SMARTSAT
COOPERATIVE RESEARCH CENTRE

TECHNICAL REPORT 9

Development of an Evil Digital Twin for LEO Small Satellite Constellations

Part Two of Two

This report should be cited as:
SmartSat 2021, Development of an Evil Digital Twin for LEO Small Satellite Constellations, SmartSat Technical Report No.9 SmartSat, Adelaide, Australia.

Acknowledgement:
SmartSat acknowledges the contribution made by Dr David Ormrod, towards the writing and compilation of this technical report.

Executive Summary

This report is the second of two parts commissioned by SmartSatCRC through the University of South Australia, seeking to enhance the state of the art in cyber-security solutions for Low Earth Orbit (LEO) space systems. The aim of this two-part series is to establish a Cyber-Jeopardy and Response (CY-JAR) Concept for ongoing development and subsequent deployment into the LEO space operational environment. The first part of the report provided an overview and analysis of the body of knowledge pertaining to the concept of the evil twin, and the supporting concepts of risk, resilience, and cyber-worthiness as a means of enhancing the security posture of LEO systems. This second part of the report provides a fully worked example of a cybersecurity solution, using a generic model of a LEO space system, as a precursor to the CY-JAR concept.

The two report parts will be combined and enhanced as a third deliverable to be provided for future publication (October 2021). The subsequent third report will constitute the final report, of both parts. Additional content and integration work will be conducted for deliverable three.

Space systems frequently employ the concept of a digital twin to test engineering concepts in a simulated environment which replicates the functionality of the system in question. Digital twins can have different fidelity levels, designed for different purposes. This report introduces the concept of an ‘evil twin’ as a counterpart to the commonly utilised digital twin. The evil twin models and tests potential attacks by adversaries, to improve cyber-security outcomes. This approach builds upon the practice of threat modelling and red teaming, with the goal of enhancing the resilience of space systems and improving their survivability under cyber-attack. The evil twin is more than just a penetration test or a red-team exercise; it is intended to be a comprehensive methodology which matches the utility of a traditional digital twin in the reduction of risk to space missions.

Contents

0. Introduction – Modelling the Evil Twin

0.1. Contribution

This paper seeks to demonstrate the modelling of the Evil Digital Twin to build a capability for creation of a nascent Cyber Jeopardy and Response (CY-JAR) capability. The report should be read with both parts, which are to be integrated in deliverable three to achieve the following objectives:

Vision: Enhance LEO Space Vehicles (SV) resilience and reduce risk for Australian LEO operations through the effective application of an Evil Digital Twin, cyber-worthiness framework and CY-JAR model.

Enabler Activities – LEO SV operators will:

- identify all Advanced Persistent Threats (APTs) and any other cyber actors reported through open-source means which have the capability and intent to attack LEO systems. These actors will form a threat actor library.
- identify the Tactics, Techniques and Procedures (TTPs) employed by the threat actors included in the library.
- identify crown jewels, critical systems and assets for protection using a mission and effects-based space mission assurance process. Security controls and efforts will be prioritised to protect these systems and assets.
- undertake cyber threat intelligence monitoring, vulnerability management, threat modelling, penetration testing and research to maintain situational awareness of contemporary threats and forecast future trends.
- report breaches and share intelligence sources to ensure a secure ecosystem for all space systems.
- develop a cyber jeopardy and response capability within constellations, to provide contextualised space domain awareness through the use of anomaly attribution and intelligent sense making as a defensive function.

0.2 The Evil Digital Twin Methodology (EDTM)

The Evil Digital Twin Methodology (EDTM) is a hybrid method, built upon the literature and key concepts presented within part one of this report. The EDTM utilised in this report incorporates twenty steps:

1. Determine Scope and Collect intelligence.
2. Develop a Threat Library.
3. Develop Threat TTP Matrices.
4. Collect architectural and system information of space systems and assets.
5. Build a Digital Twin.
6. Develop vulnerability models for all space systems.
7. Conduct Threat Modelling.
8. Record a Baseline.
9. Develop an Initial Mission Resilience Sub-System Crosswalk.
10. Develop a Crown Jewels and Mission Assessment.
11. Conduct Impact Analysis.
12. Conduct security testing using the Digital Twin.
13. Conduct Countermeasure Research and Analysis.
14. Conduct Desktop Quantitative Resilience Assessment.

15. Undertake a Cyber-worthiness Design Principles Review.
16. Improve and update the system.
17. Conduct security testing using Digital Twin.
18. Record a new Baseline.
19. Undertake Risk Governance Review.
20. Iterate.

The report structure mirrors each of the steps within the EDTM.

1. Determine Scope and Collect Intelligence

Determining scope is important for the efficient focus of limited time and resources to support intelligence collection, risk assessments and efforts to improve cyber-security posture. The collection of intelligence involves the development of a collection plan, with formalised collection methods and procedures to both protect the collection team and to protect the integrity of the data collected. Intelligence sharing arrangements and tools should be agreed, including external data sources and internal procedures. Intelligence gaps should be identified for further development. Intelligence collection should include threat and friendly data relevant to securing the LEO systems. Priority Intelligence Requirements (PIRs) should be developed, sorted, catalogued, and answered. Continuous intelligence collection approaches should be established based on the lifecycle of both threats and defended systems.

The scope of the proof-of-concept example provided within this paper is limited to payload vulnerabilities as depicted in Figure 1. To reduce complexity, vulnerabilities associated with terrestrial systems and electronic warfare capabilities have not been included in this assessment. However, these should be incorporated when conducting a full assessment of the vulnerabilities of a real system.

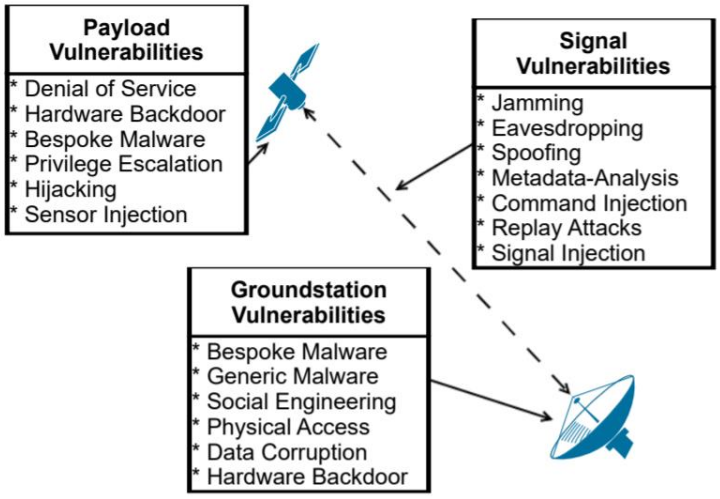

Figure 1 - Satellite Vulnerability Classifications (Pavur & Martinovic, 2020).

PIRs have the following attributes:

- They ask one question.
- They focus on a fact, event, or activity.
- They enable a single decision (Abbany, 2018; United States Army, 1994).

For this proof-of-concept report the following PIRs have been developed:

1. What objectives are adversaries likely to seek to achieve by attacking our LEO SV?
2. What LEO SV systems are adversaries likely to target to achieve their objectives?
3. What vulnerabilities are known to exist in the LEO SV systems, subsystems and software packages?

4. What tactics, techniques and procedures are adversaries likely to use to achieve their objectives?
5. What network and host artefacts are associated with adversary infrastructure?
6. What domain names are associated with adversary infrastructure?
7. What IP addresses are associated with adversary infrastructure?
8. What hash values are associated with likely adversary tools?

A focused collection activity was subsequently conducted to support the development of answers to these questions. The results of this collection activity inform subsequent sections of this report.

2. Develop a Threat Library

Conduct an Adversary Threat Assessment and develop a Threat Library of all threat actors of interest with information available through threat reporting, including tool sets and malware employed which could be a significant threat to any component of the space system.

Different threat actors will be identified as threats specific to the defender's organisation and their network architecture throughout the EDTM. The creation of a threat library is intended to support the efficient use of limited resources, by narrowing scope and supporting prioritisation of defensive efforts. The threat library does not seek to reduce threats in a belief that an actor who is not in the library cannot or will not attack the network. Rather, it uses a risk-based, intelligence-informed process to support understanding of the highest known risk and build a defensive posture from this baseline.

Threat actors in the threat library should range across the spectrum of potential adversaries. This spectrum has been described by a United States Department of Defense, Defense Science Board Task Force consisting of six threat actor tiers. "*Tiers I and II attackers primarily exploit known vulnerabilities; Tiers III and IV attackers are better funded and have a level of expertise and sophistication sufficient to discover new vulnerabilities in systems and to exploit them; and Tiers V and VI attackers can invest large amounts of money (billions) and time (years) to actually create vulnerabilities in systems, including systems that are otherwise strongly protected"* (United States Department of Defense, 2013). A depiction of the threat actor tiers is provided in Figure 2.

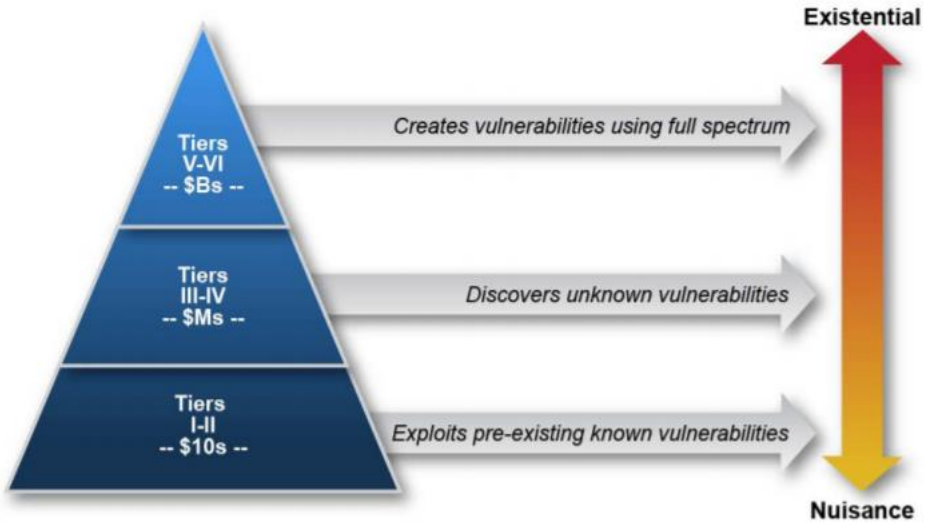

Figure 2 - Threat Actor Tiers (United States Department of Defense, 2013)

Prior incidents and historical data related to the specific industry and organisation in question should be considered in detail. For example, a LEO SV providing a telecommunications service may face different threat actors to a SV supporting a Defence service. The likely objectives of the threat actor are a critical consideration when building a threat library, as this informs what actions and TTPs are performed on the system and support risk management decisions by defenders.

Turla is an Advanced Persistent Threat (APT) actor who has been associated with many different cyber-attacks on critical and national infrastructure (MITRE, 2021b). A variety of information is available on this specific threat actor, an example is depicted in Figure 3.

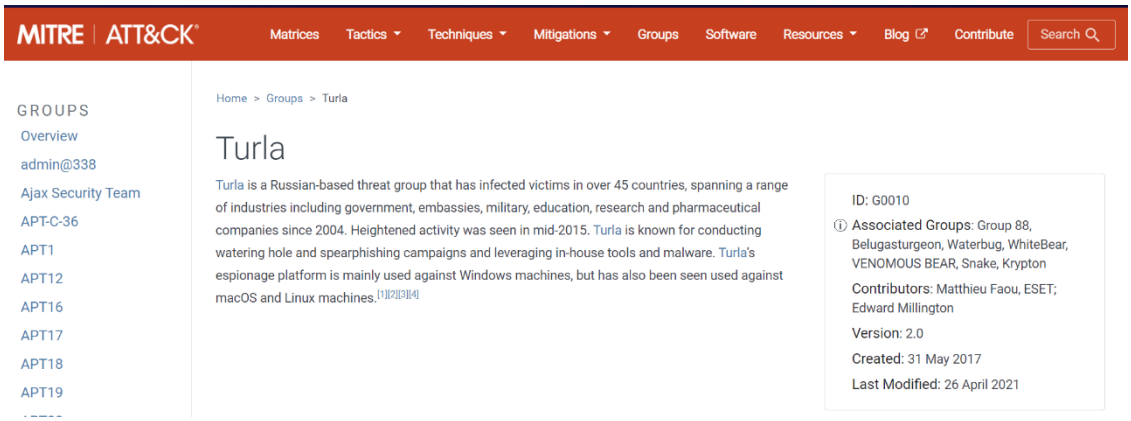

Figure 3 - MITRE Assessment Regarding the Turla Group (MITRE, 2021)

For example, the Thailand Computer Emergency Response Team (ThaiCERT) maintains a history of Turla activity, including their employment of a new malware loading tool in 2021 (ThaiCERT, 2021). This type of repository assists in understanding if the threat actor remains active, who their current targets are and supports future steps of building vulnerability models and the conduct of threat modelling. This will be expanded upon later in this report.

APTs 1, 5 and 14 have been reported by FireEye to target satellite systems (Fireeye, 2021). These suspected Chinese-based APTs have employed a variety of TTPs to obtain information on satellite systems, compromise telecommunications providers, and infect embedded technologies. An example of the APT 5 reporting by FireEye is depicted in Figure 4.

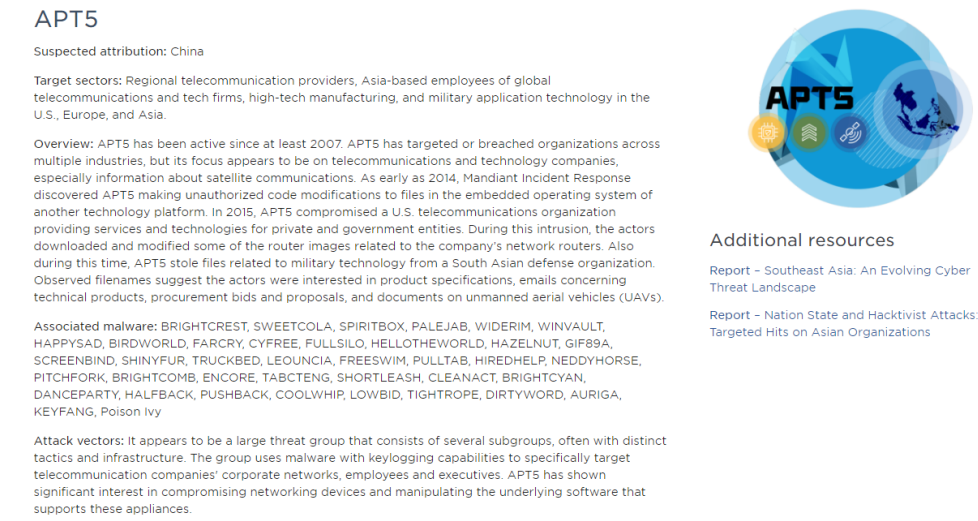

Figure 4 - FireEye Reporting on APT5

3. Develop Threat TTP Matrices

Using the MITRE ATT&CK Framework and other adversary behaviour models, build Threat TTP Matrices. Undertake further research on TTPs within the matrices.

Threat scenarios can be used to example specific methods and techniques to attack vulnerable systems, components, and software to achieve specific adversary missions. An example of simple threat scenarios for satellite systems is provided by Abbany (2018).

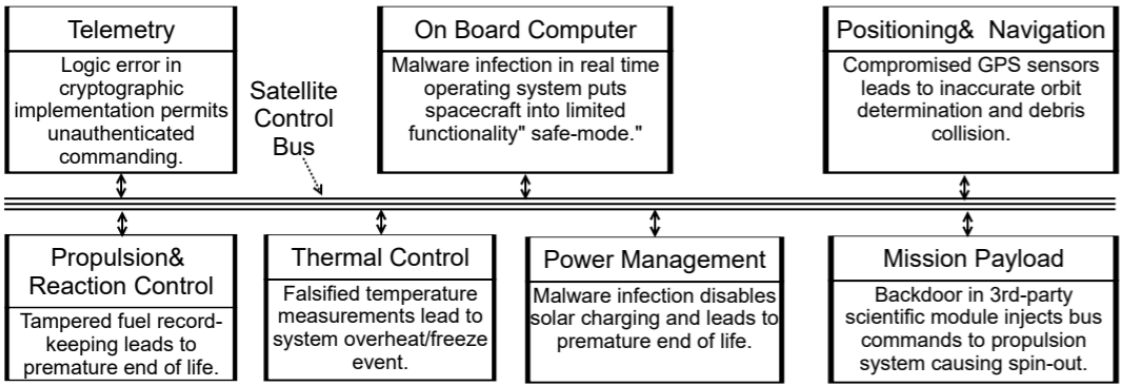

Figure 5 - Example Threat Scenarios for Satellite Systems (Abbany, 2018)

However, the reality of a cyber-attack is that TTPs are chained together based on vulnerabilities and the target architecture to achieve adversary missions. They do not normally emerge in the same way as they are represented in Figure 5. It is rare that initial access will lead immediately to complete exploitation. An example of this type of attack is demonstrated in the TTP matrix in Table 1, where a series of tactics are chained together, moving from initial access to complete exploitation and then victim impact. In this example, each tactic from initial access is used to reinforce the next; progressively providing administrative privileges, enabling lateral movement between systems and persistent control until the attacker's specific objectives have been achieved.

Initial Access	Execution	Persistence	Privilege Escalation	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Operating Mode	Modify Program	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Engineering Workstation Compromise	Execution through API	Project File Infection		Indicator Removal on Host	Remote System Discovery	Lateral Tool Transfer	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
Exploit Public-Facing Application	Graphical User Interface	System Firmware		Masquerading	Remote System Information Discovery	Program Download	I/O Image		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Exploitation of Remote Services	Hooking	Valid Accounts		Rootkit	Wireless Sniffing	Remote Services	Man in the Middle		Block Serial COM	Unauthorized Command Message	Loss of Control
External Remote Services	Modify Controller Tasking			Spoof Reporting Message		Valid Accounts	Monitor Process State		Data Destruction		Loss of Productivity and Revenue
Internet Accessible Device	Native API						Point & Tag Identification		Denial of Service		Loss of Protection
Remote Services	Scripting						Program Upload		Device Restart/Shutdown		Loss of Safety
Replication Through Removable Media	User Execution						Screen Capture		Manipulate I/O Image		Loss of View
Rogue Master							Wireless Sniffing		Modify Alarm Settings		Manipulation of Control
Spearphishing Attachment									Rootkit		Manipulation of View
Supply Chain Compromise									Service Stop		Theft of Operational Information
Wireless Compromise									System Firmware		

Table 1 - Example TTP Matrix with Selected Techniques

The creation of different TTP chains and matrixes aligned to the actors within the threat library provide a means of supporting analysis in future steps to prioritise cyber-defence efforts and toolsets. At this point, the TTP matrix and overall threat library is limited by the overall scope of the organisation and systems to be defended. However, specific details of the architecture are not incorporated at this time. By keeping a broader scope, the threat library and TTP matrices are more maintainable and retain their utility as architectures change and new vulnerabilities are identified.

An example of one type of attack associated with the Turla group is satellite internet hijacking, which has historically focused on African satellite targets to obtain cheap

anonymised internet bandwidth and a new command and control domain as infrastructure for other cyber-attacks (Tanase, 2015). This type of attack is depicted in Figure 6.

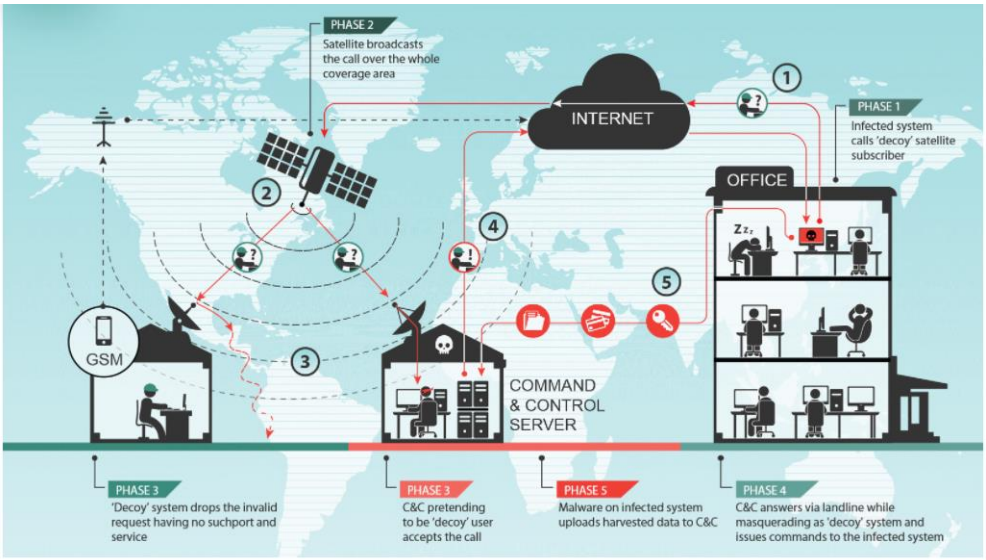

Figure 6 - Example of Satellite Internet Hijacking (Tanase, 2015)

APT TTPs can be collated into a profile, akin to a fingerprint, which can support attribution and threat hunting efforts. If a cyber-defender understands who it is attacking them, they can increase the odds not only of detection, but also thwarting the success of the adversary. For example, if a defender understands how persistence is achieved, efforts to eject the APT from the network may be more efficient. An example of the Turla ATT&CK Techniques collated by MITRE (2021a) and depicted in Figure 7.

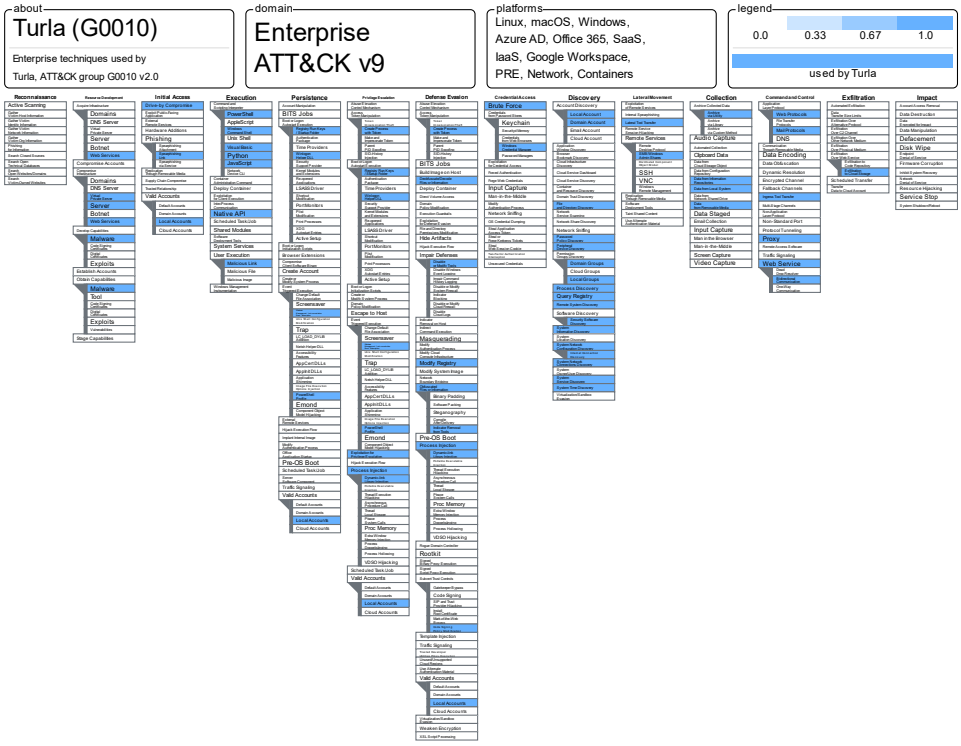

Figure 7 - Turla ATT&CK Techniques (MITRE, 2021a)

4. Collect architectural and system information of space systems and assets

Develop a clear and consistent view of the system architecture and develop bill of materials pertaining to software and hardware. Document relevant protocols and networking connections.

The example architecture considered within this report consists of the following components:
1. BeagleBone Black running KubOS;
2. Jetson Xavier NX running Jetpack 4.4;
3. Sony Spresense Camera; and
4. LoRa RF System - Adafruit Feather M0 RFM69HCW Packet Radio.

The above hardware components contain numerous software packages and protocols. A subset is examined below. A full analysis has not been conducted in the interests of managing the complexity of this report. However, a full report is recommended for a live system, prior to deployment, as part of a standard DevSecOps (Development, Security and Operations) approach.

4.1. Beaglebone Black

The BeagleBone black is a community supported, low-cost development platform. It has a wide range of uses and is compatible with several software packages including Debian, Android and Ubuntu (BeagleBoard.org Foundation, 2021a). The Github BeagleBone Black branch contains design and document files (BeagleBoard.org Foundation, 2021b). As an open-source project, the code is broadly available for analysis and security review. However, this provides an opportunity for specific exploit packages to be developed with greater ease. The BeagleBone black has been widely adopted throughout the world, as is evident through several tutorials which can be found online, including the Chinese social media site QQ.

4.1.1. KubOS

KubOS has been referred to as the *"android of space systems"* and is partnered with RUAG (Frost, 2019), an international provider of radiation-hardened and fault-tolerant space systems such as the Next Generation On Board Computer (RUAG, 2020). KubOS is documented to be onboard systems such as the Educational Irish Research Satellite 1 (EIRSAT-1) (Doyle et al, 2020) and is listed as a component system of the Cal Poly CubeSat Laboratory (Cal Poly CubeSat Laboratory, 2021).

KubOS is designed for satellite developers leveraging multiple open-source projects and utilising a custom framework and Software Development Kit (SDK) to support a variety of hardware services (Kubos Corporation, 2020). KubOS provides a typical deployment architecture, which is intended for integration into the Major Tom cloud-based ground station service and primarily aimed at the CubeSat market (Kubos Corporation, 2021c; Miranda, Ferreira, Kucinskis, & McComas, 2019). The 2020 NASA State-of-the-Art Small Spacecraft Technology report listed Major Tom as a Technical Readiness Level (TRL) of 8+, indicating that it is a mature space technology system offered by a capable provider (NASA, 2020). Other sources of information on KubOS include Github (Kubos Corporation, 2021a) and the Slack KubOS Community (Kubos Corporation, 2021b). The slack community also provides

information on system users and their use cases, which is important for both system support and intelligence collection.

4.1.2. KubOS Packages and Architecture

The general architecture provided for Kubos is depicted in Figure 8.

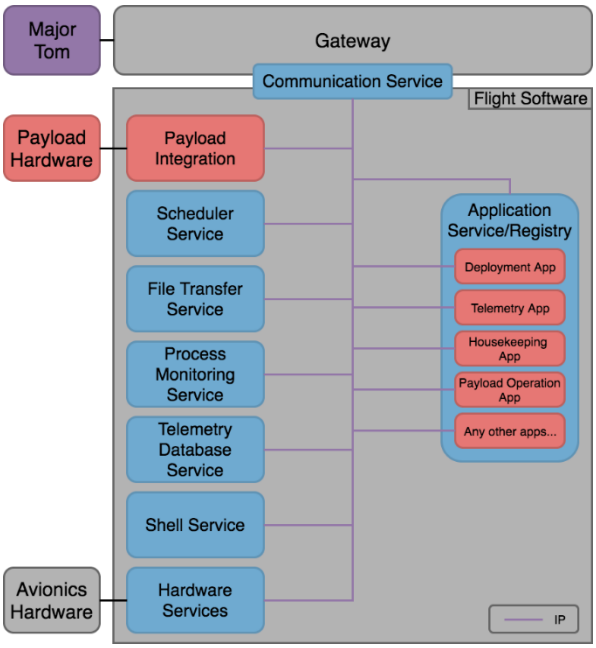

Figure 8 - KubOS General Architecture

Some of the core software packages deployed on the testbed BeagleBone Black using KubOS, including the deployed software versions and currently available versions, are described in Table 2. This is a simple example and is not comprehensive, as the full list of packages is large. The process of decomposition of software systems into their component parts and packages is critical information to enable effective vulnerability analysis throughout the EDTM process and described later in this report. The documentation and configuration management of software to capture versioning is important to support patching and vulnerability analysis. Golden images are a useful management tool to support fleet-wide configuration management and change control, which supports enhanced vulnerability management outcomes.

Software/Package Name	Current Version Available	Version Used	System Information
KubOS	1.21	1.20	Information on system libraries and protocols: https://docs.kubos.com/1.20.0/deep-dive/apis/kubos-libs.html
Linux	4.4	4.4	Basis for KubOS, utilised for scalability

Table 2 - Beaglebone Black KubOS Core Software Packages – Example (Not Comprehensive)

It is recommended that system owners acquire or develop a full understanding of the software supply chain, including package dependencies, patching and the process for

notification of vulnerabilities. The creation of a Software Bill of Materials (SBoM) and the subsequent management of the SBoM through a configuration management process and supporting vulnerability management system are foundational capabilities all space system operators should seek to achieve across all their architectures. Ultimately, the objective is to maintain *"accurate and up-to-date data, provenance (i.e. origin) of software code or components, and controls on internal and third-party software components, tools, and services present in software development processes, and performing audits and enforcement of these controls on a recurring basis"* (Whitehouse, 2021). An example of a SBoM provided by National Telecommunications and Information Administration (NTIA) is depicted in Figure 9 (NTIA, 2020). The use of an SBoM and associated scanning tools (such as CycloneDX and Nexus) support the management of larger production systems as they become increasingly complex without appropriate tools.

Best practice not only determines the currently version of software in use, but also identifies if the system is subject to long-term support, how vulnerabilities and patches are managed by the distributor of the software, and when support is likely to end. This information enables effective risk management and cyber-security hardening decisions, when combined with an understanding of how patches and security updates will be applied in the production environment (including for deployed SVs).

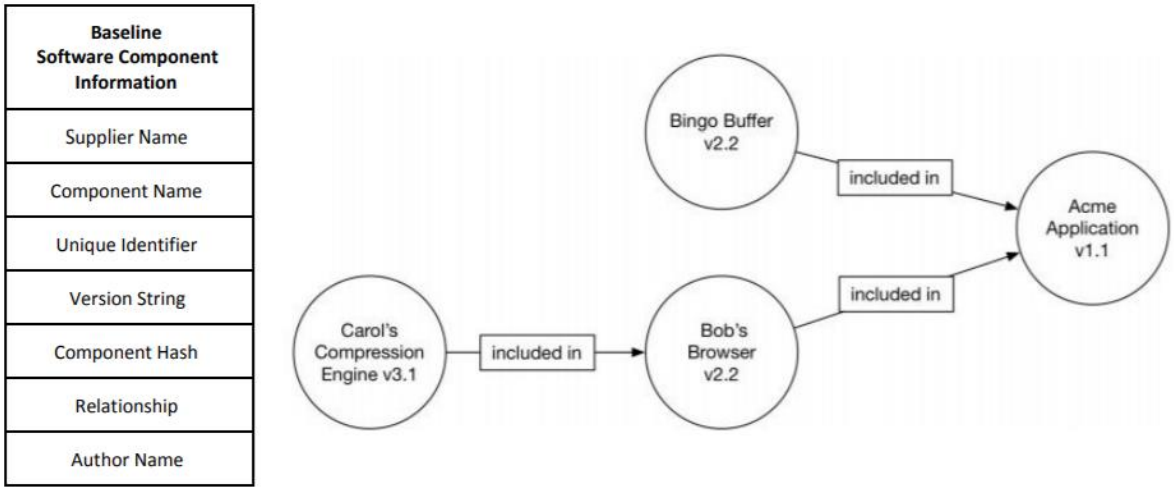

Figure 9 - NTIA SBoM Example

4.2. Jetson Xavier NX

The Jetson Xavier NX is a small form factor system-on-module (SOM) computer manufactured by NVIDIA. The Jetson Xavier NX is commonly employed on critical embedded systems, including robots, instruments, smart cameras, and sensors.

4.2.1. Jetpack SDK Packages and Architecture

JetPack SDK includes the Jetson Linux Driver Package (L4T) with Linux operating system and CUDA-X accelerated libraries and APIs (Nvidia, 2021). Some of the core software packages deployed on the testbed Jetson Xavier NX using the Jetpack SDK, including the deployed software versions and currently available versions, are described in Table 3. This list is provided as an example only.

Software Name	Current Version Available	Version Used	Additional comments
Jetpack SDK	4.6	4.4	JetPack 4.6 is the latest production release which supports all Jetson modules. The testbed employs Jetson 4.4.
L4T	32.6.1	32.4.3	NVIDIA L4T provides the bootloader, Linux kernel, necessary firmwares, NVIDIA drivers, sample filesystem, and more. Based on Ubuntu 18.04.
Linux Kernel	5.13.11	4.9	An open-source Unix-like operating system kernel.
Sample rootfs	Ubuntu 20.04.2.0	Ubuntu 18.04	Ubuntu 18.04 (arm64 distribution)
Vulkan	1.2	1.2	Vulkan is a low-level API that gives direct access of the GPU to developers. The Vulkan driver is a default component of the Linux For Tegra BSP.

Table 3 - Jetson Xavier NX using the Jetpack SDK Core Software Packages (Example – not comprehensive)

4.3. Sony Spresense Camera

Sony SPRESENSE is a low power board computer which additionally supports GPS locating together with high resolution audio and visual codecs. The C-based Spresense SDK is open source and is based on real-time OS NuttX. SPRESENSE has also support for the Arduino IDE. As SPRESENSE is open-source, Sony has published resources to assist as developer guides and references.

4.3.1. Sony Spresense Architecture and Packages

An example of the Sony Spresense software packages is provided in Table 4. This list is provided as an example.

Software Name	Current Version Available	Version Used	Additional comments
Arduino IDE	1.8.15	1.8.13	Using the Spresense Arduino Library allows for software development to be undertaken through Arduino IDE
NuttX	10.0	8.2	The Spresense SDK is Sony's original development environment for the CXD5602 chipset. It based on NuttX and uses GNU Make
CircuitPython	6.3.0	6.3.0	CircuitPython is a programming language with added device libraries and drivers to support microcontroller hardware and sensors. Sony has ported CircuitPython for Spresense.

Table 4 - Sony Spresense Software Packages (Example – not comprehensive)

4.4. Adafruit Feather LoRa RF System

The Adafruit Feather M0 RFM69 Packet Radio is an open-source design, portable microcontroller with a Long Range (LoRa) packet radio transceiver. Beyond the radio transceiver, the primary component of the device is an ATSAMD21G18 ARM Cortex M0 processor.

4.4.1. LoRa Protocol

LoRa is a low-power wide-area network modulation technique using 915-928 MHz in Australia (Seneviratne, 2019). LoRa is designed to *"wirelessly connect battery operated 'things' to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements"* through a specification managed by the LoRa Alliance (LoRa Alliance, 2021).

4.5. UART Communications

UART communications employ RS-232 Serial Communications though an asynchronous duplex receive and transmit connection together with a ground terminal. UART connections are used throughout the testbed to connect the devices together (Dallas Semiconductor, 1983).

5. Build a Digital Twin

Develop a digital twin using simulation systems and/or a replication of the satellite systems, with a methodology to support testing, data collection, data storage and Verification, Validation and Accreditation (VV&A) as appropriate.

LEO SV Constellations provide a stacked array of systems, integrated within the framework of larger systems of systems. The interconnected nature of the constellation, ground station and various supporting systems provides a large attack surface. Table 5 and Figure 10 depict potential generic versions of LEO SV and supporting system architectures, which can be developed into representative digital twin systems for testing purposes. These generic versions are helpful because they do not create a security, safety, or intellectual property risk for any satellite operators. However, they do allow for the testing of principles and approaches to determine appropriate methodologies and models that can subsequently support real systems in the future.

LEO Vehicle		**Ground Station**	**Launch Vehicle**	**Launch Site**
Bus System	**Payload System**			
Command and Data Handling (BCDH)	Payload Processing Module (PPM)	Encryption and Certificate Management (GECM)	Launch Vehicle Software Stack (LCSS)	Launch Control Software Stack (SCSS)
Electrical Power System (BEPS)	Payload Sensor Systems (PSS)	Application Programming Interfaces (GAPIs)	Propulsion System (LPS)	Fuel System (SFS)
Telemetry and Tracking (BTT)	Payload Data Storage (PDS)	Directory Services (GDS)	Avionics and Telemetry (LAT)	Launch Site Management System (SMS)
Communication Subsystem (BCS)	Payload Antenna Array (PAA)	Ground Control Network (GCN)	Launch Vehicle Communications System (LCS)	Encryption & Certificate Management (SECM)
Attitude Determination and Control System (BADCS)	Mission Systems (PMS)	Flight Control Software Stack (GFCSS)	Fuel System (LFS)	Application Programming Interfaces (SAPIs)
Thermal Control (BTC)	Payload System User (PSU)	Cloud Services (GCS)	Electric Pump System (LEPS)	Directory Services (SDS)
Services Control (BSC)	Payload System Admin (PSA)	Human Social Network (GHSN)	Launch Vehicle User (LSU)	Cloud Services (SCS)
Bus System User (BSU)	Payload System Network (PSN)	Ground Station User (GSU)	Launch Vehicle Admin (LSA)	Launch Site User (LSU)
Bus System Admin (BSA)		Ground Station Admin (GSA)	Launch Vehicle Network (LVN)	Launch Site Admin (LSA)
Bus System Network (BSN)		Ground Station Network (GSN)		Launch Site Network (LSN)

Table 5 - Common Generic LEO Space-System Cyber-Security Digital Twin Testbed SubSystems

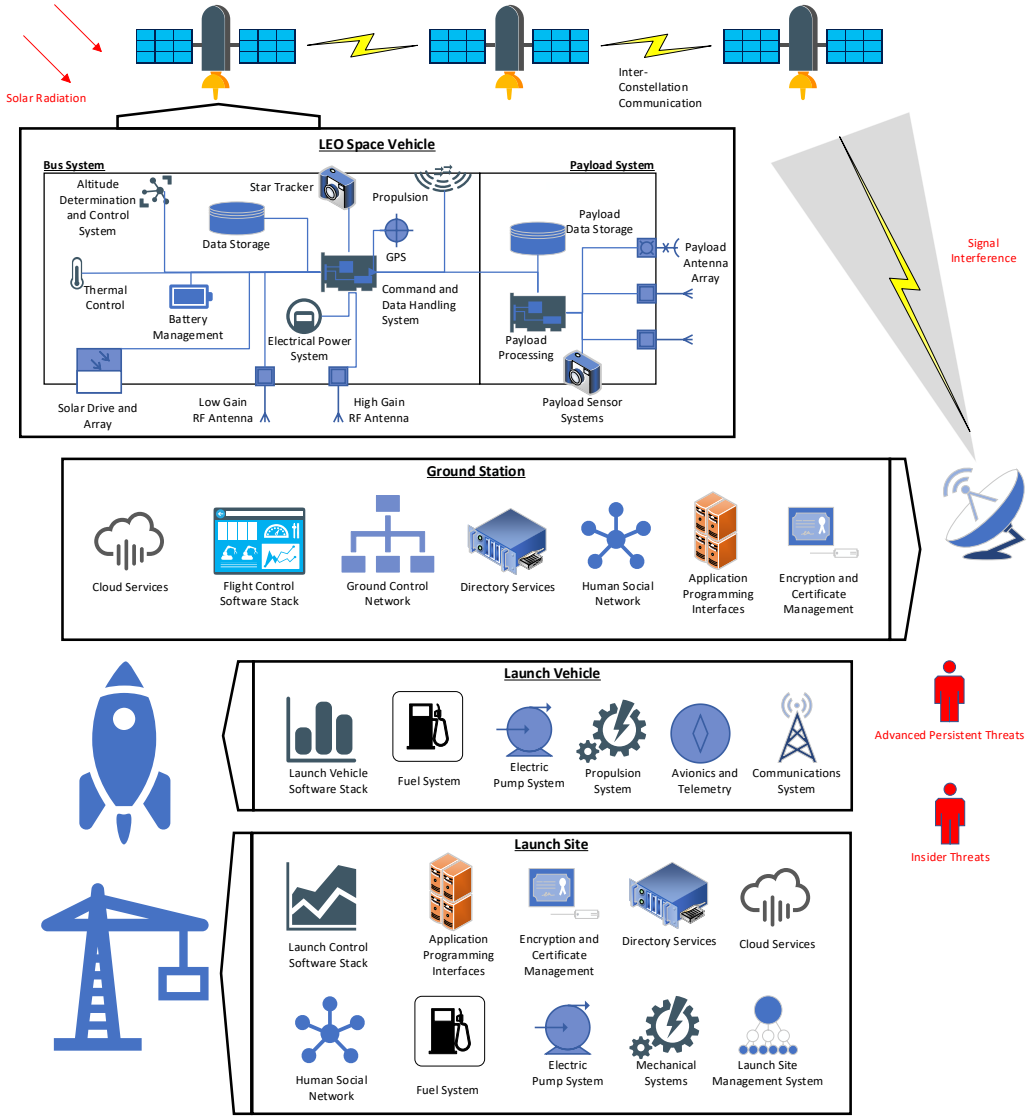

Figure 10 - Common Generic LEO Space-System Cyber-Security Digital Twin Testbed High Level Architecture

For this proof-of-concept report, the scope of the systems under analysis have been reduced to a smaller system, representing a simulated LEO SV. The system is depicted in Figure 11.

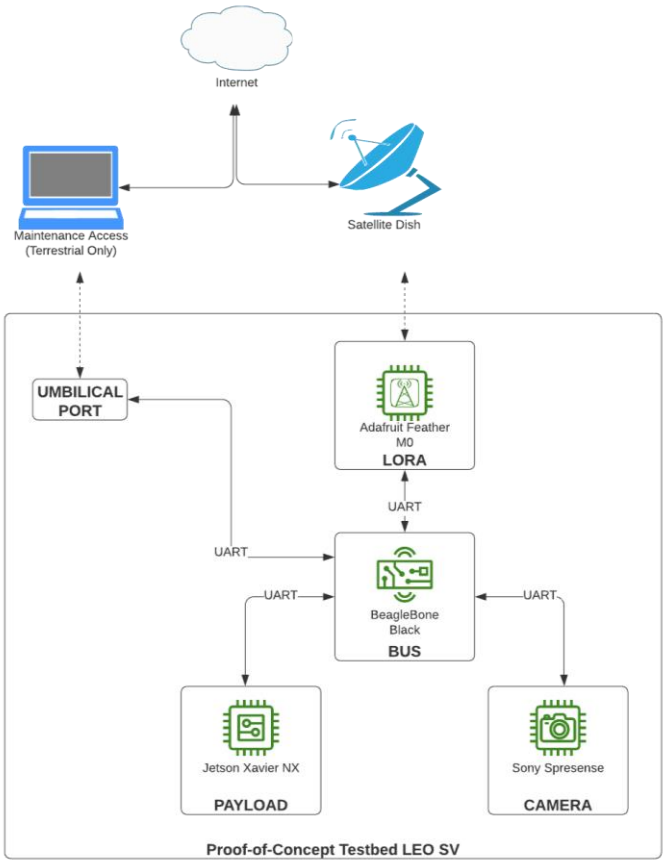

Figure 11 - Proof-of-Concept Testbed Example LEO SV Architecture

5.1. Beaglebone Black

Representing the LEO SV Bus System. Depicted in Figure 12.
Beaglebone Black hardware (https://beagleboard.org/black).
KubOS (https://github.com/kubos/kubos) is the software platform on the Beaglebone.
KubOS Communication (https://docs.kubos.com/1.21.0/ecosystem/services/comms-framework.html)

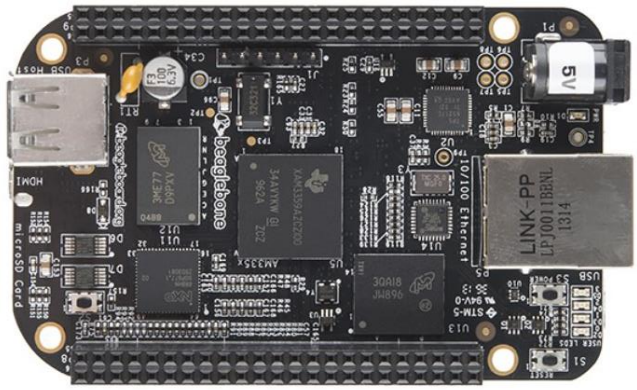

Figure 12 - Beaglebone Black

5.2. Jetson Xavier NX

Representing the LEO SV Payload System. Depicted in Figure 13.
Jetson Xavier NX hardware (https://www.nvidia.com/en-au/autonomous-machines/embedded-systems/jetson-xavier-nx/) combined with a Capable Robot AntMicro Jetson Nano / Xavier NX Baseboard depicted in Figure 14 (https://capablerobot.com/products/nx-baseboard/)
Jetpack 4.4 (https://developer.nvidia.com/embedded/jetpack) software.

Figure 13 - Jetson Xavier NX

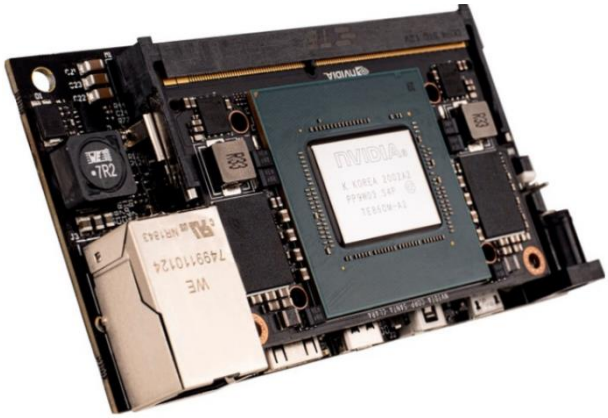

Figure 14 - Jetson Nano / Xavier NX Baseboard

5.3. Sony Spresense Camera

Representing a LEO SV camera. Depicted in Figure 15.
Sony Spresense Camera (https://developer.sony.com/develop/spresense/specifications) (https://www.hackster.io/jpenner64/sony-spresense-camera-basics-fa5476).

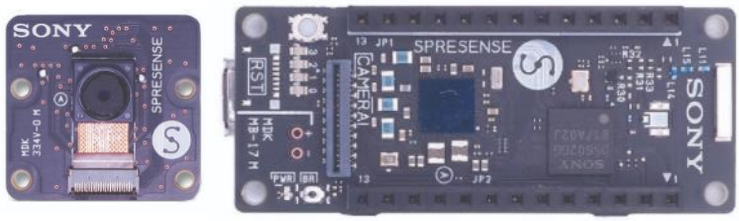

Figure 15 - Sony Spresense Camera and Board

5.4. Adafruit Feather LoRa Radio

Representing a LEO SV Radio. Depicted in Figure 16.
Adafruit Feather M0 RFM69 Packet Radio
(https://www.adafruit.com/product/3178?gclid=CjwKCAjwmeiIBhA6EiwA-uaeFUEPu8JLqxHlPDB5I8pc0KE42AfXnfHfL7SDyWuJHL8S1aWkpmmjKxoCc0sQAvD_BwE)

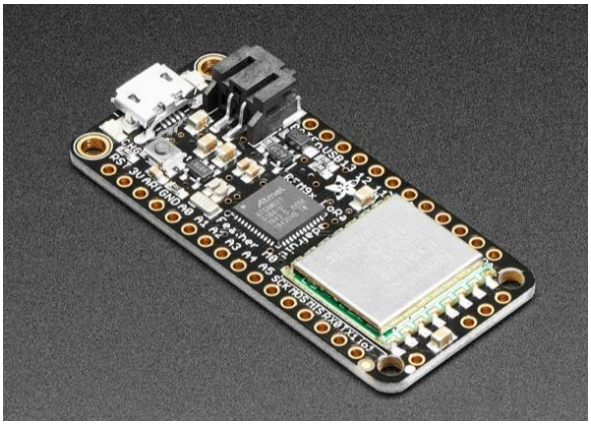

Figure 16 - Adafruit Feather LoRa

5.5. Connections

UART serial connection Beaglebone Black (Bus) to Adafruit Feather LoRa (Radio);
UART serial connection Beaglebone Black (Bus) to Jetson Xavier NX (Payload);
UART serial connection Beaglebone Black (Bus) to Sony Spresense (Camera);
UART serial connection Beaglebone Black (Bus) to Umbilical terrestrial maintenance port.
No USB or Ethernet connections.
Some custom boards were required to facilitate these connections.

6. Develop Vulnerability Models

Vulnerability models extend on the architecture of the target system using the decomposition provided by the SBoM. Vulnerabilities should be assessed across the NIST Cybersecurity Framework Functions of Identify, Protect, Detect, Respond and Recover. Vulnerability models consider hardware, software and protocol weaknesses which could be exploited.

Vulnerabilities can be classified as follows:

1. Known vulnerabilities with patches, which have been applied across all target systems.
2. Known vulnerabilities with patches, which have NOT been applied across all target systems.
3. Known vulnerabilities WITHOUT patches, which have been hardened to reduce risk.
4. Known vulnerabilities WITHOUT patches, which have NOT been hardened.
5. Potential vulnerabilities which have not been reported formally but are theoretically feasible.

In some cases, vulnerability information on different systems that share specific architectural attributes or protocols can be invaluable in understanding potential system vulnerabilities. Vulnerability information can be collected from numerous sources. Using the example proof-of-concept LEO Testbed in this report and the identified APT Turla from the Threat Library, sources include:

6.1. The NIST National Vulnerability Database.

For example - https://nvd.nist.gov/vuln/detail/CVE-2017-6283 and https://nvd.nist.gov/vuln/detail/CVE-2020-27402

"CVE-2017-6283 Detail - NVIDIA Security Engine contains a vulnerability in the RSA function where the keyslot read/write lock permissions are cleared on a chip reset which may lead to information disclosure. This issue is rated as high.
CVE-2020-27402 Detail - The HK1 Box S905X3 TV Box contains a vulnerability that allows a local unprivileged user to escalate to root using the /system/xbin/su binary via a serial port (UART) connection or using adb".

6.2. Exploit Database.

For example - https://www.exploit-db.com/exploits/49789
"Hasura GraphQL 1.3.3 - Denial of Service".

6.3. NVIDIA customer support articles.

For example – https://nvidia.custhelp.com/app/answers/detail/a_id/4635/kw/security
"Security Bulletin: NVIDIA Jetson TX1, Jetson TK1, Jetson TX2, and Tegra K1 L4T Security Updates for Multiple Vulnerabilities. JETSON AND TEGRA L4T CONTAIN VULNERABILITIES WHICH MAY LEAD TO DENIAL OF SERVICE, ESCALATION OF PRIVILEGES, OR INFORMATION DISCLOSURE".
Provides a list of relevant CVEs.

6.4. Research and Academic papers.

For example - https://blog.securityinnovation.com/iot_uart
"IoT Devices - The Not-So-Hidden Risk of UART Interface.
Since UART interface can be used as a debugging interface on the device or to view the serial logs, it is possible for an attacker to gain shell or even root shell access to the device.

Root access over UART is not too uncommon and the same steps can be followed to gain root access on potentially a lot of IoT and embedded devices available in the market today. Once an attacker has access to the root shell they can download/reverse engineer the firmware, retrieve sensitive certificates or API keys stored, identify the communication protocols and potentially target devices of other users or companies".

6.5. LoRaWAN

For example - https://act-on.ioactive.com/acton/attachment/34793/f-87b45f5f-f181-44fc-82a8-8e53c501dc4e/1/-/-/-/-/LoRaWAN%20Networks%20Susceptible%20to%20Hacking.pdf and https://www.haystacktechnologies.com/2020/01/29/where-you-can-go-in-the-aftermath-of-the-lorawan-hack/

"LoRaWAN is fast becoming the most popular wireless, low-power WAN protocol. It is used around the world for smart cities, industrial IoT, smart homes, etc., with millions of devices already connected.
The LoRaWAN protocol is advertised as having "built-in encryption" making it "secure by default." As a result, users are blindly trusting LoRaWAN networks and not paying attention to cyber security; however, implementation issues and weaknesses can make these networks easy to hack.
Currently, cyber security vulnerabilities in LoRaWAN networks are not well known, and there are no existing tools for testing LoRaWAN networks or for detecting cyber attacks, which makes LoRaWAN deployments an easy target for attackers.
In this paper, we describe LoRaWAN network cyber security vulnerabilities and possible cyber attacks, and provide useful techniques for detecting them with the help of our open-source tools".

In the example of the Jetson Xavier NX software Jetpack SDK and L4T only, vulnerabilities were identified based on the versions being used in the testbed, as depicted in Table 6. The bulk of these vulnerabilities had a Common Vulnerability Scoring System (CVSS) score of HIGH.

Software Name	Current Version Available	Version Used	Vulnerability Data
Jetpack SDK	4.6	4.4	https://nvidia.custhelp.com/app/answers/detail/a_id/5039/~/security-bulletin%3A-nvidia-jetson-agx-xavier%2C-tx1%2C-tx2%2C-and-nano-l4t---july-2020 CVE-2020-5974 – CVSS - HIGH https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=jetpack+4.2&search_type=all&isCpeNameSearch=false CVE-2020-5974 – Repeat (above)
L4T	32.6.1	32.4.3	https://nvidia.custhelp.com/app/answers/detail/a_id/5205 CVE-2021-34372 – CVSS - HIGH CVE-2021-34374 – CVSS - HIGH CVE-2021-34375 – CVSS - HIGH CVE-2021-34376 – CVSS - HIGH CVE-2021-34377 – CVSS - HIGH

			CVE-2021-34378 – CVSS - HIGH CVE-2021-34379 – CVSS - HIGH CVE-2021-34380 – CVSS - HIGH CVE-2021-34383 – CVSS - MEDIUM CVE-2021-34384 – CVSS - HIGH CVE-2021-34389 – CVSS - MEDIUM CVE-2021-34393 – CVSS - MEDIUM CVE-2021-34394 – CVSS - MEDIUM CVE-2021-34396 – CVSS - LOW CVE-2021-34397 – CVSS - LOW https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=L4T&search_type=all&isCpeNameSearch=false CVE-2021-1071 – CVSS - MEDIUM CVE-2021-1070 – CVSS - HIGH

Table 6 - Jetpack SDK and L4T Vulnerabilities - Jetson Xavier NX Testbed Vulnerability Modelling

Managing and detecting vulnerabilities is a key component of good cyber-security hygiene, patching practices, governance, and configuration management.

7. Conduct Threat Modelling

Undertake threat modelling using shared toolsets, to ensure consistency and coverage of agreed threats and vulnerabilities. Align vulnerabilities with system assets and architecture. Confirm overlap of Threat TTPs with vulnerable systems.

Threat modelling requires five enablers:
1. Information about threat actors and their TTPs;
2. Information about the target environment;
3. Understanding and modelling of adversary intent;
4. Collect vulnerability and detection information; and
5. A threat modelling toolset.

7.1. Information about threat actors and their TTPs

Threat actor data can be collected from numerous sources. Using the example APT Turla from the Threat Library, examples of threat actor and TTP sources include:

7.1.1. Securelist reports by Kaspersky.

For example - https://securelist.com/the-epic-turla-operation/65545/ and https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/
"The Epic Turla Operation. APT report.
Over the last 10 months, Kaspersky Lab researchers have analysed a massive cyber-espionage operation which we call "Epic Turla". The attackers behind Epic Turla have infected several hundred computers in more than 45 countries, including government institutions, embassies, military, education, research, and pharmaceutical companies. The attacks are known to have used at least two zero-day exploits.
Satellite Turla: APT Command and Control in the Sky
What makes the Turla group special is not just the complexity of its tools, which include the Uroboros rootkit, aka "Snake", as well as mechanisms designed to bypass air gaps through multi-stage proxy networks inside LANs, but the exquisite satellite-based C&C mechanism used in the latter stages of the attack"

7.1.2. MITRE ATT&CK information.

For example - https://attack.mitre.org/groups/G0010/
Provides TTP and software information mapped to the MITRE ATT&CK Framework.

7.1.3. Alienvault posts.

For example - https://otx.alienvault.com/pulse/55f08e374637f26df8744429/history
Satellite Turla: APT Command and Control in the Sky (refers to Securelist report above).

7.1.4. ThaiCERT website.

For example - https://apt.thaicert.or.th/cgi-bin/showcard.cgi?g=Turla%2C%20Waterbug%2C%20Venomous%20Bear
"Threat Group Cards: A Threat Actor Encyclopedia.
APT group: Turla, Waterbug, Venomous Bear".
Provides a comprehensive summary of historical activities.

7.1.5. Unit 42.

For example - https://unit42.paloaltonetworks.com/ironnetinjector/

"IronNetInjector: Turla's New Malware Loading Tool
Unit 42 researchers have found several malicious IronPython scripts whose purpose is to load and run Turla's malware tools on a victim's system. The use of IronPython for malicious purposes isn't new, but the way Turla uses it is new. The overall method is known as Bring Your Own Interpreter (BYOI). It describes the use of an interpreter, not present on a system by default, to run malicious code of an interpreted programming or scripting language"

7.1.6. Recorded Future.

For example - https://www.recordedfuture.com/turla-apt-infrastructure/

"Swallowing the Snake's Tail: Tracking Turla Infrastructure.
Recorded Future's Insikt Group® has developed new detection methods for Turla malware and infrastructure as part of an in-depth investigation into recent Turla activities. Data sources included the Recorded Future® Platform, ReversingLabs, VirusTotal, Shodan, BinaryEdge, and various OSINT tools. The target audience for this research includes security practitioners, network defenders, and threat intelligence professionals who are interested in Russian nation-state computer network operations activity".

7.1.7. Crowdstrike.

For example - https://www.crowdstrike.com/blog/meet-crowdstrikes-adversary-of-the-month-for-march-venomous-bear/

"Meet CrowdStrike's Adversary of the Month for March: VENOMOUS BEAR.
VENOMOUS BEAR is an advanced, Russia-based adversary that's been active since at least 2004. Some of its aliases include Turla, Snake, and Krypton. Recent public reporting has surfaced indicating that this threat actor is suspected of breaching a Western government's foreign ministry, and there have been new innovations by this threat group in its tools and capabilities".

7.1.8. Research and Academic papers.

For example - https://www.welivesecurity.com/wp-content/uploads/2020/05/ESET_Turla_ComRAT.pdf

"FROM AGENT.BTZ TO COMRAT V4. A ten-year journey. By Matthieu Faou.
Turla, also known as Snake, is one of oldest cyberespionage groups still active, with more than a decade of experience. Its operators mainly focus on high-profile targets such as governments and diplomatic entities in Europe, Central Asia, and the Middle East. They are known for having breached major organizations such as the US Department of Defense in 2008 and the Swiss Defense company RUAG in 2014. More recently, several European countries including France and the Czech Republic went public to denounce Turla's attacks against their governments. To perform these operations, Turla's operators maintain a large arsenal of malware including a rootkit, several complex backdoors aimed at different platforms, including Microsoft Exchange mail servers, and a large range of tools to enable pivoting on a network. In this white paper, we present our analysis of the latest version of one their oldest backdoors, publicly known as ComRAT"

7.2. Information about the target environment

The understanding of architectural and system information documented earlier in this process will inform an understanding of the target environment. Often this requires

considerable effort due to missing documentation and limited information relating to supply chain and package dependency information. In addition, different versions and variants of software and systems may be in use in parallel across an organisation.

7.3. Understanding and modelling of adversary intent

Threat actor intent is documented as a desired effect derived from the 5Ds (deceive, degrade, deny, disrupt, destroy). These effects are achieved through objectives described as threat events. Threat events can be achieved by performing a set of TTPs. The TTPs which can achieve a threat event can be proceeded by follow up TTPs, which do not form part of the threat event itself, but link to further actions the adversary may undertake once the threat event is complete. This allows for chaining of threat events to be conducted if desired.

An example of adversary intent is documented below:
Desired Threat Effect: Deceive.
Threat Event: Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.
TTPs Required to Achieve: Reconnaissance; Resource Development; Initial Access.

Threat Effects should cascade down into multiple Threat Events. Threat Events should be supported with TTPs required to achieve and misuse cases (UcedaVelez & Morana, 2015), as depicted in Figure 17.

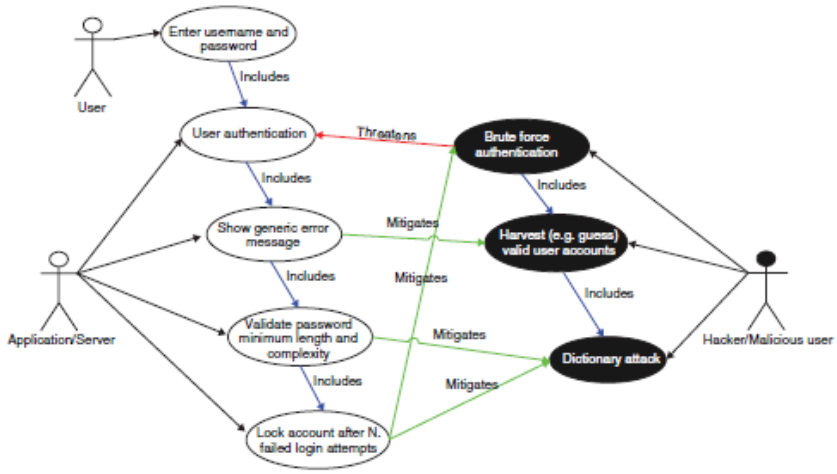

Figure 17 - Use and Misuse Case of User Logon (UcedaVelez & Morana, 2015)

7.4. Collect vulnerability and detection information

Vulnerability models relating to each system should be collected and matched against threat events. New threat events should be considered where vulnerabilities may enable initial access or support specific TTPs (such as escalation of privilege or lateral movement).

Detection information should be collected to map against the pyramid of pain, as depicted in Figure 18.

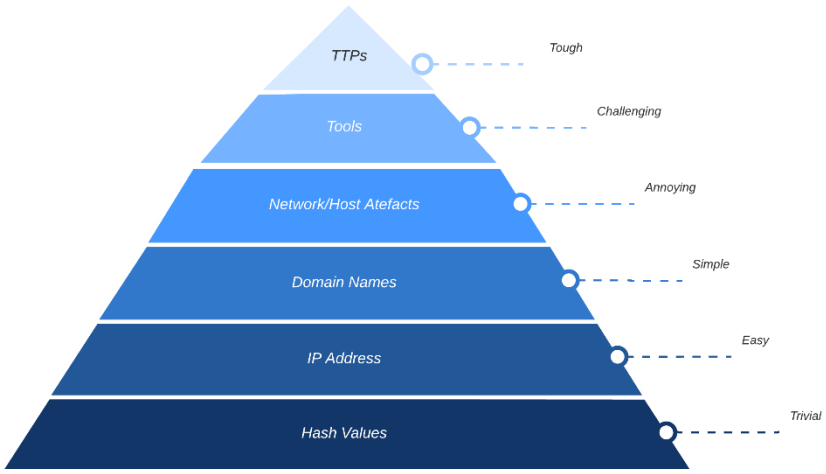

Figure 18 - The Pyramid of Pain

For example, the following information can be obtained as it relates to the Turla group:

7.4.1. TTPs.

T1134.002. Access Token Manipulation: Create Process with Token. Turla RPC backdoors can impersonate or steal process tokens before executing commands.
https://attack.mitre.org/groups/G0010/

7.4.2. Tools.

S0099. Arp. System Network Configuration Discovery.
https://attack.mitre.org/groups/G0010/
a3cbf6179d437909eb532b7319b3dafe - custom keylogger
https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08080105/KL_Epic_Turla_Technical_Appendix_20140806.pdf

7.4.3. Network/Host Artefacts.

Backdoor.Win32.Turla.cd
Backdoor.Win32.Turla.ce
https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/

7.4.4. Domain Names.

hxxp://losdivulgadores[.]com/wp-content/plugins/wp-themes/
hxxp://gspersia[.]com/first/fa/components/com_sitemap/
https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08080105/KL_Epic_Turla_Technical_Appendix_20140806.pdf
accessdest.strangled[.]net
bookstore.strangled[.]net
https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/

7.4.5. IP Address

84.11.79.6
41.190.233.29
https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/

7.4.6. Hash Values.

Fake Flash Player (MD5: 030f5fdb78bfc1ce7b459d3cc2cf1877)
https://securelist.com/the-epic-turla-operation/65545/
ComRAT variant (SHA256: 3aa37559ef282ee3ee67c4a61ce4786e38d5bbe19bdcbeae0ef504d79be752b6)
ComRAT DLL payload (SHA256: a62e1a866bc248398b6abe48fdb44f482f91d19ccd52d9447cda9bc074617d56)
https://unit42.paloaltonetworks.com/ironnetinjector/

7.5. A threat modelling toolset

Using the information collected throughout the earlier process, a threat modelling tool should be selected. For this report, a commercial tool called YAKINDU Security Analyst by Itemis has been used. The tool can be found here: https://www.itemis.com/en/yakindu/security-analyst/
Threat modelling tools allow for the use of libraries and common threats and risks to be explored with significantly less overhead than manual assessments. An example of the type of libraries is provided in Figure 19. These libraries reduce rework and provide a repository of commonly utilised threat information, aligned to industry standards.

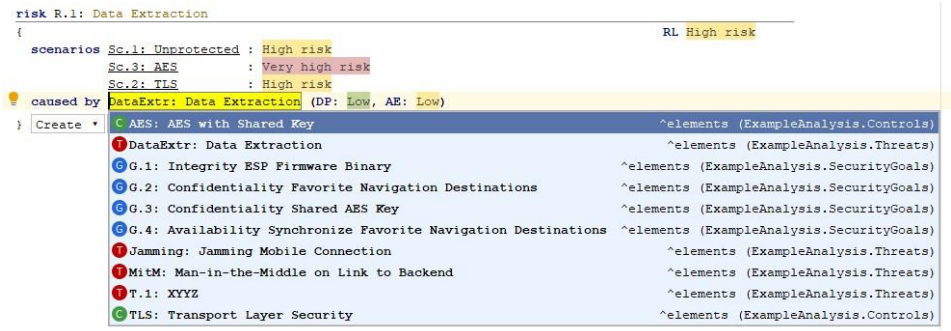

Figure 19 - Libraries available in Threat Modelling Software Reduce Rework

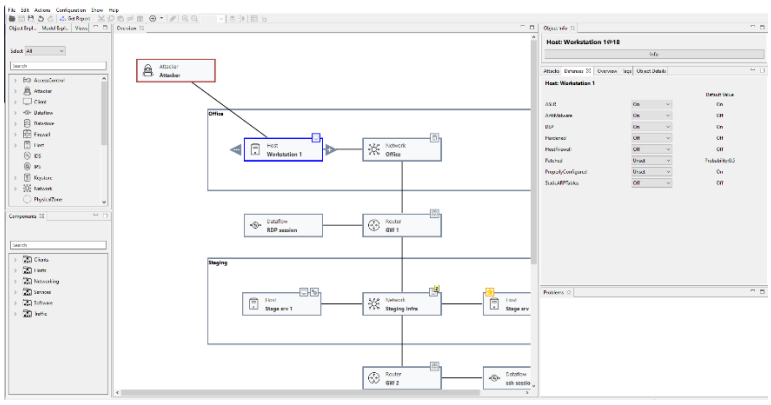
Figure 20 - Threat Modelling Software Allows for Integrated Visualisation of Architecture and Threat Actor Interactions

Architecture can be rapidly visualised and threat actor interactions can be mapped as depicted in Figure 20.

Reports and risk assessment outcomes can be exported and provided to governance committees and decision makers, as depicted in Figure 21.

Figure 21 - Threat Modelling Software Allows for Rapid and Integrated Reporting

The use of a threat modelling tool is recommended to support ongoing maintenance, alignment to standards and effective cyber-security risk management.

8. Record a Baseline

Using the digital twin, understand what is considered 'normal' (or expected) behaviour and build models of the system under various operational conditions where a cyber-attack is not occurring. This will support future testing as well as the detection of unusual behaviour.

The testbed has been used to collect data and understand normal traffic and behaviours on the system to inform future development and testing.

9. Develop an Initial Mission Resilience Sub-System Crosswalk

Conduct a cross-walk of each sub-system against the MITRE Mission Resilience Engineering framework, to determine both defence-in-depth and defence-in-breadth coverage at a sub-system level.

The critical systems enabling each mission will be mapped in the chart below as part of the CY-JAR project. This is future work.

SubSystem	Anticipate			Withstand		Recover	Evolve	
	Understand	Prepare	Prevent	Continue	Constrain	Reconstitute	Transform	Rearchitect
Bus								
Radio								
Payload								

Table 7 - Mission Cyber-Resilience SubSystem CrossWalk

10. Develop a Crown Jewels and Mission Assessment

Undertake a crown jewels assessment and map mission-essential functions and systems to support prioritisation.

Crown Jewels Assessment:

- The BeagleBone Black and KubOS operating system are crown jewels, as they provide the satellite Bus. Without this system the SV will not be operational.
- The Adafruit Feather LoRa Radio is crown jewels, as it provides communication. Without this system the SV cannot communicate.
- The Jetson Xavier NX provides the payload for the satellite system and is critical to the SV service clients. Whilst the Payload is crown jewels, it is secondary to the Bus and Radio systems.

Mission Assessment:

- Retain control of the SV: Critical systems are the Bus and Radio.
- Communicate with the SV: Critical systems are the Radio.
- Provide client services: Critical systems are Radio and Payload.

Mission Essential Systems (in priority order):

- Bus: BeagleBone Black and KubOS;
- Radio: Adafruit Feather LoRa; and
- Payload: Jetson Xavier NX.

Security Measures of Effectiveness (examples):

- Zero onboard SV system communication anomalies occur with unknown causation;
- 100% of applications and software running on the SV have been whitelisted and authorised;
- 100% of applications and software running on the SV are contained in a SBoM and subject to regular configuration management;
- 100% of critical security software patches are applied within 48 hours of release;
- 100% of applications are tested in a digital twin prior to patch; and
- A complete security review is conducted before all configuration changes on all SV systems and software.

Extensive analysis and mapping are recommended for more complex systems. However, a basic review as provided above supports prioritisation and analysis efforts. A short assessment can allow for initial system analysis. Outcomes should be refined and improved over time.

11. Conduct Impact Analysis

Determine impact of specific adversarial targets if they are achieved. Map prior threat modelling and crown jewels assessment results to likely adversary targets and threat surface. Utilise mapping to review high-value vulnerabilities, entry, and egress points into and out of major systems, lateral movement paths, adversary countermeasures to security controls, and likely points for privilege escalation to support TTPs.

There are a number of sophisticated approaches to conduct impact analysis. However, an effective method is to generate a visualisation as depicted in Figure 22. A visualisation can be used to test for second order mission impacts by wargaming attacks on dependency subsystems and external access points, examining threat actor TTPs and then tracing the impact on Confidentiality, Integrity and Availability from bottom to top, and laterally across, each of the connected dependency subsystems, systems, functions and missions. As systems become larger and more complex than the example provided, these diagrams and visualisations can also support incident response and recovery actions.

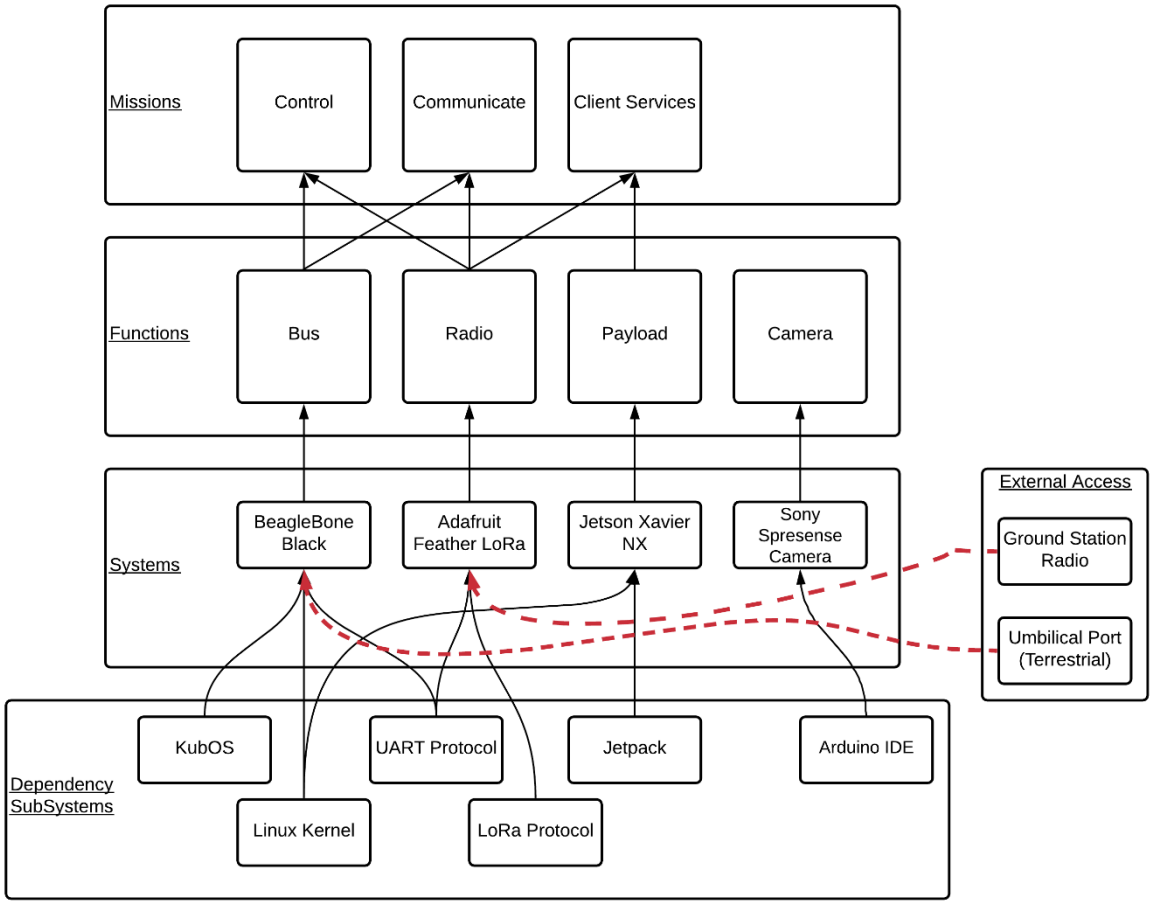

Figure 22 - Visualisation of Impact Analysis - Testbed Example

12. Conduct security testing using the Digital Twin

Undertake hands-on penetration testing and experimentation with the digital twin to test assumptions and confirm TTPs.

This activity will be conducted in preparation of deliverable three as an example and extended as part of the CY-JAR project.

13. Conduct Countermeasure Research and Analysis

Develop additional security controls and mitigations, including resilience and recovery measures, as required to enhance the overall security of the LEO space system. Maintain a focus on mission assurance capabilities and hardening of systems. Ensure an understanding of impact on system functionality is considered; security can reduce usability.

This process is being undertaken as part of the CY-JAR project.

14. Conduct Desktop Quantitative Resilience Assessment

Undertake a desktop quantitative resilience assessment to confirm the desired changes to the system value add and contribute to the overall resilience of the LEO space system.

This report hereby presents an entirely unique method for conducting a quantitative resilience assessment, utilising pre-existing techniques combined in a probabilistic process. This example is provided to support space system owners to develop resilience and risk assessment approaches, grounded in literature, which are suited to their purpose and provide actionable outcomes to support decision-making. In the example used for this report to support the desktop quantitative resilience assessment, two threat events are considered. Both involve the compromise of passwords through their reuse in another system which has been leaked. However, in one scenario the target user has privileged access (admin rights) and in the other they do not (user rights).

An example of the type of breach possible is depicted in Figure 23, using the tool NexusXplore (https://www.osintcombine.com/nexusxplore) which is an Australian open-source intelligence tool. In this example, the email XXXX@gmail.com has been leaked in four breach databases. Two usernames are associated with these breaches – XXxX and XxXX. Four hashed passwords are provided. These hashed passwords can be reverse engineered through a variety of methods, depending on the complexity of the encryption algorithm applied by the system and if other security measures, such as salting, have been utilised. Tools such as hashcrack are capable of brute force cracking an MD5 hashed eight-character password in just 4.2 hours using an 8 GPU machine. Using AWS, the same can be achieved for approximately $150 (Kenny, 2020).

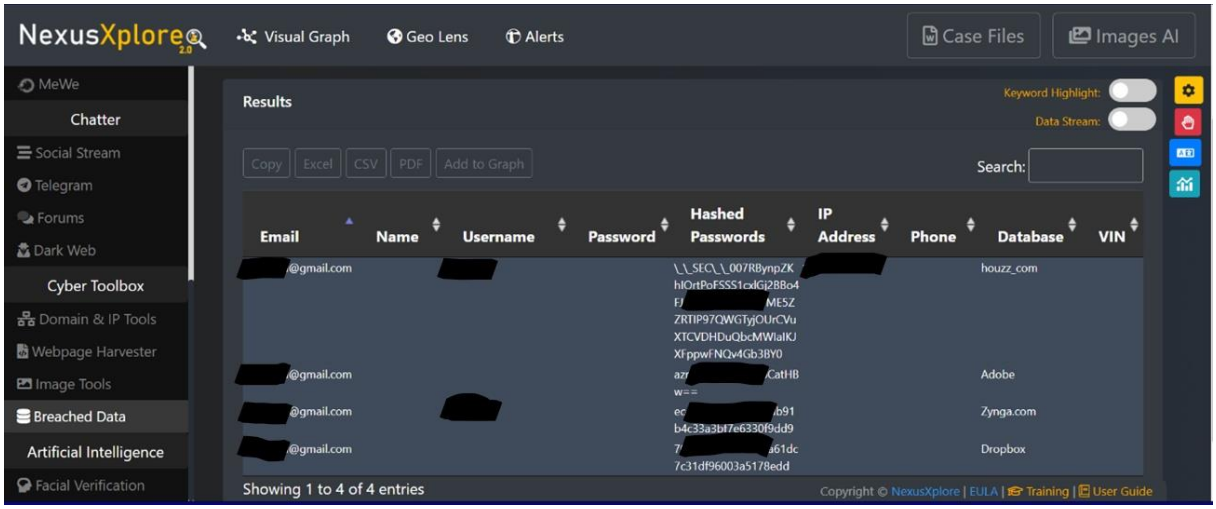

Figure 23 - Example of Breached Data - NexusXplore Output

Password and user data can be sourced by threat actors from sites such as Raid Forums, depicted in Figure 24.

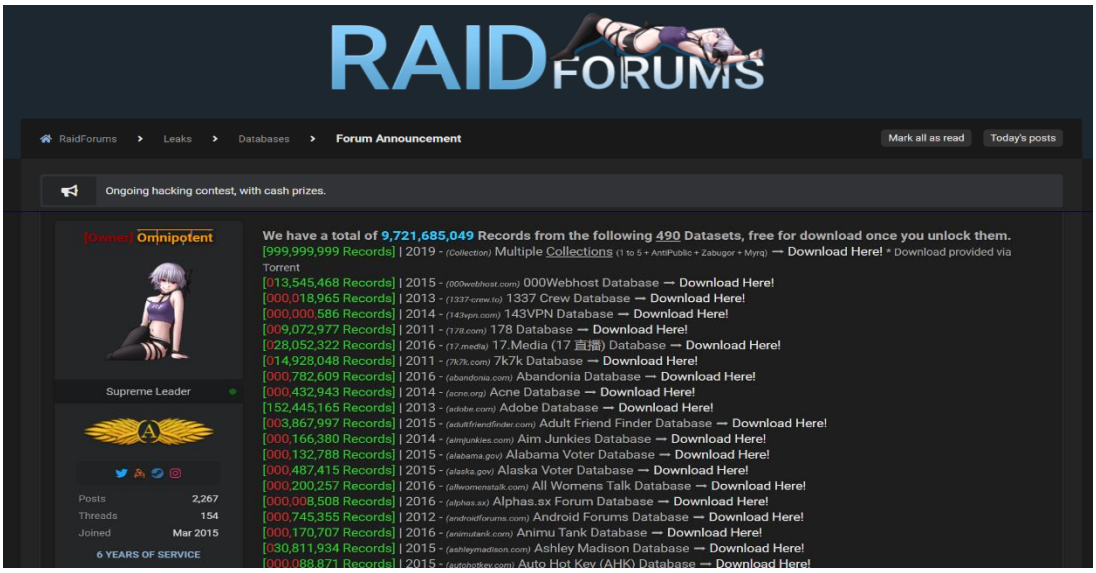

Figure 24 - Raid Forums Breach Databases

The tables below depict the outcome of this simple example, when applied to a quantitative resilience assessment. The following series of tables describe the process and underlying methodology to develop a resilience score and enable effective decision making as to the optimum utilisation of limited resources to enhance the security posture of a system, network, or platform.

Part one of the resilience assessment considers the adversary intent, applied to a specific architecture, and focused on a specific threat actor, using threat event descriptions. New tables are needed for each new threat actor or architecture considered. For each of the two threat event examples used in this report, the threat actor has an intent, reflected in the desired effect derived from the 5Ds (deceive, degrade, deny, disrupt, destroy). The threat event is a description of an objective which will support the achievement of the desired threat effect. The threat event can be achieved by performing a set of TTPs, described as TTPs achieved. Follow up TTPs do not form part of the threat event itself, but link to further actions the adversary may undertake once the threat event in question is complete. This allows for chaining of threat events to be conducted if desired. The next set of five columns provides an assessment of the adversary perception of the system; their perception of what capability they need, their ability to persist and/or elevate privileges, the likelihood of detection, likelihood of attribution and target attractiveness. In each case, this view may not be accurate; however, it reflects the intelligence assessment of the mental state of the specific threat actor being considered. These values provide an adversary score. The higher the score, the more likely that the threat event will be desired by the threat actor and acted upon.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Perceived Capability Required	Perceived Ability to Persist/ Elevate Privileges	Perceived Likelihood of Detection	Perceived Likelihood of Attribution	Perceived Target Attractiveness	Adversary Score
Architecture Design 001 - Threat 00A											
Bus System	**Bus System User (BSU)**	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.4	0.3	0.1	0.1	0.5	**0.3**
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	0.5	0.9	0.4	0.1	1	**0.9**

Table 8 – Assessment of Adversary Intent – Part One of the Resilience Assessment

Part two of the resilience assessment calculates the impact of the event if it occurred as described. Impact is calculated across the Confidentiality, Integrity and Availability (CIA) triad. A higher score relates to a greater impact, in the event the threat event occurs (noting the impact will change depending on the TTPs achieved within the specific event). This analysis is followed by an assessment of the impact on critical layers, using a variation to the TCP/IP stack. This reflects the fact that different layers of systems can be impacted to different extents, depending on the threat event that occurs. A total impact score is calculated as an average of both the CIA impacts and critical layer impacts.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Impact Type			Impact on Critical Layers					Impact Score
Architecture Design 001 - Threat 00A						Confidentiality	Availability	Integrity	Applications	Transport	Network	Links	Data	
Bus System	**Bus System User (BSU)**	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.5	0.1	0.5	0.2	0	0	0	0.5	0.323333333
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	1	1	1	0.7	0.4	0.2	0.1	0.8	0.94

Table 9 – Assessment of Event Impact – Part Two of the Resilience Assessment

Parts one and two of the resilience assessment process, to calculate adversary intent and impact, were developed using the NIST SP800-30 Adversarial Risk Calculation Template (National Institute of Standards and Technology, 2012) as inspiration, depicted in Table 10.

Threat Event	Threat Sources	Threat Source Chacteristics			Relevance	Likelihood of Attack Initiation	Vulnerabilities and Predisposing Conditions	Severity and Pervasiveness	Likelihood Initiated Attack Succeeds	Overall Likelihood	Level of Impact	Risk
		Capability	**Intent**	**Targeting**								

Table 10 - SP800-30 Adversarial Risk Calculation Template (National Institute of Standards and Technology, 2012)

Part three of the resilience assessment determines the probability of avoidance of each threat event. The assessment examines two states: one pre-control and one post-control. The two states are necessary because changes in system configuration which may incur costs, reduce efficiency, and cause unforeseen impacts on other security controls in a network. By comparing two configurations, the resilience assessment is comprehensive, and the impact descriptions provide a relative rather than an absolute value. Pre and post control states are scored against their probability to deter, pre-empt, effect, detect, counter, and subsequently avoid the threat event. This is calculated using the methodology and formula described by Burch (2019) and depicted below in Table 11 and Equation 1.

Variable	Avoidance Component	Value
P_{DT}	Probability of deterrence =	0.50
P_P	Probability of preemption =	0.10
P_E	Probability of effectiveness =	0.95
P_D	Probability of detection =	0.75
P_C	Probability of counter =	0.90
R_{AV}	**Probability of Avoidance =**	**0.86**

Table 11 - Avoidance Variables (Burch, 2019)

$$R_{AV} = P_{DT} + (1 - P_{DT})P_P + (1 - P_{DT})(1 - P_P)(1 - P_E) + (1 - P_{DT})(1 - P_P)P_E P_D P_C$$

Equation 1 - Avoidance Equation (Burch, 2019)

The pre and post control states are assigned probabilities. Between these two states, the controls applicable to achieving a change between states are described. These controls are categorised as understand, prepare and prevent; applying the cyber resilience objectives described by

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Probability of Avoidance																	
						Pre-Control						Anticipate - Controls			Cost Impacts	Design Impacts	Post-Control						
Architecture Design 001 - Threat 00A						Deter	Preempt	Effect	Detect	Counter	Avoid	Understand	Prepare	Prevent			Deter	Preempt	Effect	Detect	Counter	Avoid	
Bus System	Bus System User (BSU)	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.3	0.2	0.6	0.3	0.4	0.69928	Regular scans of Darknet and password sharing sites/pastebins; Monitoring of privileged user accounts	Policy to prevent password reuse and use of commercial email for non-work related sites; Use of passphrases and password resets when breaches are identified; Principle of least privilege	Monitoring of multiple password login attempts from VPNs and external sites; Monitoring of IPs used for access; 24/7 monitoring of gateway traffic and user behaviours	$300k per annum for SOC monitoring; $100k per annum for monitoring	Data feeds and collection procedures and tools to support SOC monitoring	0.6	0.7	0.6	0.8	0.8	0.98944
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	0.3	0.2	1	0.3	0.4	0.4988	Monitoring of privileged user accounts logins and behaviours; monitoring of attempts to reset administration passwords	Utilisation of jump servers; restricted use of privileged access; Principle of least privilege	Restricted number of privileged access users; seperation of duties; mandatory logging and additional security for logging; Multi-factor authentication	$250 per privileged user (multi-factor)	Reduced freedom for privileged users; multi-factor authentication application; additional logging security; jump server; no internet access from privileged accounts	0.6	0.7	1	0.6	0.7	0.9472

Table 12 - Probability of Avoidance Table - Part Three of the Resilience Assessment

MITRE in the cyber resiliency framework of the MITRE Systems Engineering Guide (MITRE Corporation, 2014). Cost and design impacts are also documented at this point, as they need to be balanced with the intended outcomes of the change in state.
Part four of the resilience assessment determines the robustness metric of the target architecture in pre and post control states. Robustness is calculated as the relative capability of the two states after avoidance has failed, leading to a capability loss. The subsequent percentage of capability retained by the system is a function of the robustness of the system (Burch, 2019). The controls utilised to withstand (continue and constrain) are described by the assessor in accordance with the MITRE cyber resiliency framework (MITRE, 2014). Cost and design impacts are once again captured to support a balanced assessment. The higher the score, the greater the robustness metric of the system.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Robustness Metric									
						Pre-Control			Withstand - Controls		Cost Impacts	Design Impacts	Post-Control		
Architecture Design 001 - Threat 00A						Pre-Event Capability	Post-Event Capability	Robustness	Continue	Constrain			Pre-Event Capability	Post-Event Capability	Robustness
Bus System	Bus System User (BSU)	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.69928	0.3	0.429012699	SOC enabled threat hunt capability; Seperation of duties to support continued operations in event of breach; Password and certificate handling to prevent user level access	Privileged access controls designed to reduce risk of escalation; hardened operating systems; whitelisting of applications	Whitelisting tools - $50k	Whitelisting of applications and associated toolsets	0.98944	0.85	0.859071798
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	0.4988	0.1	0.200481155	SOC enabled threat hunt capability; Jump server controls; logging to support post-breach hunt activities; Privileged access not equated to superuser/root administration level	No internet access from privileged accounts	Nil	Role based access controls established based on principle of least privilege	0.9472	0.4	0.422297297

Table 13 - Robustness Metric Table - Part Four of the Resilience Assessment

Part five of the resilience assessment determines the recovery metric of the target architecture in pre and post control states. This metric is calculated using the post-event capability, which was also used in the robustness metric, but now considers that value relative to the minimum capability required by the system. A time is assigned to understand the relative duration required to enable recovery to the minimum capability. The equation provided by Burch (2019) as depicted in Equation 2 is utilised to calculate the metric.

$$R_{RV} = f(C,v)(t) = \int_{t=0}^{T} C(t)v(t)\,dt$$

Recovery metric — Capability recovery vs. time — Time-based value function

Equation 2 - Recovery Metric Equation

Controls are described using the MITRE recover function of the cyber resiliency framework (MITRE, 2014). Cost and design impacts are captured. The higher the recovery score, the greater its ability to rapidly recover to a minimum level of capability.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Recovery Metric										
						Pre-Control				Recover - Controls	Cost Impacts	Design Impacts	Post-Control			
Architecture Design 001 - Threat 00A						Post-Event Capability	Minimum Capability	Time to Recover	Recovery				Post-Event Capability	Minimum Capability	Time to Recover	Recovery
Bus System	**Bus System User (BSU)**	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.3	0.9	6	0.1	SOC monitoring and user account auditing allows for rapid resets; Follow up threat hunt with well maintained logs	Nil	Logging considerations and access to remote logs	0.85	0.9	3	0.6
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	0.1	1	10	0.111111111	Superuser account for root being segregated and not used except in emergency; Follow up threat hunt with well maintained logs	Nil	Logging considerations and access to remote logs	0.4	1	5	0.083333333

Table 14 - Recovery Metric Table - Part Five of the Resilience Assessment

Part six of the resilience assessment determines the reconstitution metric of the target architecture in pre and post control states. In a similar process used to understand the recovery metric to the minimum level of capability, reconstitution seeks to calculate the replenishment of the full capability of the system post-event, such that the original capability has been restored. A time value is also utilised to understand how quickly reconstitution can be achieved as a relative value, in accordance with Burch (2019) and depicted in Equation 3.

$$R_{RC} = f(C,v)(t) = \int_{t=0}^{T} C(t)v(t)\,dt$$

Reconstitution metric — Capability recovery vs. time — Time-based value function

Equation 3 - Reconstitution Metric Equation

Controls are described using the MITRE recover evolve of the cyber resiliency framework, incorporating the ability to transform and rearchitect the system (MITRE, 2014). Cost and design impacts are captured. The higher the reconstitution score, the greater the system's ability to return to full capability in as short a time as possible.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Reconstitution Metric									
						Pre-Control			Evolve		Cost Impacts	Design Impacts	Post-Control		
									Transform	Rearchitect					
Architecture Design 001 - Threat 00A						Minimum Capability	Full Capability	Reconstitution incl time					Minimum Capability	Full Capability	Reconstitution incl time
Bus System	Bus System User (BSU)	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	0.9	1	0.1	Full user account audit and behavioural monitoring	Persistent threat hunt	$100k for full audit; $200k per annum for persistent threat hunt; $200k per annum for SOC behavioural monitoring	Requires network taps and remote monitoring capability; behavioural monitoring tools and personnel in SOC	0.9	1	0.1
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	1	1	0	Full user account audit and behavioural monitoring	Persistent threat hunt	$100k for full audit; $200k per annum for persistent threat hunt		1	1	0

Table 15 - Reconstitution Metric Table - Part Six of the Resilience Assessment

Part seven of the resilience assessment finalises the process and provides a summary calculation. The complete resilience assessment is conducted through an equation provided by Burch (2019) and depicted in Figure 25. This equation has been supplemented by the assessment of adversary intent and system impact. However, the main principles and underlying mathematics remain unaltered.

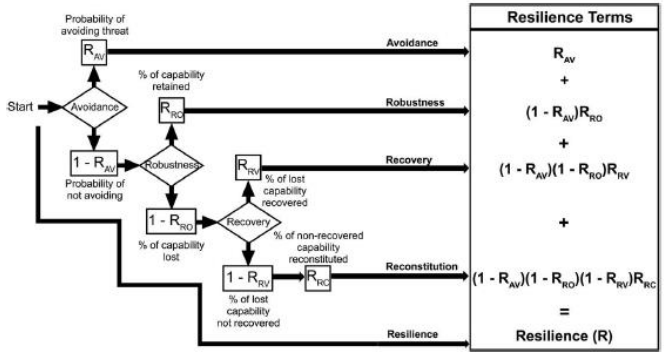

Figure 25 - Resilience Assessment Calculation Flow Chart

This calculation is applied throughout the tables described in this report, providing the overall result in **Error! Reference source not found.**.

System		Desired Threat Effect	Threat Event	TTPs Achieved	Follow Up TTPs	Adversary Score	Impact Score	Resilience Score Without Controls	Resilience Score With All Controls
Architecture Design 001 - Threat 00A									
Bus System	**Bus System User (BSU)**	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Reconnaisance; Resource Development; Initial Access	Execution; Persistence; Privilege Escalation	**0.3**	0.323333333	**0.861**	**0.999**
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Reconnaisance; Resource Development; Initial Access	Defence Evasion; Discovery; Lateral Movement	**0.9**	0.94	**0.644**	**0.972**

Table 16 - Resilience Assessment Overall Result

These results indicate that the resilience of the system is lowest if controls are not applied to protect the Bus System Admin privileged user function. This function is also very attractive to an adversary, with a high score; and has a very high potential impact. Conversely, the Bus System User function is extremely resilient when all controls are applied and has a lower level of attractiveness relative to Admin access, together with a lower impact score. This type of quantitative analysis supports effective risk management and allows the overall cost of resources and design changes to be considered in a logical and balanced manner. It remains a subjective process, but with the appropriate management and use of tools to ensure assessments are appropriately calibrated, this process can provide consistent and accurate results as recommended in Hubbard and Seiersen (2016).

The summary of controls recommended aligned to the MITRE Cyber Resiliency Framework is depicted in Table 17.

System		Desired Threat Effect	Threat Event	Probability of Avoidance			Robustness Metric		Recovery Metric	Reconstitution Metric	
				Anticipate - Controls			Withstand - Controls		Recover - Controls	Evolve	
Architecture Design 001 - Threat 00A				Understand	Prepare	Prevent	Continue	Constrain		Transform	Rearchitect
Bus System	Bus System User (BSU)	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	Regular scans of Darknet and password sharing sites/pastebins; Monitoring of privileged user accounts	Policy to prevent password reuse and use of commercial email for non-work related sites; Use of passphrases and password resets when breaches are identified; Principle of least privilege	Monitoring of multiple password login attempts from VPNs and external sites; Monitoring of IPs used for access; 24/7 monitoring of gateway traffic and user behaviours	SOC enabled threat hunt capability; Seperation of duties to support continued operations in event of breach; Password and certificate handling to prevent user level access	Privileged access controls designed to reduce risk of escalation; hardened operating systems; whitelisting of applications	SOC monitoring and user account auditing allows for rapid resets; Follow up threat hunt with well maintained logs	Full user account audit and behavioural monitoring	Persistent threat hunt
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	Monitoring of privileged user accounts logins and behaviours; monitoring of attempts to reset administration passwords	Utilisation of jump servers; restricted use of privileged access; Principle of least privilege	Restricted number of privileged access users; seperation of duties; mandatory logging and additional security for logging; Multi-factor authentication	SOC enabled threat hunt capability; Jump server controls; logging to support post-breach hunt activities; Privileged access not equated to superuser/root administration level	No internet access from privileged accounts	Superuser account for root being segregated and not used except in emergency; Follow up threat hunt with well maintained logs	Full user account audit and behavioural monitoring	Persistent threat hunt

Table 17 - Recommended Controls Summary

Finally, a summary of cost and design impacts associated with the recommended controls is available to support risk management decision-making as depicted in Table 18.

System		Desired Threat Effect	Threat Event	Probability of Avoidance		Robustness Metric		Recovery Metric		Reconstitution Metric	
				Cost Impacts	Design Impacts	Cost Impacts	Design Impacts	Cost Impacts	Design Impacts	Cost Impacts	Design Impacts
Architecture Design 001 - Threat 00A											
Bus System	**Bus System User (BSU)**	Deceive	Reused password harvested from online leak and used to obtain user access, allowing for follow up lateral movement and escalation of privileges.	$300k per annum for SOC monitoring; $100k per annum for monitoring	Data feeds and collection procedures and tools to support SOC monitoring	Whitelisting tools - $50k	Whitelisting of applications and associated toolsets	Nil	Logging considerations and access to remote logs	$100k for full audit; $200k per annum for persistent threat hunt; $200k per annum for SOC behavioural monitoring	Requires network taps and remote monitoring capability; behavioural monitoring tools and personnel in SOC
	Bus System Admin (BSA)	Deceive	Reused password harvested from online leak and used to obtain privileged access, leading to complete loss of ground station control of the bus system.	$250 per privileged user (multi-factor)	Reduced freedom for privileged users; multi-factor authentication application; additional logging security; jump server; no internet access from privileged accounts	Nil	Role based access controls established based on principle of least privilege	Nil	Logging considerations and access to remote logs	$100k for full audit; $200k per annum for persistent threat hunt	

Table 18 - Cost and Design Impacts to Implement Controls

The aggregation of this data supports a comprehensive understanding of resilience as it relates to critical systems and a corresponding understanding of risk to support effective prioritisation of resources and effective governance. Although this process requires expertise and some time, the reward of being able to comprehensively assess risk in a standardised manner is considerable. However, the reduction of subjectivity in the process is critical to achieve a degree of standardised and predictable response. As a result, it is highly recommended that these efforts are matched with calibration activities as described by Hubbard and Seiersen (2016). Despite these efforts the subjective nature of such an assessment means that *"cognitive issues such as overconfidence and anchoring typically include significant bias that may result in wildly inaccurate estimates… expert assessments may be seen as an exploratory first step within a more comprehensive approach that include quantitative measures in later steps"* (Ligo, Kott, & Linkov, 2021). As future research, the author proposes the development of a bayesian method to conduct space system resilience and risk assessments.

15. Undertake a Cyber-worthiness Design Principles Review

Conduct a cyber-worthiness design principle review using the following points (Ormrod, Slay, & Ormrod, 2021):

- *Identify the crown jewels – protect the mission and dependent services;*
- *Fail safe and gracefully - default to a secure state with alerts;*
- *Avoid security through obscurity – embrace open design principles;*
- *Implement Role Based Authentication Controls (RBAC) – separate duties;*
- *Provide minimum privilege by default – make escalation hard for the attacker;*
- *Reduce the attack surface - identify vulnerabilities early;*
- *Harden architecture - layer security controls;*
- *Provide incident response capabilities – aligned to predicted adversary profiles;*
- *Embed resilient systems and practices - the spacecraft must be its own root of recovery; and*
- *Identify and protect the weakest links in the security system - prioritise risks and controls.*

In the context of the LEO SV testbed, the principles review is captured below:

Identify the crown jewels:
The BeagleBone Black and KubOS operating system are crown jewels, as they provide the satellite Bus. Without this system the SV will not be operational.
The Adafruit Feather LoRa Radio is a crown jewel, as it provides communication. Without this system the SV cannot communicate.
The Jetson Xavier NX provides the payload for the satellite system and is critical to the SV service clients. Whilst the Payload is a crown jewel, it is secondary to the Bus and Radio systems.

Fail safe and gracefully:
An alert functionality is required within the testbed, together with functionality to support fail safe functions. This will be developed as part of the CY-JAR capability.

Avoid security through obscurity:
Development of a functional testbed is part of the effort to move beyond obscurity as a security function.

Implement Role Based Authentication Controls:
This function is under development within the testbed.

Provide minimum privilege by default:
This function is under development within the testbed.

Reduce the attack surface:
This function is under development within the testbed.

Harden architecture:
This function is under development within the testbed.

Provide incident response capabilities:
This will be developed as part of the CY-JAR capability.

Embed resilient systems and practices:
This will be developed as part of the CY-JAR capability.

Identify and protect the weakest links in the security system:
This will be developed as part of the CY-JAR capability.

16. Improve and Update the system

Iterate back through the system architecture, design, and digital twin setup to enhance security using identified countermeasures. Review any impact on system effectiveness and efficiency. Update and enhance the security of the system. Review documentation and golden images. Refresh security documentation and assessments developed so far, including threat models.

This process is being undertaken as part of the CY-JAR project.

17. Conduct security testing using Digital Twin

Undertake another hands-on penetration test and experiment with the digital twin to test assumptions and confirm the effectiveness of the new controls.

This process is being undertaken as part of the CY-JAR project.

18. Record a new Baseline

Using the digital twin, understand what is considered 'normal' behaviour and build models of the system under various operational conditions where a cyber-attack is not occurring. This will support future testing as well as the detection of unusual behaviour.

This process is being undertaken as part of the CY-JAR project.

19. Undertake Risk Governance Review

Provide senior management with a full risk assessment and document residual risks for governance review and endorsement.

Recommended in a real-life scenario.

20. Iterate

Continuously undertake the process, beginning from determining scope and intelligence collection (1) through to governance review (19). Just as the adversary evolves, the security controls employed on LEO space systems must keep up with the threats and not be allowed to languish.

Recommended in a real-life scenario.

21. Conclusion

The EDTM proof-of-concept has been developed to enable the development and testing of a process to support space system operators conduct cyber-security assessments and develop an enhanced security posture. This proof-of-concept is being expanded and further developed through two activities:
1. The continued development of this report and integration into a single, final report as part of SmartSatCRC Evil Digital Twin Project deliverable three.
2. The commencement of the CY-JAR project to extend upon the EDTM, building a complete response and resilience capability.

References

Abbany, Z. (2018). SpaceX's Starlink satellite internet: It's time for tough talk on cyber security in space. Retrieved from https://www.dw.com/en/spacexs-starlink-satellite-internet-its-time-for-tough-talk-on-cyber-security-in-space/a-42678704

BeagleBoard.org Foundation. (2021a). Beaglebone Black. Retrieved from https://beagleboard.org/black

BeagleBoard.org Foundation. (2021b). Github - Beaglebone Black. Retrieved from https://github.com/beagleboard/beaglebone-black

Burch, R. W. (2019). *Resilient space systems design : an introduction*. [S.l.]: S.l. : CRC PRESS.

Cal Poly CubeSat Laboratory. (2021). CubeSat. Retrieved from https://www.cubesat.org/descriptions

Dallas Semiconductor. (1983). Application Note 83: Fundamentals of RS–232 Serial Communications. Retrieved from https://web.archive.org/web/20170305041309/http://ecee.colorado.edu/~mcclurel/dan83.pdf

Doyle et al. (2020). Flight Software Development for the EIRSAT-1 Mission. Retrieved from https://arxiv.org/ftp/arxiv/papers/2008/2008.09074.pdf

Fireeye. (2021). Advanced Persistent Threat Groups. Retrieved from https://www.fireeye.com/current-threats/apt-groups.html

Frost, J. (2019). Kubos and Ruag partner to provide computer systems for megaconstellations. Retrieved from https://spacenews.com/kubos-and-ruag-partner-to-provide-computer-systems-for-megaconstellations/

Hubbard, D. W., & Seiersen, R. (2016). *How to measure anything in cybersecurity risk*. Hoboken, New Jersey: Hoboken, New Jersey : Wiley.

Kenny, L. (2020). Password Cracking Is Easy: Here's How to Do It. Retrieved from https://kennymuli.medium.com/password-cracking-is-easy-heres-how-to-do-it-875806a1e42a

Kubos Corporation. (2020). Kubos. Retrieved from https://docs.kubos.com/1.21.0/index.html

Kubos Corporation. (2021a). Github - KubOS. Retrieved from https://github.com/kubos/kubos

Kubos Corporation. (2021b). KubOS Community. Retrieved from https://slofile.com/slack/openkosmos

Kubos Corporation. (2021c). KubOS Design. Retrieved from https://docs.kubos.com/1.21.0/kubos-design.html

Ligo, A. K., Kott, A., & Linkov, I. (2021). *How to Measure Cyber Resilience of an Autonomous Agent: Approaches and Challenges*. Paper presented at the AICA 2021, 1st International Conference on Autonomous Intelligent Cyber-defence Agents, Paris, France. https://arxiv.org/ftp/arxiv/papers/2102/2102.00528.pdf

LoRa Alliance. (2021). What is LoRaWAN Specification. Retrieved from https://lora-alliance.org/about-lorawan/

Miranda, D., Ferreira, M., Kucinskis, F., & McComas, D. (2019). A Comparative Survey on Flight Software Frameworks for 'New Space' Nanosatellite Missions. *Journal of Aerospace Technology and Management, 11*. Retrieved from https://www.scielo.br/j/jatm/a/ZNY8mTKcbh8MT5xvbvyytqG/?lang=en

MITRE. (2014). Systems Engineering Guide. Retrieved from https://www.mitre.org/publications/technical-papers/the-mitre-systems-engineering-guide

MITRE. (2021a). MITRE ATT&CK Navigator - Github site - Turla APT. Retrieved from https://mitre-attack.github.io/attack-navigator//#layerURL=https%3A%2F%2Fattack.mitre.org%2Fgroups%2FG0010%2FG0010-enterprise-layer.json

MITRE. (2021b). Turla. Retrieved from https://attack.mitre.org/groups/G0010/

MITRE Corporation. (2014). The MITRE Systems Engineering Guide. Retrieved from https://www.mitre.org/publications/technical-papers/the-mitre-systems-engineering-guide

NASA. (2020). State of the Art of Small Spacecraft Technology. Retrieved from https://www.nasa.gov/smallsat-institute/sst-soa-2020

National Institute of Standards and Technology. (2012). *NIST Special Publication 800-30: Guide for Conducting Risk Assessments*. National Institute of Standards and Technology,
NTIA. (2020). Software Bill of Materials (SBOM). Retrieved from https://www.ntia.gov/files/ntia/publications/sbom_overview_20200818.pdf
Nvidia. (2021). Jetpack SDK. Retrieved from https://developer.nvidia.com/embedded/jetpack
Ormrod, D., Slay, J., & Ormrod, A. (2021). *Cyber-Worthiness and Cyber-Resilience to Secure Low Earth Orbit Satellites.* Paper presented at the ICCWS 2021 16th International Conference on Cyber Warfare and Security.
RUAG. (2020). Next Generation On Board Computer. Retrieved from https://www.ruag.com/system/files/media_document/2020-12/Datasheet_Next%20Generation%20On%20Board%20Computer_Dec%202020.pdf
Seneviratne, P. (2019). Beginning LoRa Radio Networks with Arduino - Appendix 1 - LoRaWAN Channel Plans. Retrieved from https://link.springer.com/content/pdf/bbm%3A978-1-4842-4357-2%2F1.pdf
Tanase, S. (2015). Satellite Turla: APT Command and Control in the Sky. Retrieved from https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/
ThaiCERT. (2021). APT group: Turla, Waterbug, Venomous Bear. *Threat Group Cards: A Threat Actor Encyclopedia.* Retrieved from https://apt.thaicert.or.th/cgi-bin/showcard.cgi?g=Turla%2C%20Waterbug%2C%20Venomous%20Bear
UcedaVelez, T., & Morana, M. M. (2015). *Risk centric threat modeling*: Wiley Online Library.
United States Army. (1994). *FM34-2 Collection Management and Synchronisation Planning*. Washington, DC: United States Department of Defence,
United States Department of Defense. (2013). Task Force Report: Resilient Military Systems and the Advanced Cyber Threat. Retrieved from https://nsarchive2.gwu.edu/NSAEBB/NSAEBB424/docs/Cyber-081.pdf
Whitehouse. (2021). Executive Order on Improving the Nation's Cybersecurity. *12 May 2021,.* Retrieved from https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

USER INFORMED INDUSTRY DRIVEN RESEARCH POWERED

SMARTSAT
COOPERATIVE RESEARCH CENTRE

Building Australia's Space Industry

Australian Government
Department of Industry, Science, Energy and Resources

AusIndustry
Cooperative Research Centres Program

SmartSat CRC Head Office:
Lot Fourteen, Level 3, McEwin Building
North Terrace, Adelaide, SA

info@smartsatcrc.com
smartsatcrc.com