Towards Effective Cybersecurity for Modular, Open Architecture Satellite Systems

Daniel E. Cunningham, Geancarlo Palavicini Jr., Jose Romero-Mariona
SPAWAR Systems Center Pacific
53560 Hull Street, San Diego, CA 92152, 619-553-3731
daniel.cunningham@navy.mil

ABSTRACT

In a fresh twist on early incorporation of cybersecurity engineering, SSC Pacific is embarking on a 5-year small satellite capability development effort for the U.S. Navy. One of the key objectives is to infuse cybersecurity methodologies, technologies, and tools into each phase of the small-satellite life-cycle, from concept design to operations. In this first year, we report out progress to develop a new nanosatellite integration and test laboratory environment that incorporates cybersecurity into every step. The effort leverages technologies and lessons learned from ongoing U.S. Navy-funded research and development of tools and systems for securing commercial Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICS). The paper describes the nanosatellite integration environment being developed at SSC Pacific, along with our approach to overlaying cyber security design and testing into the small satellite acquisition lifecycle. Lessons learned from SCADA/ICS cybersecurity research are then described, along with description of cybersecurity tools and methods applicable to small satellites. Finally, ongoing cybersecurity testing of a Beagle Bone Black processor is described, along with initial findings and comments about how to harden the processor against cyberattack.

LIFECYCLE CYBERSECURITY FOR SMALL SATELLITES

Small satellite development costs are trending downward, but cybersecurity complexity and costs are trending upward. The same commercial off-the-shelf (COTS) technologies and modular, open-network architectures that promise low-cost, rapid development also come with a potentially heavy cybersecurity price tag. This is especially true for small military satellite programs, which are striving to achieve high mission assurance without killing off the culture of innovation, rapid development, and low cost that makes small satellites desirable in the first place.

Cybersecurity challenges often faced by small satellite programs include:

• Cyber security risks inherent in COTS computer processors and software. Vulnerabilities may be more difficult to assess due to proprietary software and designs, but patches to known vulnerabilities may be commercially available.

• Increased reliance on large software libraries (both free/open and vendor provided). Open source libraries can aid software modularity, but fully investigating the potential vulnerabilities can take significant time and resources.

• Internet protocol (IP) architectures. Net-centric architectures aid system accessibility, modularity and integration, but also expand the cyberattack surface.

• Increased automation within ground and satellite systems. Automation aids in on-orbit processing and reduced operational costs, but adds cybersecurity complexity.

The challenges call for lifecycle cybersecurity management solutions, but this is a serious difficulty for small satellite integrators who intentionally do not want to own (or pay for) complete supply chain management of every cable, processor chip, and line of code. Rapid technology prototyping and insertion requires the ability to cobble together busses, payloads, and components into plug and play network architectures using standard interfaces and protocols.

This paper develops an approach to lifecycle cybersecurity that makes use of lifecycle steps the integrator does control, and applies the strategic use of both commercial and government-developed vulnerability assessment tools to drive down the risks inherent in low cost COTS-based procurement and integration. The lessons are focused toward the government integrator, but applicable to integrators in industry and academia as well.

ACTION LABORATORY ENVIRONMENT

SSC Pacific has launched a 5-year capability development effort to infuse cybersecurity methodologies, technologies, and tools into each phase of the small-satellite life-cycle, from concept design, to COTS procurement, to operations. This cyber technology initiative is part of a larger initiative to establish a nationally recognized government nanosatellite integration capability, enabling SSC Pacific to integrate small satellite buses with specialized payloads for military applications and test rapid response small satellite capacities for warfighter. The approach is to build a complete small satellite research and development environment, supporting design, modeling and simulation, prototype fabrication, test and evaluation, and operations. The environment will include basic in-house facilities for integration and testing, as well as access to specialized outside facilities and contracts for commercially-procured supplies and components. (Figure 1).

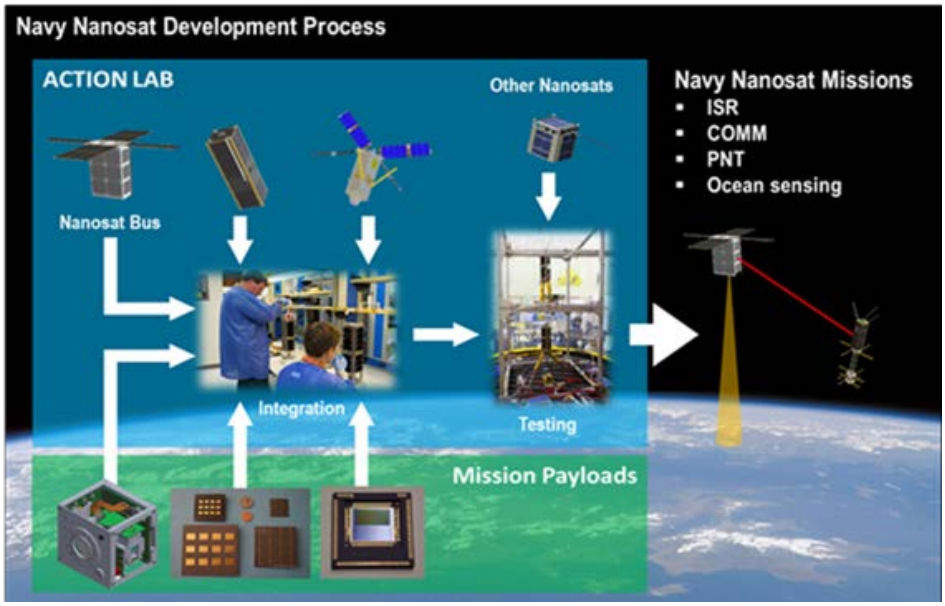

Figure 1: SSC Pacific Accelerated Capability for Integration and Testing of Nanosats (ACTION) Laboratory environment will enable the U.S. Navy to prototype innovative small satellite technologies to meet urgent warfighter needs, including intelligence, surveillance and reconnaissance (ISR), position, navigation, and time (PNT), ocean sensing, and assured communications.

CYBERSECURITY OVERLAY

The SSC Pacific ACTION Lab approach to small satellite cybersecurity is to make maximum use of the small satellite lifecycle portions where the integrator has control. These include, in most cases, the following lifecycle steps:

- Concept design
- Payload and subsystem development and fabrication
- Bus, payload and ground subsystem acceptance and testing
- System-level integration and testing
- Launch, on-orbit operation, and maintenance

Where the integrator does not have direct control, such as COTS-component development and commercial interface adaptation, our approach is to aggressively assess cyber vulnerabilities, through analysis and testing, at the lowest level possible as early as possible. Mitigations are then implemented as part of the system integration.

Within every step life-cycle of a satellite, from design to operations, we aim to blend cybersecurity objectives, tools, and activities into other mission assurance activities. Process time is then conserved by planning information assurance and mission assurance milestones in step with physical and software system milestones. Cyber testing activities are dovetailed with physical and software testing activities. Cybersecurity functions and staff are integrated with the other mission assurance functions and staff. (Table 1).

Table 1: SSC Pacific ACTION laboratory environment inserts cybersecurity objectives and risk reduction measures into each phase of the small satellite development lifecycle

Lifecycle Phase	Cybersecurity Overlay
Concept design	Survey HW/SW vulnerabilities Plan security controls
Payload and subsystem development	Incorporate security code and controls Graybox and blackbox testing of subsystem interfaces
Bus, payload and ground subsystem acceptance	Static-dynamic analysis of payloads, buses Reverse engineering of subsystems and protocols
System-level integration and testing	Dynamic analysis and testing of communications interfaces, signals interference, interception, and injection
Launch, on-orbit operation, and maintenance	Monitor and defend network health

The concept design phase is a critical point to build cybersecurity controls into the entire system from the beginning. A survey of the commercially available devices that meet the functional requirements of the satellite mission is required, in order to select COTS solutions with the best security controls and support. Next, a thorough investigation of known vulnerabilities used against the proposed list of COTS hardware and software should be executed. An integrated report of findings from both the hardware and software attack vectors should be compiled into recommendations to ensure that security controls & defense mechanisms are designed into the base system.

At the payload and subsystem development stage we ensure that proper secure coding practices are followed during any development of interfaces or "glue code". Whitebox testing, such as static code analysis, should be performed to avoid common programing pitfalls like memory corruption, arithmetic boundaries, type conversions, string handling, among other errors. Graybox and blackbox testing of software interfaces should be performed, including, if needed, development of custom software for dynamic testing of interfaces and protocols. Each of these steps requires a thorough understanding of the subsystems involved and may require reverse engineering of subsystems or protocols used in proprietary technologies.

At the bus, payload and ground subsystem acceptance phase, we introduce fuzz-testing of software-based environmental controls, including possible development of custom hardware and/or software for reverse engineering of sensor logic. Special attention should be placed on the potential of malformed input sent through sensors to inject logic into a bus in its many forms [1, 2]. These steps enable early incorporation of mitigations and controls, including effective anti-tampering and active defense mechanisms, which are very costly to add later.

At the system-level integration and testing phase, we perform dynamic analysis and testing of individual sub-systems prior to integrating components into a larger system. As each subsystem is integrated, we employ blackbox-testing techniques for discovery of emergent properties within the integrated system. This type of dynamic analysis aids in the discovery previously hidden vulnerabilities, leading to system hardening and vulnerability mitigation.

At the launch, on-orbit operation, and maintenance phase, cyber testing is expanded to include ground station devices and software used to manage and communicate with satellites on-orbit. The full spectrum attack methodology followed for the satellite testing is applied to the ground station devices, operating systems, and software. Security controls and defense mechanisms must be incorporated into ground station system (i.e. anti-tampering, logging & reporting, etc.), including possible development of new active defensive capabilities. Continued monitoring of the embedded anti-tamper and defensive features within the nanosatellite should also be performed, with a focus on-ground system defense and on-orbit system protection.

NAVY SCADA/ICS RESEARCH

Our small satellite cybersecurity approach leverages technologies and lessons learned from ongoing U.S. Navy-funded research and development of tools and systems for securing commercial Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICS). The research is relevant as SCADA and ICS components are often found in applications similar to satellite platforms and space networks [3]. The goal of this research is to develop a framework that enables the proper and efficient evaluation of technologies for securing networks for various industrial government applications [4]. SCADA and ICS networks allow digital input from computing components, which might be thousands of miles away, to control physical assets such as motors, actuators, pumps, radios, etc. The problem is that the unprotected nature, interconnected paradigm, and global footprint of SCADA networks make them extremely vulnerable to cyber-attack.

As part of the of SCADA/ICS cybersecurity work, SSC Pacific has developed and continues to expand the following lab capabilities [5]:
• Evaluation capabilities (software tools) that enable streamlined evaluation of SCADA security technologies, including security metrics, which provide a granular decomposition of how well current security technologies secure SCADA networks and its components.
• A laboratory environment to baseline a sample SCADA network (with real SCADA equipment) and explore current SCADA vulnerabilities.
• A collaborative environment that enables the sharing/reuse of Cyber SCADA knowledge and test results.

The approach provides not only the process necessary to determine if a certain technology meets security needs, but also provides the metrics to measure how well those needs are met, and a framework to enable the comparison of multiple technologies of interest.

There are some important differences between SCADA/ICS information assurance and traditional information assurance, particularly in the areas of risks and priorities [6]. For example, many ICS require time-critical operations, where the focus is not on throughput, but rather assurance of timely relaying of messages within the system. Availability is a critical ICS property, particularly while the potential for physical damage or injury from service failure exists. As a result some common practices, such as re-booting systems when errors occur, cannot be applied to the ICS environment. ICS environments frequently prioritize availability over confidentiality and integrity, which increases the likelihood of security flaws [7].

Approaching security from the "ground up" rather than addressing security as an add-on, is a key goal. ICS security approaches directly relate to small satellite cybersecurity, as availability and dependability are also often at direct odds with security. Furthermore, the efforts described in this paper are also trying to bring cybersecurity to the forefront of nanosatellite efforts, as we have done with SCADA and ICS efforts,

ICS METHODOLOGY FOR SATELLITES

For SCADA/ICS systems, the standard information assurance paradigm of confidentiality, integrity and availability [8] often flips into a prioritization paradigm of availability, integrity and then confidentiality [9]. As such, many of the "expected" security controls are simply not available in the SCADA/ICS realm. Systems have been designed with process requirements in mind, however cybersecurity is often not one of those requirements. Security is often sprinkled in at the end, leading to bolt-on solutions lacking cohesion instead of designing the system with security in mind from the start.

In our work with SCADA and ICS security, we frequently run into the following hurdles, many of which are also relevant to small satellite security:

• Devices and processes are not designed with cybersecurity in mind

• Systems will run for a very long time, far beyond vendor support availability

• Software updates and patching can be difficult, and slow in implementation

• Reliance on proprietary protocols and proprietary software for critical operations

• Hardware is constrained in terms of memory & CPU specs

• Presence of dissimilar hardware across large processes (and need for interoperability)

• Due to hardware constraints, standard vulnerability assessments can break or disable devices

• Devices designed to be air-gapped or inter-connect with private corporate networks, rather than on enterprise networks or the Internet

• Systems have not been security tested, thus are easily compromised due to large attack surface

• Community is apprehensive to current cybersecurity approaches to critical systems due to concerns of introducing new security vulnerabilities.

• In some instances, the thought of learning penetration testing techniques is seen as irrelevant, or possibly harmful, to securing industrial control networks

• Lack of manpower that understands both industrial control systems and cybersecurity

In application to small satellite systems, many of the above-mentioned hurdles with SCADA/ICS systems share relevance. As such, we propose employing a layered approach to securing small satellite systems, in a similar manner as we employ with SCADA/ICS. Securing each layer starts at the hardware level, and then moves up the architecture stack into the firmware, operating system, application, and network levels. At each stage, we employ a holistic attack methodology to uncover and mitigate potential vulnerabilities early in the system development. Our approach includes implementing security controls and mitigations for each discovered vulnerability and then applying preventative controls at each layer. Controls we employ include

hardening the operating system stack, correcting common misconfigurations, unnecessary services, unused protocols, and superfluous kernel support for non-mission essential capabilities, among others.

SCADA/ICS TOOLS FOR SATELLITES

Some of the capabilities that we are developing in our nanosatellite cyber lab involve transitioning tools and lessons learned from our SCADA/ICS work. Capabilities such as vulnerability discovery and assessments through use of tools such as the DoD-mandated Assured Compliance Assessment Solution (ACAS) scanner that incorporate custom scripts to aid in the assessment without compromising the availability of the overall system.

Essential tools include the ability to discover and exercise vulnerabilities, develop custom exploits, and perform general network penetration testing using tools such as Kali Linux and Metasploit. Equally important is gaining a better understanding of any proprietary software, as well as an understanding of possible missuses of executable code for which source code is not available.

Disassemblers such as IDA Pro can aid in this type of analysis. Hardware hacking techniques can provide a better understanding of the underlying hardware and how underlying vulnerabilities might be used by an attacker to gain access to the software stack. Firmware analysis and binary analysis are also used for inspecting the underlying embedded software running on the COTS and identifying unexpected interfaces or backdoors. Tools such as the Sophia SCADA monitoring tool [10] are being explored to help solve the continued monitoring aspect of space cybersecurity at the ground stations.

LESSONS FROM BEAGLE BONE BLACK

Due to low cost and rapid programmability, Advanced RISC Machine (ARM) processors are increasing in popularity for use in small satellite systems. In order to better assess the cybersecurity risks of using ARMs in small satellites, the SSC Pacific ACTION Lab team has been investigating cyber vulnerabilities and risk mitigations for the Beagle Bone Black (BBB) ARM processor.

We are developing a "test like you fly" approach to cybersecurity testing of small satellite technology and our BBB testing procedure will be used to help establish our process. Network-based vulnerability assessments of services and protocols externally available have been performed. In-depth vulnerability scans of network services and the operating system have been analyzed, as well as analysis of captured traffic to extract an understanding of the amount of information leaked out through external interfaces. This gives us an understanding of what an attacker would be able to gather in the reconnaissance stage of an attack. For the most part the BBB system-under-test (SUT) performed well in this area, as limited information was gleaned through this type of external recon scans.

Next the operating BBB system stack was evaluated for known vulnerabilities, as well as against attack vectors used against similar Linux-base environments. The kernel was tested and various attack scenarios were exercised against the SUT. This proved to be a more fertile ground for exploitation, as the common practice of securing the external interfaces, but not hardening the system internally was found to be the case for this particular SUT. Various recommendations were developed to harden the kernel and internal configurations of the operating system to achieve a more effective defense-in-depth approach to securing this particular BBB implementation.

Further cybersecurity testing of the hardware, firmware and application level software are still underway. Once the results from these test are completed, they will be included in future publications.

CONCLUSIONS

Cybersecurity costs cannot be avoided, but with methodical application of the right tools to the right risks at the right time in the lifecycle, we are building an approach for lifecycle security and assurance into our small satellite systems and ground architectures with a goal to minimize reactive actions and costs. We have shared how ongoing work in the SCADA/ICS realm can be applied to gain insight into the small satellite world, as well as provide capabilities that can be customized to secure small satellite systems and ground stations.

References

1. Govindavajhala, S. and Appel A. "Using memory errors to attack a virtual machine", Proceedings of the IEEE Symposium on Security and Privacy, 2003.
2. Foo Kune, D., et al., "Ghost talk: Mitigating EMI signal injection attacks against analog sensors", Proceedings of the IEEE Symposium on Security and Privacy, May 2013.
3. Romero-Mariona, J., Hallman, R., Kline, M., San Miguel, J., Major, M., and Kerr, L., "Security in the industrial internet of things - the c-sec

approach", Proceedings of the International Conference on Internet of Things and Big Data, pp. 421–428. INSTICC, SCITEPRESS Science and Technology Publications (2016).

4. Romero-Mariona, J., Kline, M., and Miguel, J.S., "C-sec (cyber SCADA evaluation capability): Securing critical infrastructures", Software Reliability Engineering Workshops (ISSREW), 2015 IEEE International Symposium, pp. 38–38. IEEE (2015).
5. Romero-Mariona, J., "Ditec (DoD-centric and independent technology evaluation capability): A process for testing security", Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on, pp. 24–25 (2014). DOI10.1109/ICSTW.2014.52.
6. Drias, Z., Serhrouchni, A., and Vogel, O., "Taxonomy of attacks on industrial control protocols", Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), 2015 International Conference on, pp. 1–6. IEEE (2015).
7. Cruz, T., Barrigas, J., Proena, J., Graziano, A., Panzieri, S., Lev, L., and Simoes, P., "Improving network security monitoring for industrial control systems", Integrated Network Management (I), 2015 IFIP/IEEE International Symposium on, pp. 878–881. IEEE (2015).
8. NIST Comp. Security Division: Underlying Technical Models for Information Technology Security. Special Tech. Rep. 800-33, U.S. Nat. Inst. of Standards and Technology. 2001.
9. Erbes, R.: Sophia.: Idaho National Laboratory, 26 Jan 2012. Web. 3 June 2016. https://files.sans.org/summit/scada12/PDFs/Sophia.pdf>
10. Rueff, G., Thuen C. and Davidson, J., "Shophia Proof of Concept Report", INL Research Library Digital Repository, March 2010. Web. 3 June 2016.