2020-04-27

Architecture-Based Security for UxVs

Berzins, Valdis

Monterey, California. Naval Postgraduate School

SYM-AM-20-090

PROCEEDINGS
OF THE
SEVENTEENTH ANNUAL
ACQUISITION RESEARCH SYMPOSIUM

Acquisition Research:
Creating Synergy for Informed Change

May 13–14, 2020

Published: April 23, 2020

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Architecture-Based Security for UxVs

Valdis Berzins—is a Professor of Computer Science at the Naval Postgraduate School. His research interests include software engineering, software architecture, computer-aided design, and software evolution. His work includes software testing, reuse, automatic software generation, architecture, requirements, prototyping, re-engineering, specification languages, and engineering databases. Berzins received BS, MS, EE, and PhD degrees from MIT and has been on the faculty at the University of Texas and the University of Minnesota. He has developed several specification languages, software tools for computer-aided software design, and fundamental theory of software merging.

Abstract

Current fleet objectives include increased use of unmanned and autonomous systems, including a variety of vehicles. Such systems are attractive due to their potential to increase effectiveness with reduced cost, size, and risk exposure for personnel. Realizing this vision requires better methods of assuring the security of these software-intensive systems, to prevent use of these systems from creating new risks, such as that of adversaries taking control of our systems and using them against us.

Navy acquisition has applied Open Systems Architecture principles to improve affordability of system development, test, evaluation, and upgrade. This paper explores extension of such principles to improve security of unmanned systems within affordable costs. The paper illustrates the proposed principles in terms of a case study on development of a secure architecture for unmanned surface vehicles that support anti-submarine warfare missions.

Keywords: Security, Risk Reduction, Unmanned Systems, Open Architecture, Upgrades, Affordable Certification

Introduction

Many systems of interest to the Navy, including unmanned vehicles of all types (UxVs), have physical as well as software and communications components and belong to the category known as cyber-physical systems. The interactions between the three kinds of components are challenging to analyze and have been studied extensively in the contexts of safety and reliability. Certification of software for these types of systems requires qualitatively different kinds of approaches than for other types of software. This paper proposes that acquisition of these systems also requires different kinds of approaches to ensure that unmanned systems will have the needed security properties.

Security is a key concern in any military application of unmanned systems, because their potential advantages could be reversed if adversaries were able to compromise their control systems. In the worst case, a capable adversary could take control of our unmanned systems and use them against us. Mitigations for this hazard need to fit into a larger context that includes policy considerations and unpredictable future conditions. These considerations have unexpected implications for requirements and architecture that can affect how such systems can be successfully acquired.

One example of unexpected implications is defending against physical intrusion. Even for peacetime applications in friendly areas, vandalism has been a significant problem for unmanned systems, for example, in locations such as U.S. inland waterways. Exposure to this type of hazard will be more severe for unmanned military vehicles in potentially hostile areas.

Mitigating the hazard of physical intrusion requires rethinking system requirements because the rules of engagement and the broader context are different for manned and

unmanned platforms. These new requirements involve cost/benefit trade-offs that can be difficult to quantify and resolve.

For example, mitigations for physical intrusions into manned platforms typically include deterrence based on defensive weapons on the platform and the possibility of large-scale retaliation against such hostile action. This is long-standing tacit knowledge usually taken for granted without explicit mention, which implicitly affects established requirements and designs of military systems. However, these mitigation strategies are less effective for unmanned platforms, since retaliation against an attack is less likely for unmanned platforms than for manned ones, and unmanned platforms are less likely to carry defensive weapons. Multiple considerations limit benefits of defensive weapons on unmanned platforms, including

- Reluctance to authorize autonomous use of weapons due to risk of severe consequences if the software makes an incorrect decision,
- Unanswered questions about who would be legally responsible for damages in case of such a system failure, particularly if these failures occur during peacetime,
- Vulnerability of defensive responses to disruption of communications if use of unmanned weapons requires authorization by remote human decision-makers, and
- Negative impact on cost, size, weight, and energy usage if weapons are installed on unmanned platforms that ideally should be cheap, numerous, and expendable.

Desired system behavior may change substantially from peacetime to active conflict. Consequences for acquisition include possible requirements for rapid system reconfiguration and need for corresponding features in the system architecture to support such requirements. For example, unmanned military vehicles may need a kinetic self-destruct mechanism for use during active conflict, to render captured unmanned vehicles useless to adversaries. However, such capabilities should not be deployed during peacetime, to prevent collateral damage in situations such as fishermen accidentally catching an unmanned underwater vehicle (UUV) in their nets and then being blown up when the UUV detects that something is wrong and self-destructs.

Case Study: USVs for Anti-Submarine Warfare

Anti-submarine warfare (ASW) was identified by the U.S. Navy as a vital mission area for unmanned surface vehicles (USVs) in 2007 [1], second in priority only to minesweeping. The Navy seeks to minimize the exposure of sailors to missions that are dull, dirty, or dangerous. Dull missions require vigilance for extended periods and tax a person's concentration. This matches the detection aspect of the ASW mission, which is like searching for a needle in a haystack. The Navy USV Master Plan [1] specifically recommends USVs for ASW missions known as Maritime Shield and Protected Passage. These missions are illustrated in Figure 1, which is reproduced from [2].

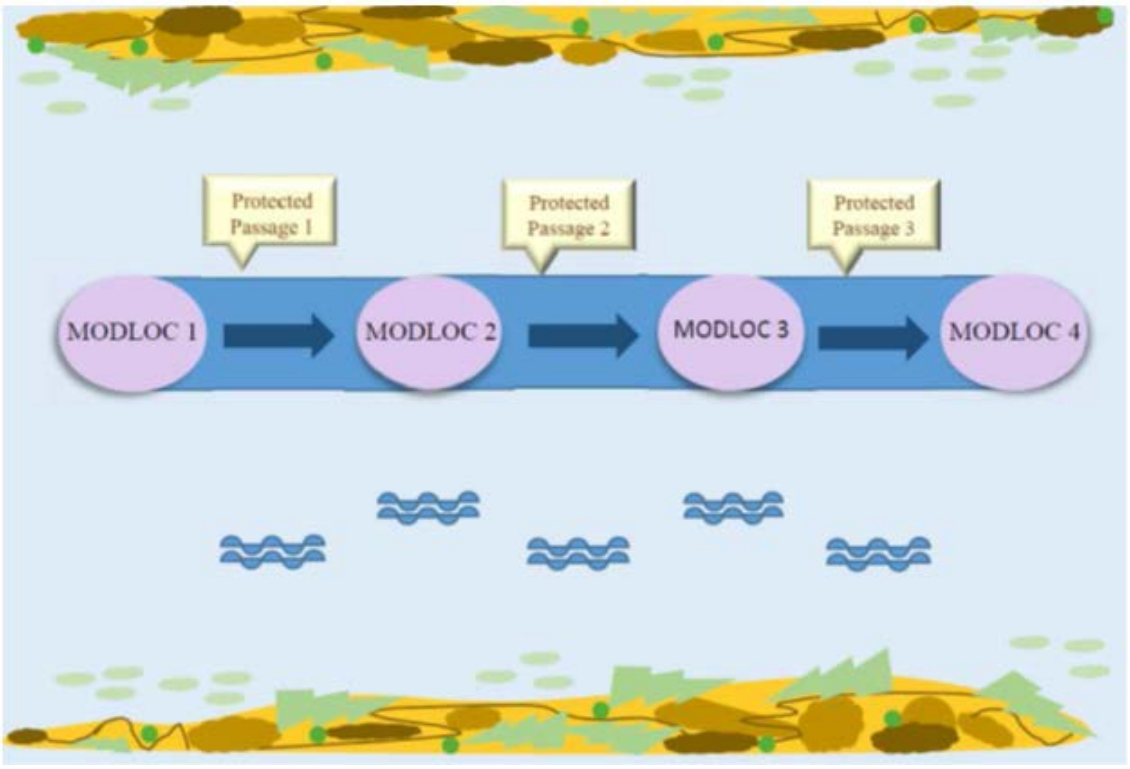

Figure 1. Maritime Shield and Protected Passage

"Miscellaneous Operational Details, Local Operation" (MODLOC) refers to a relatively static area of operations. The purpose of a Maritime Shield mission is to protect a strike group from attack submarines while it remains in a MODLOC, while the purpose of a Protected Passage mission is to provide similar protection while the strike group is in transit from one MODLOC to another [3].

Attack submarines are a threat to strike groups because they are capable of attacking surface ships and are very difficult to detect when submerged, especially for diesel-electric submarines, which are extremely quiet when running on their batteries. Ideally, a Maritime Shield mission seeks to form one or more protective rings around a high value unit in a strike group, such as an aircraft carrier. An attack submarine would have to pass through these rings without detection to become an active threat. Manned platforms such as the LCS are used to perform this mission, but their high cost limits their numbers, and hence the size and coverage of the protective ring. Ideally, the ring should be large enough to provide safe freedom of movement to the high-value unit, and there should be multiple rings to provide defense in depth and mitigate the possibility of imperfect detection. The motivation for using USVs is that lower cost should enable higher numbers and better coverage [4].

Considerations for the Protected Passage mission are similar, except that the protective ring must move along with the strike group. This is harder to accomplish because it is difficult for small USVs to keep up with a transiting strike group in the open ocean, especially in high sea states. Larger USVs such as Sea Hunter [5] are more expensive than smaller ones, although they are still less expensive than manned platforms such as the LCS.

Some of the USV types recommended for ASW missions in the USV Master Plan are shown in Figure 2 and Figure 3, reproduced from [1].

Figure 2. Snorkeler Class USV [1]

Figure 3. Fleet Class USV [1]

Student projects at NPS developed requirements analyses and architectural designs for hypothetical USVs that perform the Maritime Shield and Protected Passage missions. These works have contributed to the ideas that have been refined and presented here.

Security Mitigations for UxVs

Security threats and mitigations have been analyzed in detail in [2]. Here we summarize salient points and point out implications for requirements, open architectures, and acquisition.

Physical Intrusions

A factor that distinguishes security concerns for UxVs from those of other kinds of military systems is that they are more susceptible to capture by adversaries. Since physical tamper resistance has limits, it is extremely difficult to protect systems from attackers that have physical access [6]. Mitigations for this hazard include the following:

- Limit the sensitive information contained in UxVs to the bare minimum needed.
- Encrypt all sensitive information held in non-volatile memory.
- Protect the encryption keys with multiple redundant methods for defense in depth.
- Use multiple methods for sensing intrusions and erase sensitive data if intrusions are detected.
- Provide locally controlled physical self-destruct mechanisms that can be activated during conflict.

Mitigations such as these should be formulated as standard reusable requirements fragments that can be incorporated by reference into any contract for unmanned military systems.

Since professional adversaries will eventually find ways to compromise barriers, an arms race in developing countermeasures, counter-countermeasures, and so on is likely to ensue. This implies the following:

1. Frequent changes to requirements related to the above concerns are to be expected,
2. The methods used to mitigate these hazards should be isolated as separate components in the open architecture, with standard interfaces controlled by the government, and
3. An ongoing improvement process producing a series of frequently updated implementations should be included in the acquisition plan.

This part of development is a prime candidate for rapid prototyping, Middle-Tier Acquisition (MTA), and continuing penetration testing and improvement to discover and counter vulnerabilities before adversaries do so. Due to the similarities of concerns for all types of UxVs, this process should be common to all programs developing unmanned systems. The Navy and the DoD need methods for allocating resources, distributing results, and coordinating fielding of improvements developed in this way.

Spoofed Communication

Communication is critical for unmanned systems, especially those that are not fully autonomous. If a UxV is captured and its encryption keys are compromised, they could be used to send misleading information to the strike group, or to infect ships' computers with malware. Mitigations include:

- Ensuring that every UxV has a unique encryption key that is different for each mission, so that compromising the keys of one UxV does not affect the

operation of the others and the time window for potential spoofed interactions with the strike group is limited.

- Monitoring the locations of the UxVs from the controlling ships, and block communications from them if they deviate too far from planned positions.
- Limiting connections between UxV communications and ship's networks, and intensively monitor/analyze traffic along those connections, looking for potential cyber threats.
- Using frequency hopping for communication, with a different pattern for each mission and UxV, to make the communications more difficult for adversaries to monitor and manipulate.
- Protecting the frequency hopping patterns with tamper resistance and intrusion detection.

Acquisition implications are similar: all of the above should be incorporated into standard reusable requirements modules and open architecture components, shared across different types of UxVs, and refined by a continuous prototyping and red teaming process.

Compromised Sensors

Since GPS signals are not encrypted, they can be spoofed. Adversaries could therefore control the movement of our unmanned systems by manipulating these signals if they rely solely on GPS for determining where they are and deciding which way they should move. Implications are as follows:

- Own platform location should be a standard service in the open architecture for UxVs.
- The architecture should include a standard structure for multiple sources of position data.
- This structure should include standard interfaces and methods for fusing the data, detecting inconsistencies, and reacting to them as possible indications of attacks in progress.
- In the longer term, an encrypted channel should be added to GPS and feeds from other remote positioning systems.

Standard requirements and implemented components for these functions should be developed that can be shared across different types of UxVs.

Implications for System Architecture and Acquisition

Many different types of unmanned vehicles and systems have similar security concerns. This suggests that principles of Navy Open Architecture should be applied and extended to organize, manage, and continue to improve the best known mitigations for these concerns, and to ensure that the same solutions, components, and improvements can be systematically shared across this family of diverse systems that nevertheless share many common characteristics.

The above approach is easiest when requirements can be factored into independent parts, variants in each part can be characterized by feature parameters [6], and each part can be allocated to a single subsystem. In such cases, these subsystems can become components in a standardized open architecture, standardized interfaces can be developed that can be incorporated into the architecture, and system improvement can proceed by replacing these subsystems with variants that conform to the same standardized interface. The key to this approach is that all UxVs should share the same architecture, or at least the

same architecture fragments that include the common subsystems and their interfaces to the rest of the UxV. Such fragments should be formulated as Technical Reference Frameworks (TRFs) [7].Benefits enabled by this approach include the following:

- Streamlining acquisition of security improvements by contracting for new versions of specific subsystems, rather than entire systems. Each development step can then be smaller, faster, and less expensive.
- Reuse of security-critical subsystems and associated improvements across many different unmanned platforms.
- More competition, leading to better and more cost-effective products.
- Less time lag between discovery of new solutions and fielded capabilities.

Unexpected changes in circumstances can require changes in the capabilities of unmanned systems. In the extreme case, this can include sudden changes in mission due to emergence of new threats, which would require rapid and reliable system reconfiguration in the field. Acquiring systems with such capabilities poses new challenges, such as how to write requirements for such capabilities into a contract, how to ensure that they can be met before including them in a contract, and how to check whether they have or have not been met when the product is delivered at the end of development.

Conclusions and Recommendations for Future Work

Conclusions

Maintaining the security of unmanned systems is a dynamic process that can be strongly affected by changing circumstances. Responding to such changes requires capabilities for rapid reconfiguration of these systems, possibly in the field with short notice. This suggests that the requirements and architectures of such systems should be organized around standardized, modular parts, and that each of those parts should have multiple variants matching likely future circumstances. The intent is to enable rapid reconfiguration by component swapping, matching capabilities to current situations using a plug-and-fight concept.

As illustrated by the example in the first section of this paper, one possible event that can trigger the need for rapid reconfiguration is a transition from peacetime to active conflict. There are many other possible triggers, such as discovery of a new software vulnerability, development of a method for breaking an encryption method that was previously believed to be secure, development of new adversary capabilities to deny wireless communications, to falsify sensor readings, to expose sensitive information by reverse engineering captured unmanned systems, and so on.

The plug-and-fight vision raises significant challenges for test and evaluation. Previous work has shown that cost of testing reconfigurable systems increases rapidly with the number of independent configuration choices, so that traditional test and evaluation methods become unaffordable if the goal is to precertify high reliability for all possible configurations. Current practice to avoid this problem is to certify only the configurations that are actually fielded. This slows down reconfiguration because each new configuration would need testing before it can be fielded. Some mitigations for this problem are described in [6], [7], [8], [9], [10], [11], and [12].

Recommendations

- Develop a Technical Reference Framework (TRF) for unmanned systems that defines fragments of system and software architecture for unmanned platforms. This TRF should include standard interfaces for services needed

for all unmanned platforms, such as mitigations for common security risks that include physical intrusion, spoofed communication, and compromised sensor feeds. These interfaces should be standardized because the details of their behavior will need to be tailored to different situations and different platforms, and variants may need to be swapped out in the field as circumstances change. Standardized interfaces are needed to effectively support interchangeable components that can adapt capabilities in a plug-and-fight mode.

- Establish a Navy or Joint organization for the purpose of developing, negotiating, refining, evaluating, and managing improvements to the standardized interfaces recommended above and provide it with the resources needed to support such an ongoing effort.

References

[1] "The Navy Unmanned Surface Vehicle (USV) Master Plan," 23 July 2007. http://www.navy.mil/navydata/technology/usvmppr.pdf.

[2] Luqi, "Control Systems for Unmanned Surface Vehicles in Anti-Submarine Warfare," Technical Report NPS-CS-001R, 2017.

[3] S. Savitz et al., "U.S. Navy Employment Options for Unmanned Surface Vehicles (USVs)," RAND Corporation, 2013. http://www.rand.org/content/dam/rand/pubs/research_reports/RR300/RR384/RAND_RR384.pdf.

[4] S. Fahey, "Software Architecture for Anti-Submarine Warfare Unmanned Surface Vehicles," M.Sc. in Computer Science, NPS, September 2016.

[5] "ACTUV Christening Press Kit," 2016. [Online]. https://www.darpa.mil/attachments/ACTUVChristeningPressKitPostEvent1.zip.

[6] V. Berzins, M. Rodriguez, and M. Wessman, "Putting Teeth into Open Architectures: Infrastructure for Reducing the Need for Retesting," in *Proceedings of the Fourth Annual Research Symposium—Acquisition Research: Creating Synergy for Informed Change,* pp. 285–312, 16–18 May 2007.

[7] V. Berzins, "Which Unchanged Components to Retest after a Technology Upgrade," in *Proceedings of the Fourth Annual Research Symposium—Acquisition Research: Creating Synergy for Informed Change*, pp. 142–153, 14–15 May 2008.

[8] V. Berzins and P. Dailey, "How to Check If It Is Safe Not to Retest a Component," in *Proceedings of the Sixth Annual Research Symposium—Acquisition Research: Defense Acquisition in Transition*, pp. 189–200, 12–14 May 2009.

[9] V. Berzins and P. Dailey, "Improved Software Testing for Open Architecture," *Proceedings of the Seventh Annual Research Symposium—Acquisition Research: Creating Synergy for Informed Change*, pp. 385–398, 11–13 May 2010.

[10] V. Berzins, P. Lim, and M. B. Kahia, "Test Reduction in Open Architecture via Dependency Analysis," in *Proceedings of the Eighth Annual Acquisition Research Symposium*, pp. 333–344, 11–12 May 2011.

[11] V. Berzins, "Certifying Tools for Test Reduction in Open Architecture," in *Proceedings of the Ninth Annual Acquisition Research Symposium,* 2012, Monterey, CA.

[12] V. Berzins, "Combining Risk Analysis and Slicing for Test Reduction in Open Architecture," in *Proceedings of the Eleventh Annual Acquisition Research Symposium*, pp. 199–210, 13–15 May 2014, Monterey, CA.

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net