Description
Compute Core	Two Raspberry Pi 5 units: • Raspberry Pi 5 #1: NASA NOS3, NASA OpenMCT ground control simulator, T-Pot Mobile • Raspberry Pi 5 #2: NASA NOS3, NASA cFS satellite simulator, T-Pot Mobile
Power	• 5.1V/5A USB-C power supply
Flight Software and Subsystem Simulation	• NASA NOS3 + cFS + STF running on Raspberry Pi 2
Communication & TM/TC Parsing	• Telemetry generated from subsystem-simulation scripts • Telecommand parsed from honeypot traffic and injected into cFS • Delay/latency simulated via ION-DTN
RF Communication Simulation via LoRaWAN	• SX1303 915 MHz LoRaWAN Gateway Module + HAT on Raspberry Pi • CCSDS 256-bit (32 bytes) TM/TC frames *encapsulated* in LoRaWAN FRMPayload • Optional fragmentation/reassembly based on regional payload limits/data rate • Emulates low-bitrate CubeSat link with real RF loss/latency
Ground Software and Interface Subsystem Simulation	• NASA NOS3 + Open MCT for telemetry visualisation and ground control

Figure 4: HADES System Architecture

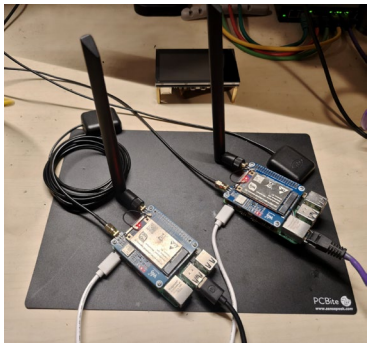

Figure 5: The photo illustrates the HADES setup with two Raspberry Pi

Figures 5 and 6 show that the first Raspberry Pi (on the left) operates as the ground segment which the ground interface is implemented using Open MCT. This provides a web-based interface for mission operators to visualise real-time telemetry, send telecommands, and monitor subsystem performance, similar to professional mission control systems. The second Raspberry Pi (on the right) operates as the space segment, which includes the T-Pot Honeypot, hosting a multi-service deception stack to attract, log and analyse malicious activities targeting satellite infrastructure. It also functions as a cFS satellite simulator, enabling execution of on-board software, telemetry handling, and subsystem emulation in a realistic flight environment.

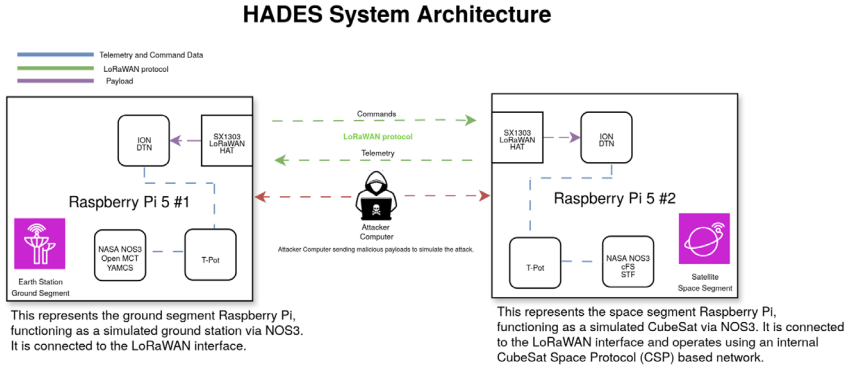

Figure 6: The system architecture diagram shows the HADES system core components and data flows

The system is powered by the 5.1V/5A USB-C power delivery source, which ensures stable and reliable operation of both Pis and connected peripherals. For Flight Software and Subsystem Simulation, cFS is deployed on the second Raspberry Pi. This platform handles essential satellite operation, including telemetry generation, telemetry parsing, command execution, and subsystem behaviours, bringing high fidelity realism

to the honeypot framework. The Communication and TM/TC (Telemetry and Telecommand) parsing subsystem enables bidirectional data exchange between the simulated spacecraft and honeypot services. Telemetry is automatically generated by subsystem simulation scripts, while telecommands originating from captured honey pot traffic are translated, validated, and injected back into the cFS environment. To further increase operational realism, transmission delays and communication latency are modelled using the Interplanetary Overlay Network (ION) Delay Tolerant Networking (DTN) protocol. Together, these components form a functional, scalable, and cyber-deceptive satellite-ground architecture, enabling adversary interaction, telemetry manipulation, and mission-like behaviour within a controlled experimental environment.

1.6 Communication Architecture

The LoRaWAN module is configured to act as a physical-layer substitute for CCSDS (Consultative Committee for Space Data Systems) TM and TC protocols. While CCSDS typically supports 256-bit fixed-length transfer frames, these frames are encapsulated and transmitted using the LoRaWAN Physical Layer (PHY)/Media Access Control (MAC) protocol. The system converts simulated cFS TM packets into CCSDS-format binary frames, then fragments or encapsulates them into LoRaWAN payloads for uplink/downlink transmission. Similarly, TC packets generated from honeypot interactions are encoded into CCSDS 256-bit commands and delivered via LoRaWAN to the Raspberry Pi-based satellite simulator.

The HADES architecture integrates a realistic end-to-end satellite communication environment using commercially available hardware. In addition to the dual-Raspberry Pi compute core and cFS running on Pi 2, the system incorporates an SX1303 915 MHz LoRaWAN Gateway Module with HAT for Raspberry Pi, which enables low-power, long-range communication to emulate space-to-ground data transmission. This approach creates a low-cost, radio-based emulation of real satellite radio frequency (RF) communications, including:

- RF delay, packet loss, and low bitrate, similar to CubeSat S-band/UHF links.
- Encrypted or unencrypted frames, configurable depending on test conditions.
- Support for DTN (Delay/Disruption-Tolerant Networking) layered over LoRa for deep-space simulation, when combined with ION-DTN software.

The LoRaWAN gateway thus becomes a bridge between cyber deception (honeypots) and realistic space communications, allowing researchers to observe how adversaries might interact with real TM/TC links and how malformed or malicious commands propagate through a simulated CCSDS-compliant system.

1.7 HADES Software Architecture

This section describes the approach to the trade studies for the HADES software architecture, including the objectives to be achieved and the criteria based on the decisions. The main goal is to optimise the subsystem for its reliability, cost-effectiveness and compatibility with the extreme environmental conditions of space environment. The following trade study compared different software design approaches for simulation usage, which evaluated the options like integrated software suites, off-the-shelf solutions and open-source software based on criteria such as adaptability, development time, cost and complexity. For the best selected option, it will emerge as the favoured option and will be used for the diverse mission requirements. This involves representing the weighted values as percentages. The cells in the column underneath the option are populated with the grade given to the option. The final percentage at the bottom is determined by multiplying each given grade by the weighted percentage. The average value of the weighted grades is summed up at the bottom of the chart and the option with the highest average value is considered the best option. Figure 7 shows the comparison analysis results and findings as per the commercial versus open-source mission software trade study.

Commercial Software vs Open Source Mission Software Trade Study						
Criteria	**Explanation**	**Grade**	**Weight**	**Integrated Software Suite** (Keszthelyi 2014)	**Off-the-Shelf Software Solutions** (Brownsword, Carney, and Oberndorf 1998)	**Open Source** ("NASA's next Lunar Rover Will Run Open-Source Software | MIT Technology Review," 2021)
Technological Readiness Level	Evaluation of technology's maturity which is based on its testing and proven performance.	10 = high, 5 = medium 1 = low 0 = Fail	20%	8	8	9
Cost Effectiveness	Evaluates and review the financial feasibility of the software design and consider the mission's budget constrants and its long-term value.	10 = high, 5 = medium 1 = low 0 = Fail	15%	7	8	10
Scientific Data Handling	Evaluates and analyse the software capability that it shall be designed to acquire, analyse, process, and transmit scientific data back to Earth.	10 = high, 5 = medium 1 = low 0 = Fail	15%	8	8	9
Scalability Assessment	Evalutes and review the software which it shall be evaluated on its potential to heighten current capacities or enhance operational capabilities	10 = high, 5 = medium 1 = low 0 = Fail	10%	6	5	8
Upgradability Score	Evaluates the software which it shall be capable of upgrading or enhancing post-deployment missions.	10 = high, 5 = medium 1 = low 0 = Fail	10%	7	8	8
Software Complexity	Evaluates the software design which it shall be simple without much complexity to minimise potential points of failure.	10 = high, 5 = medium 1 = low 0 = Fail	15%	7	8	9
Development Time	Evaluates the software design which should take less time to be developed and ready in production usage.	10 = high, 5 = medium 1 = low 0 = Fail	15%	7	9	10
		TOTALS:	**100%**	72.50%	78.50%	91.00%

Figure 7: Commercial vs open-source mission software trade study

The selected option for this project is open-source software because of its high adaptability to the mission changes, ready availability, lowest cost implication and very low complexity, which makes it the best option for the choice. It reflects the overall best fit according to the mission's unique needs and constraints, considering both the pros and cons of each software type. Open-source stands out in this assessment for its balance of readiness, cost and adaptability.

2. Literature Review

2.1 Simulating Adversary Behaviour in Space Systems

The recent research of López-Morales et al. (2025) has highlighted that high-interaction honeypots for satellites have demonstrated an ability to resemble real space systems and entice genuine cyber adversaries. For example, the HoneySat framework is made to make a CubeSat mission look so real that 71.4% of satellite operators who were surveyed said they would have thought it was a real CubeSat if they hadn't been told it was a simulation. In practice, HoneySat succeeded in deceiving human attackers "in the wild"; attackers engaged with the honeypot over extended periods (one stayed connected for two hours) and issued authentic satellite flight software commands, indicating they were convinced by the simulation. In one test, four different intrusions via a Telnet interface resulted in 16 requests unique to satellites from attackers. Importantly, the honeypot environment supports a broad range of attacker tactics and techniques, covering 100% of adversary tactics and 86.8% of known techniques applicable to satellite systems. This comprehensive coverage provides adversaries with lots of opportunities for interaction, closely mirroring real-world attack scenarios. These results collectively show that a CubeSat honeypot can simulate space infrastructure convincingly and capture realistic adversarial behaviour, providing valuable threat intelligence from genuine attacker interactions.

2.2 Technical Constraints in Simulating Satellite Operations and Adversary Attacks

Simulating an entire satellite system with a high level of detail presents considerable technical challenges. One of the major constraints is to mimic the actual spacecraft's complex physical operations and subsystems. For instance, satellites need to know how to control their orientation, power management and deal with temperature changes and other things. One of the main problems with the HoneySat project was replicating all the physics and subsystems (such as orbital motion, attitude changes, and power supply behaviour) that a real satellite environment would need (López-Morales et al. 2025). Achieving this realism can be data-intensive. For instance, the fidelity of the camera simulation depends on image data quality, so simulating a high-resolution Earth observation mission is not feasible without equally high-resolution source imagery. In general, the accuracy of the simulation is constrained by the quality and completeness of available data for sensors and orbits, and any simplification might tip off an attacker.

2.3 Integration of Open-Source Tools for Satellite and Ground Segment Emulation

On the ground segment side, the honeypot uses an open-source mission control software to simulate a satellite operations centre. In the case of CubeSat Space Protocol (CSP) missions, by presenting attackers with

familiar tools and interfaces (a web dashboard, a terminal for the ground station, tracking software, etc.), all based on real open-source software, the honeypot environment feels legitimate. For the space segment and communications, open-source technologies are similarly integrated. The López-Morales et al. (2025) HoneySat uses libCSP, the open-source implementation of CSP, to format and route packets as a real CubeSat would. Actual CubeSat flight software is reused when possible: for example, the cFS, which is open-source and has flown on multiple CubeSat missions, is used as the satellite's onboard software (McComas and Tandy 2022). This provides realistic command-handling and telemetry generation logic, complete with a library of standard satellite commands (dozens of typical operations like reboots, sensor readings, etc., which are parsed from CSP packets). The honeypot's modular design makes it possible to swap or extend these components to emulate different mission architectures. In one case study, the HoneySat team demonstrated integration of a completely different open-source ecosystem: they added support for the CCSDS space protocol (commonly used by larger satellites). This was done by incorporating Yet Another Mission Control System (Yamcs), an open-source mission control framework that supports CCSDS TM/TC standards (Sela, Mihalache and Moreau 2012). The honeypot seamlessly simulated a different class of mission without starting from scratch. In summary, open-source tools are integral at every level, providing the building blocks (mission control software, satellite OS libraries, network simulators, tracking tools and databases) that can be integrated and configured to emulate a realistic satellite-ground system, all without proprietary hardware or software. The literature review identified a lack of prior research that formally conducted or documented a comparative trade study of mission software options. This gap in the existing literature informed the subsequent selection of an open-source software design.

2.4 Logging of Adversary Activity

Collecting detailed logs of attacker interactions is central to a honeypot's value, but making sense of these logs is equally important. An effective CubeSat honeypot should log, parse and present attack data in ways that both security researchers and satellite system engineers find meaningful. On the satellite side, the flight software and each simulated subsystem also feed into the logging system, so one can trace how a malicious command propagates. For security researchers, the logs can be parsed to extract the attackers' Tactics, Techniques and Procedures (TTPs). The goal is to bridge the gap between security and engineering: security researchers gain a clear view of how attackers operate in a space system (which attack vectors they choose, how far they get, what tactics they employ) and satellite engineers gain an understanding of how their systems might behave under malicious pressure (which subsystems are most at risk and how an attacker's inputs translate into system actions). By logging every interaction with context and timing, and by using common data formats, a CubeSat honeypot makes it possible to analyse and visualise attacks in a comprehensive yet accessible way. This ensures that the insights gleaned are actionable for improving both cybersecurity strategies and the robust design of satellite systems.

3. Methodology

The research methodology adopted in the HADES project involved a review of existing literature, such as López-Morales et al. (2025) on honeypot-related space sector usage. The selected research methodology in this paper aimed to address the research questions presented earlier by exploring methods for simulating the space systems logs and adversary malicious activities logs. It will evaluate how the honeypot deception will be introduced into space systems for their potential reactive capability to security threats. This methodology aimed to address research questions by exploring methods and evaluating how open-source deception technology could be integrated into space systems for enhanced security. Figure 8 illustrates the systematic steps of this research, represented in a flowchart diagram.

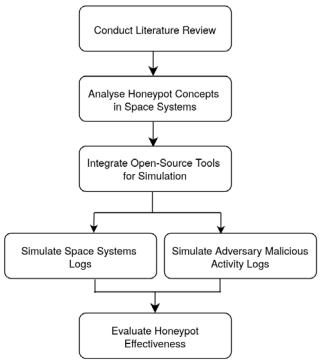

Figure 8: Flow chart diagram representing the steps of the research methodology for HADES

The core methodological decision involved conducting a trade study of mission software options, which ultimately led to the selection of an open-source software design. The literature review indicated that no previous research has formally conducted or documented such a trade study and this identified gap directly informed the selection of an open-source software design. This selection was justified by its high adaptability, ready availability, lowest cost, and very low complexity compared to commercial and integrated software suites, as demonstrated in Figure 3 trade study. The HADES architecture integrated several critical open-source components, as shown in Table 4. The methodology conceptually included the integration of an SX1303 LoRaWAN Gateway Module to emulate the physical layer of the low-bitrate CubeSat radio frequency (RF) link, encapsulating CCSDS TM/TC frames for uplink/downlink transmission (CCSDS 2012; Araújo 2023).

4. Findings

The HADES system was successfully deployed and demonstrated its ability to emulate the operational behaviour of a CubeSat-style space-ground architecture while capturing hostile activity. However, it is deployed separately for separate functions, not working simultaneously at the same time. Some of the requirements are still to be addressed. Core components, including cFS, Open MCT, NOS3 simulation environment and the T-Pot honeypot framework, operated as intended separately. The system was able to generate and visualise TM data, simulate late TC and collect malicious network interactions directed at the honeypot interface. Open MCT successfully replayed telemetry deviations associated with simulated attack scenarios, allowing the effects of cyber interactions to be observed in a mission control-style interface in NOS3. The T-Pot honeypot stack captured associated log data. However, the intended RF simulation using LoRaWAN could not be fully implemented due to unresolved configuration issues at the physical and MAC layers. As a result, all telemetry and telecommand exchanges had to be done over a wireless ground-based network instead of RF channels.

5. Recommendations and Future Works

Further development of the HADES platform should prioritise the completion of the LoRaWAN-based RF link to enable the transmission of CCSDS-compliant TM frames over a realistic radio channel. This would allow the system to better simulate the effects of latency, link degradation and signal interference present in Low Earth Orbit (LEO) communications. For future research, we should also explore the deployment of HADES across multiple distributed honeypot nodes to form a swarm-based architecture via LoRaWAN, which is to enable the study of sophisticated attacks against space constellations. In addition, linking the test bed to software-defined radio ground stations would allow partial exposure to space like radio frequency environment. The honeypot supports mapping events to known attack techniques; researchers could correlate log entries with frameworks like the US's MITRE ATT&CK (Hong 2023), Aerospace's SPARTA (Ear, Bailey and Xu 2024) or the European Space Agency (ESA)'s SPACE-SHIELD (European Space Agency n.d.) matrix of space cyber tactics. In practice, the HADES can be used to collect data to identify which attack techniques were attempted by adversaries and to confirm the coverage of their honeypot. Meyers (2025) highlights recent developments of AI-powered on-orbit intrusion detection systems that are capable of monitoring satellite telemetry in real time. Any structured attack data acquired through HADES could also form the basis of an open-source space cyber threat intelligence dataset to aid the space industry and academic benefits. In summary, while the current implementation is limited to ground-based experimental only, the HADES can aid as a foundation for satellite cyber deception research and can be a potential tool to be used for space security training, threat intelligence and mission assurance in the space sector.

6. Discussion and Conclusion

The results indicate that the HADES prototype is capable of attracting genuine adversarial behaviour and reproducing key elements of satellite-ground communication workflows using affordable, open-source technologies. The fact that attackers engaged with the system using real-world techniques such as default credential brute forcing, directory traversal and basic reconnaissance suggests that even a ground-based emulation of a satellite system is sufficient to trigger realistic adversary interest. This validates the underlying premise that deception technologies can be effectively adapted to the space domain.

The resilience of cFS when subjected to malformed TM packets indicates that NASA's flight software architecture already exhibits robustness against certain classes of protocol fuzzing. However, it must be acknowledged that no authentication or encryption mechanisms were applied to the telemetry and telecommand interfaces in this prototype. This omission mirrors existing vulnerabilities identified in legacy satellite infrastructure and emphasises the need for secure uplink/downlink mechanisms in future implementations.

Although the system succeeded in simulating subsystem interactions and attacker impacts on telemetry, the absence of a functioning LoRaWAN communication link meant that the testbed did not fully replicate the bandwidth constraints, latency variations or packet loss typically observed in real space-based radio-frequency environments. Likewise, no real-time alerting, automated anomaly detection or adversarial artificial intelligence behaviours were implemented. Consequently, HADES currently operates as a passive data collection system rather than an autonomous defensive architecture.

In summary, this HADES project demonstrates that ground-based, modular honeypots can simulate realistic CubeSat systems and attract meaningful attack behaviour. As threats toward space systems increase, such deployable, interactive testbeds offer critical insights for mission designers, researchers and operators alike. The system's reliance on open-source components ensures affordability, reproducibility and accessibility for further academic research.

Acknowledgements

The authors would like to express their heartfelt gratitude to the University of South Australia for providing library access and all the necessary resources that enabled the successful completion of the paper.

AI Declaration: Grammarly AI was used only for grammar checking and improving clarity and readability. It did not contribute to the research design, analysis, interpretation or conclusion of the paper, which are solely the authors' work.

Ethical Statement: The research involves only simulated cyber-attack scenarios and controlled environments, therefore formal ethical approval was not required.

References

Araújo, A. C. M. (2025). *Analysis of the application and optimization of CCSDS telemetry and telecommand standards for CubeSat missions*. ResearchGate. Available at: https://www.researchgate.net/publication/390095312_Analysis_of_the_Application_and_Optimization_of_CCSDS_Telemetry_and_Telecommand_Standards_for_CubeSat_Missions

Boschetti, N., Gordon, N., Sigholm, J. and Falco, G. (2022) 'Commercial space risk framework: assessing the satellite ground station security landscape for NATO in the Arctic and High North', *Proceedings of the IEEE Military Communications Conference (MILCOM 2022)*, pp. 679–686. Available at: https://doi.org/10.1109/MILCOM55135.2022.10017538.

Boschetti, N., Gordon, N. and Falco, G. (2022) *Space cybersecurity lessons learned from the ViaSat cyberattack*. Available at: https://www.researchgate.net/publication/363558808_Space_Cybersecurity_Lessons_Learned_from_The_ViaSat_Cyberattack.

Brownsword, L., Carney, D. and Oberndorf, T. (1998) *The opportunities and complexities of applying commercial-off-the-shelf components*. Available at: https://nepp.nasa.gov/DocUploads/8F67CC1B-1C86-42A0-ACFAE7328AE71A4B/The%20opportunities%20and%20complexities%20of%20applying%20commercial%20off%20the%20shelf%20components.pdf.

Cisar, P. (2025) 'The place and role of honeypot solutions in network intrusion detection systems', *IPSI Transactions on Internet Research*, 21(2), p. 107. Available at: https://doi.org/10.58245/ipsi.tir.2503.11.

Consultative Committee for Space Data Systems (CCSDS) (2012) *Introduction to CCSDS space data system standards*. CCSDS 100.0-G-2, Washington, DC. Available at: https://ccsds.org/publications/bluebooks/

Ear, E., Bailey, B. and Xu, S., 2024. Towards principled risk scores for space cyber risk management. arXiv preprint arXiv:2402.02635. Available at: https://arxiv.org/abs/2402.02635.

European Space Agency (ESA) n.d. *SpaceSHIELD: Space Attacks and Countermeasures Engineering Database*. Available at: https://spaceshield.esa.int/.

Goward, D. (2017) 'GPS spoofing incident points to fragility of navigation satellites', *National Defense*, 102(766), pp. 18–19. Available at: https://www.jstor.org/stable/27021938.

Guan, C., Liu, H., Cao, G., Zhu, S. and Porta, T.L. (2023) 'HoneyIoT: adaptive high-interaction honeypot for IoT devices through reinforcement learning', *arXiv preprint*, arXiv:2305.06430. Available at: https://doi.org/10.48550/arXiv.2305.06430.

Hong, J.-K. (2023) "Efforts against Cybersecurity Attack of Space Systems," *Journal of Positioning, Navigation, and Timing*. 항법시스템학회, 12(4), pp. 437–445. doi: 10.11003/JPNT.2023.12.4.437. Available at: https://koreascience.or.kr/article/JAKO202335557814416.page.

Keszthelyi, L. (2014) 'Utilizing the Integrated Software for Imagers and Spectrometers (ISIS) to support future missions', *Lunar and Planetary Science Conference*, 45. Available at: https://www.researchgate.net/publication/303137594_Utilizing_the_Integrated_Software_for_Imagers_and_Spectrometers_ISIS_to_support_future_missions.

López-Morales, E., Planta, U., Marra, G., González, C., Hopkins, J., Garoosi, M., Obreque, E.,

McComas, D. and Tandy, W. (2022) *Making Space for NASA's Core Flight System Applications.* In: Proceedings of the Small Satellite Conference 2022, Utah State University, Logan, UT, Technical Poster Session 3, 10 August 2022. Available at: https://doi.org/10.26077/m6mw-8930.

Meyers, J. (2025) *AI-powered on-orbit intrusion detection becomes a new line of defense*. Kratos Space. Available at: https://www.kratosspace.com/constellations/articles/ai-powered-on-orbit-intrusion-detection-becomes-a-new-line-of-defense.

Morić, Z., Dakić, V. and Regvart, D. (2025) 'Advancing cybersecurity with honeypots and deception strategies', *Informatics*, 12(1), p. 14. Available at: https://doi.org/10.3390/informatics12010014.

NASA (2023) *Interplanetary Overlay Network – additional information*. Available at: https://www.nasa.gov/directorates/somd/space-communications-navigation-program/interplanetary-overlay-network/.

Paganini, P. (2019) *NASA hacked! A Raspberry Pi connected to its network was the entry point*. Security Affairs. Available at: https://securityaffairs.com/87466/hacking/nasa-jpl-hacked.html.

Patel, N.V. (2021) *NASA's next lunar rover will run open-source software*. MIT Technology Review. Available at: https://www.technologyreview.com/2021/04/12/1022420/nasa-lunar-rover-viper-open-source-software/.

Pavur, J. & Martinovic, I. (2022) *Building a launchpad for satellite cyber-security research*. Cybersecurity, 8(1), tyac008. Available at: https://academic.oup.com/cybersecurity/article/8/1/tyac008/6611670

Peters, S. (2014) *NOAA blames China in hack, breaks disclosure rules*. Dark Reading. Available at: https://www.darkreading.com/cyberattacks-data-breaches/noaa-blames-china-in-hack-breaks-disclosure-rule.

Prokop, L.E. and Wilmot, J.J. (2014) *NASA's core flight software: a reusable real-time framework*. NASA Technical Reports Server. Available at: https://ntrs.nasa.gov/citations/20140017040.

Rubio-Medrano, C. and Abbasi, A. (2025) *HoneySat: a network-based satellite honeypot framework*. arXiv preprint. Available at: https://doi.org/10.48550/arXiv.2505.24008.

Sela, A.F., Mihalache, N. and Moreau, D. (2012) *YAMCS – A Lightweight Open-Source Mission Control System*, presented at the SpaceOps 2012 Conference. American Institute of Aeronautics and Astronautics (AIAA). Available at: https://doi.org/10.2514/6.2012-1280790.

Sharmin, A., Mahmud, B.U., Nabi, N., Shaima, M. & Faruk, M.J.H. (2025) 'Cyber Attacks on Space Information Networks: Vulnerabilities, Threats, and Countermeasures for Satellite Security', *Journal of Cybersecurity and Privacy*, 5(3), p. 76. Available at: https://doi.org/10.3390/jcp5030076.

Stefunko, S. (2018) *Honeypot for wireless IoT networks*. Available at: https://netmon.fit.cvut.cz/theses/stefusim-2019.pdf.

Trevithick, J. (2021) *U.S. satellites are being attacked every day according to Space Force general*. The War Zone. Available at: https://www.twz.com/43328/u-s-satellites-are-being-attacked-everyday-according-to-space-force-general.