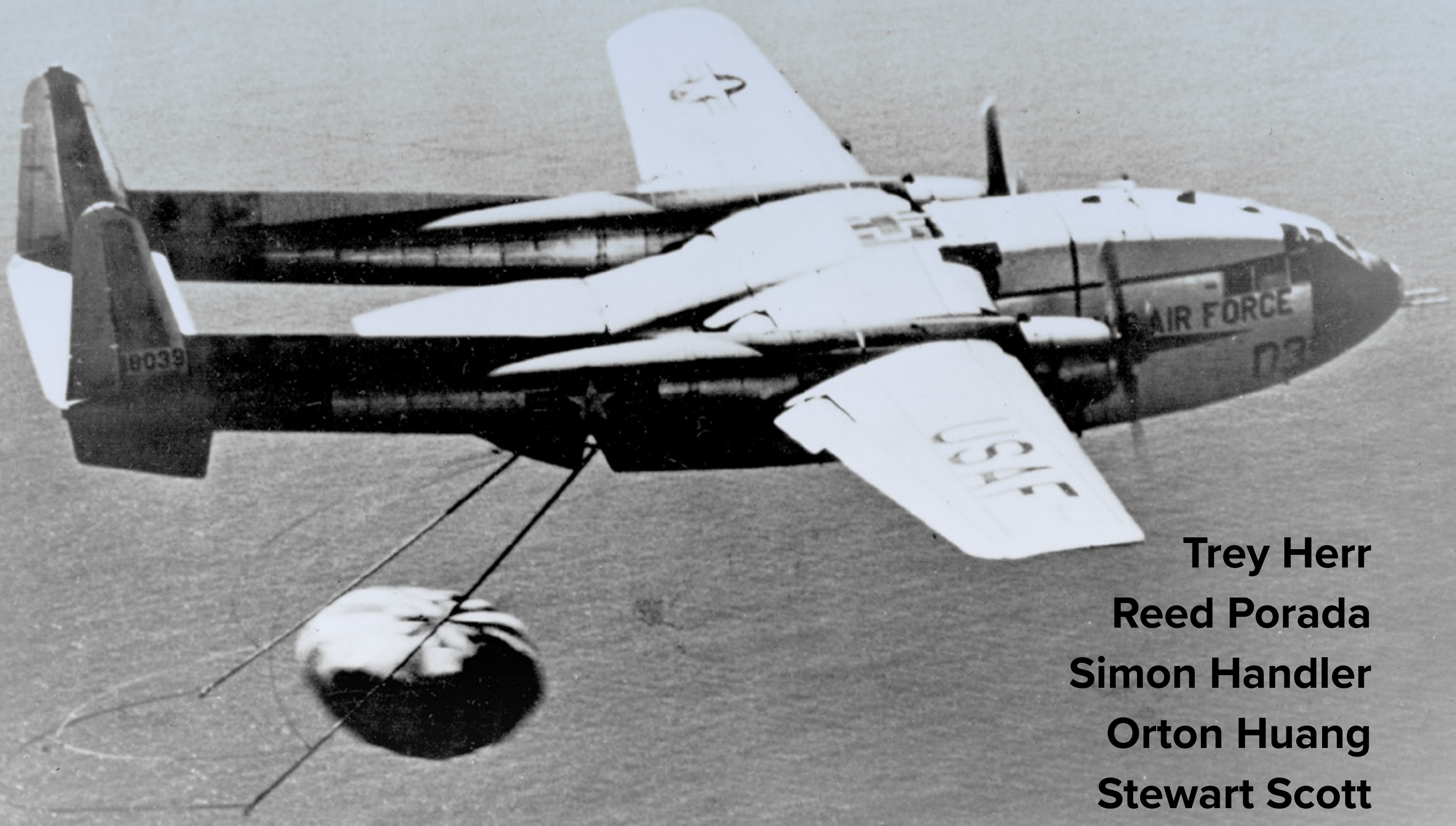
Atlantic Council
SCOWCROFT CENTER
FOR STRATEGY AND SECURITY

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
244 Wood Street ▪ Lexington, Massachusetts 02421-6426

HOW DO YOU FIX A FLYING COMPUTER?

Seeking Resilience in Software-Intensive Mission Systems



Trey Herr
Reed Porada
Simon Handler
Orton Huang
Stewart Scott
Robert Lychev
Jeremy Mineweaser

CYBER STATECRAFT
INITIATIVE

RESILIENCE

Scowcroft Center for Strategy and Security

The ***Scowcroft Center for Strategy and Security*** *works to develop sustainable, nonpartisan strategies to address the most important security challenges facing the United States and the world. The Center honors General Brent Scowcroft's legacy of service and embodies his ethos of nonpartisan commitment to the cause of security, support for US leadership in cooperation with allies and partners, and dedication to the mentorship of the next generation of leaders.*

Cyber Statecraft Initiative

The ***Cyber Statecraft Initiative*** *works at the nexus of geopolitics and cybersecurity to craft strategies to help shape the conduct of statecraft and to better inform and secure users of technology. This work extends through the competition of state and non-state actors, the security of the internet and computing systems, the safety of operational technology and physical systems, and the communities of cyberspace. The Initiative convenes a diverse network of passionate and knowledgeable contributors, bridging the gap among technical, policy, and user communities.*

Atlantic Council

SCOWCROFT CENTER
FOR STRATEGY AND SECURITY

CYBER STATECRAFT
INITIATIVE

HOW DO YOU FIX A FLYING COMPUTER?

Seeking Resilience in Software-Intensive Mission Systems

Trey Herr,[1] Reed Porada,[2] Simon Handler,[1] Orton Huang,[3]
Stewart Scott,[1] Robert Lychev,[3] and Jeremy Mineweaser[3]

ISBN-13: 978-1-61977-148-2

Cover: US Air Force C-119J recovers a CORONA capsule returned from space. *Source:* US Air Force

This report is written and published in accordance with the Atlantic Council Policy on Intellectual Independence. The author is solely responsible for its analysis and recommendations. The Atlantic Council and its donors do not determine, nor do they necessarily endorse or advocate for, any of this report's conclusions.

December 2020

[1] Atlantic Council
[2] Boston Cybernetics
[3] MIT Lincoln Laboratory

Table of Contents

Executive Summary

Defense organizations, by nature, confront unanticipated and highly impactful disruptions, but must continue to operate using complex mission systems. They must adapt these systems to withstand surprise and accomplish defined objectives despite disruption and the behavior of adversaries. It is crucial to understand a system as more than hardware or software—it is a combination of people, organizational processes, and technologies. Mission resilience is the ability of a mission system to prevent, respond to, and/or adapt to both anticipated and unanticipated disruptions, optimizing efficacy and long-term value. This means overcoming sophisticated cyberattacks and managing the risk of systemic software vulnerabilities, but it also encompasses changing operating environments, adversary innovation, and unexpected failures. Resilient mission systems should have the capacity to continue mission essential operations while contested, gracefully degrading through disruption rather than collapsing all at once.

Resilience is a key challenge for combat mission systems in the defense community as a result of accumulating technical debt, outdated procurement frameworks, and a recurring failure to prioritize learning over compliance. The result is brittle technology systems and organizations strained to the point of compromising basic mission functions in the face of changing technology and evolving threats.

Resilience is not a novel concept, but it tends to be presented as a technology issue. While technologies provide the most intuitive and concise examples for understanding resilience, people are responsible for selecting a system's purpose and mission, designing a system's technologies, and enforcing organizational processes within a system. This report provides actionable strategies and practices to combat mission system program owners who manage complex, software-intensive systems, enabling them to reshape their organizations to perform in a state beyond normal operational boundaries—otherwise known as graceful extensibility.[1]

This report translates concepts of mission resilience into practice for defense organizations. Drawing from academia, industry, and government, the authors distill four principles and specific activities as a framework for long-term change that defense organizations should adopt in pursuit of graceful extensibility: embrace failure, always be learning, improve your speed, and manage trade-offs and complexity. These principles build on previous work and combine discussion of procurement with operations, leaning on concepts and phrases used in slightly different ways by communities, like command and control (C2), which might think of managing trade-offs at speed as an issue of agility and biomimetics. Within each of these four principles are tangible practices that defense organizations can adopt to be more resilient:

Embrace Failure: Everyone and everything fails eventually—software developers are no different—so defense organizations must develop a healthy relationship with failure in order to succeed. Unwillingness to take risks creates a fear of failure and a resulting brittle culture, the consequences of which outweigh the failure itself. Practices that defense organizations can adopt to embrace failure include chaos engineering and planning for loss.

Improve Your Speed: The Department of Defense (DoD) must make improving speed of adaptation and development a focus in its transformation toward more resilient mission systems. Antiquated acquisition policies, misapplied bureaucratic oversight, and siloed knowledge make it more difficult for DoD programs to deliver capabilities than should or could be the case. This principle emphasizes speed and tight feedback loops, informed by agile methodologies of continuous integration and delivery.

Always Be Learning: Defense organizations operate in a highly contested cyber environment. As the DoD grows more complex, it becomes increasingly important how the organization learns and adapts to rapidly evolving threats. This process of continual learning embraces experimentation and measurement at all levels of systems as a tool to define and drive improvement.

Manage Trade-Offs and Complexity: Project management is a balancing act among cost, time, scope, and quality for defense organizations. The DoD should work to improve mission system programs' understanding of the trade-offs between near-term functionality and long-run complexity as well as their impact on a system's resilience.

Mission resilience must be a priority area of work for the defense community. Resilience offers a critical pathway to sustain the long-term utility of software-intensive mission systems, while avoiding organizational brittleness in technology use and resulting national security risks. The

[1] David D. Woods, "The theory of graceful extensibility: basic rules that govern adaptive systems," *Environment Systems and Decisions* 38 (2018): 433-457, accessed July 14, 2020, 10.1007/s10669-018-9708-3.

United States and its allies face an unprecedented defense landscape in the 2020s and beyond. The capabilities of both long-identified and novel adversaries continue to evolve, and bureaucratic conflict waged today will shape outcomes on battlefields in the years to come. For the first time in more than four decades, the prospect of significant great power conflict cannot be ruled out and neither the United States nor its allies can afford to acquire, maintain, and deploy mission systems with a mindset shaped in those decades past.

Introduction

The United States' most expensive weapons system, the Lockheed Martin F-35 Lightning II, was designed as a fifth-generation joint strike fighter for service in decades to come. A major selling point to differentiate the F-35 from other aircraft[2] was the Autonomic Logistics Information System (ALIS),[3] the IT backbone of the system intended to govern F-35 operations, including (but not limited to) flight scheduling, maintenance and part tracking, combat mission planning, and threat analysis.[4]

However, ALIS has been plagued by flaws and vulnerabilities, including several identified in early testing that still remain unfixed.[5] Where security audits and testing have occurred, they've taken place in isolated laboratories incapable of simulating the full breadth of the aircraft's digital attack surface. Officials, fearing failure, worried that real-world full-system tests would interrupt operations and disrupt development of the ALIS software.[6] Software vulnerabilities and programmatic issues are hampering the servicemembers whom ALIS was intended to support: "one Air Force unit estimated that it spent the equivalent of more than 45,000 hours per year performing additional tasks and manual workarounds" due to the system's malfunctions.[7] ALIS' inefficiencies have become so acute and costly that the Department of Defense (DoD) opted to overhaul it with the cloud-based Operational Data Integrated Network (ODIN), built by the same vendor.[8]

The F-35 is a combat aircraft—and a software-intensive one at that. ALIS and similar backbone IT systems promise great value, but have barely gotten off the ground. The DoD has demonstrated an inability to manage complexity and develop robust and reliable mission systems even in a relatively benign environment. A conflict or more contested environment would only exacerbate these issues. The F-35 is not alone in a generation of combat systems so dependent on IT and software that failures in code are as critical as a malfunctioning munition or faulty engine—other examples include Navy ships and military satellites.[9] Indeed, encapsulating the centrality of the aircraft's complex IT backbone, now retired Air Force Chief of Staff Gen. David L. Goldfein once posited, "when I see the F-35, I don't see a fighter. I see a computer that happens to fly."[10] Software-intensive mission systems of this and future eras will form the backbone of US and allied military capabilities. These capabilities will continue to be asked to adapt to new roles and do more with less, as budgets are rightsized and adversaries evolve. But existing acquisition, development, and deployment methodologies continue to fail these systems, failing to keep pace with the demands of users in the field and struggling to manage the complexity of ever larger and more integrated software and hardware projects.

To ensure mission systems like the F-35 remain available, capable, and lethal in conflicts to come demands the United States and its allies prioritize the resilience of these systems. Not merely security against compromise, mission resilience is the ability of a mission system to prevent, respond to, and adapt to both anticipated and unanticipated[11] disruptions, to optimize efficacy under uncertainty, and to maximize value over the long term. Adaptability is measured by the capacity to change—not only to modify lines of software code, but to overturn and replace the entire

[2] Lockheed Martin, "Multi-Mission Capability for Emerging Global Threats," F-35 Lightning II, accessed July 14, 2020, https://www.f35.com/about/capabilities.

[3] Lockheed Martin, "Multi-Mission"; Joseph Trevithick, "Replacement For F-35's Troubled ALIS Cloud-Based Brain Rebranded ODIN And Is Still Years Away," *Drive*, January 16, 2020, https://www.thedrive.com/the-war-zone/31861/replacement-for-f-35s-troubled-alis-cloud-based-brain-rebranded-odin-and-is-still-years-away.

[4] Lockheed Martin, "Autonomic Logistics Information System," accessed July 14, 2020, https://www.lockheedmartin.com/en-us/products/autonomic-logistics-information-system-alis.html.

[5] Office of the Secretary of Defense, US Department of Defense, "F-35 Joint Strike Fighter (JSF)," FY19 DOD Programs, https://www.dote.osd.mil/Portals/97/pub/reports/FY2019/dod/2019f35jsf.pdf?ver=2020-01-30-115432-173.

[6] Jeremy Herb, "Obama, Netanyahu meet at the White House," *Politico*, November 9, 2015, https://www.politico.com/tipsheets/morning-defense/2015/11/obama-netanyahu-meet-at-white-house-defense-world-sounds-an-optimistic-note-at-reagan-conference-f-35-cybersecurity-tests-delayed-211151.

[7] *F-35 Aircraft Sustainment: DOD Faces Challenges in Sustaining a Growing Fleet*, US House of Representatives Committee on Armed Services Subcommittees on Readiness and Tactical Air and Land Forces, 116th Cong. (2019), (testimony by Diana Maurer, director, defense capabilities and management, U.S. Government Accountability Office), November 13, https://www.gao.gov/assets/710/702614.pdf.

[8] Dan Grazier, Uncorrected Design Flaws, Cyber-Vulnerabilities, and Unreliability Plague the F-35 Program, Project on Government Oversight, March 24, 2020, https://www.pogo.org/analysis/2020/03/uncorrected-design-flaws-cyber-vulnerabilities-and-unreliability-plague-the-f-35-program.

[9] U.S. Government Accountability Office, Report to the Committee on Armed Services, US Senate, "Weapon System Cybersecurity: DOD Just Beginning to Grapple with Scale of Vulnerabilities," GAO-19-128, October, 2018, accessed July 14, 2020, https://www.gao.gov/assets/700/694913.pdf; Space Dynamics Laboratory, "Satellite Software Systems," Utah State University, accessed August 17, 2020, https://www.sdl.usu.edu/programs/satellite-software-systems.

[10] Sydney J. Freedberg Jr., "'A Computer That Happens To Fly': USAF, RAF Chiefs On Multi-Domain Future," *Breaking Defense*, April 16, 2018, https://breakingdefense.com/2018/04/a-computer-that-happens-to-fly-usaf-raf-chiefs-on-multi-domain-future.

[11] Simon R. Goerger, Azad M. Madni, and Owen J. Eslinger, "Engineered Resilient Systems: A DoD Perspective," *Procedia Computer Science* 28 (December 2014): 865-872, accessed July 14, 2020, https://doi.org/10.1016/j.procs.2014.03.103.

Lockheed Martin's test pilot checks a F-35 simulator before Israel's Defence Minister Moshe Yaalon's visit to the Israeli Air Force house in Herzliya, Israel. *Source:* Reuters/Baz Ratner

organization and the processes by which it performs the mission, if necessary. Any aspect that an organization cannot or will not change may turn out to be the weakest link, or at least a highly reliable target for an adversary. Moving beyond the issues that plague programs like the F-35's ALIS—a complex and evolving system in an ever-changing operational environment—will only be possible by coming to terms with past problems. But, by doubling down with similarly designed systems such as ODIN, defense organizations are bound to repeat the same expensive mistakes.

Efforts to invest in new software acquisition, and to reform policy impacting mission systems, are regularly proposed and attempted but continually fall short. At the same time, adversary capabilities, including kinetic platforms and cybered effects, evolve more rapidly than those of blue forces, and recurring, systemic difficulties in embracing commercial off-the-shelf (COTS) technology continue. The DoD's uneven move to adopt cloud computing, slow by comparison to Fortune-500-scale organizations, exemplifies this problem.

For decades, studies have recognized the vital importance of software as an integrator of defense mission systems, and they have put forth strong recommendations on how to improve it. For equally as long, however, frustrations have mounted over lack of implementation and continued stagnation in the defense enterprise. As pointed out in the Defense Innovation Board's congressionally mandated 2019 Software Acquisition and Practices Study, "the problem is not that we do not know what to do, but that we are simply not doing it."[12] The study highlights two people problems—middle management and congressional mismatch—as reasons for lack of progress.

[12] J. Michael McQuade et al., "Who Cares: Why Does Software Matter for DoD?" in *Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage* (Washington, DC: Defense Innovation Board), May 3, 2019, https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF.

In addition to these organizational and oversight factors, the DoD is making changes to the way it acquires software, but these need to address software embedded in physical and safety-critical systems, as well as where the tolerance for failure and experimentation is lower and resulting program models more risk-averse.[13] Kessel Run is a useful model to bring continuous integration/continuous deployment into responsively developed software. The scale of these projects is small enough to avoid significant systems or project management overhead; the security requirements of these projects invite relatively straightforward classification and minimal compartmentalization; and the development time and life cycle length of these projects complement the software factory approach.

But resilience requires more than new technology incubators—it necessitates taking development out of a silo and knitting it together with users, as well as security organizations like the 16th Air Force and the 10th Fleet. For more complex projects, those with more dependencies on legacy systems, and those which are embedded in or significantly impact safety-critical and physical systems, the once-off hybrid model may be insufficient.

This report addresses the significant disconnect between contemporary understandings of resilience in defense organizations and the importance of software-intensive mission systems. By focusing the conversation on adaptation, this joint effort between MIT Lincoln Laboratory and the Atlantic Council's Cyber Statecraft Initiative, under the Scowcroft Center for Strategy and Security, develops a working-level concept of mission resilience and uses this concept, along with specific practices from government, academia, and industry, to guide mission resilience in defense organizations.

Fundamentally, mission resilience is built on three pillars: robustness, the ability of a system to resist or negate the impact of disruption; responsiveness, the ability of a system to provide feedback on and incorporate changes in response to the impact of disruption; and adaptability, the ability of a system to change itself to continue operating amid disruption over its full life cycle, even when those changes dictate an adjustment of the system's objectives. This definition encompasses the ability to encourage and enable systemic adaptation and expands beyond resistance to disruption (e.g., defects, faults, attacks, and even intentional change). These pillars function in symbiosis and when exercised in concert with one another create mission resilience, an attribute that is greater than the sum of its parts.

Sustained progress and continual change are critical to the resilience of defense organizations; in this, Richard Cook's discussion of the human skeleton is an apt metaphor.[14] Despite its static appearance, human bones are continuously remodeled and replaced roughly every ten years—a process spurred by mechanical strain that enables the destruction of old bone and creation of new bone. This "dynamic balance" requires incessant inputs and energy in order to maintain bone density and prevent skeletal weakening that can be prone to disease and breakage.[15] In the event of a break, it is critical that bone be put under conditions for its natural resilience to do its best work.

Some organizations in the private sector have set an example in harnessing this natural resilience through high-tempo, continuous change. Unfortunately, inadequate strain lines have hampered defense organizations' pursuit of resilience and led to deformity. The next four sections offer four principles for defense organizations' pursuit of mission resilience—1) embrace failure, 2) improve your speed, 3) always be learning, and 4) manage trade-offs and complexity, followed by a conclusion. Each section explains concepts of mission resilience as distilled to that principle, as well as previous relevant research and discussion from government, academia, and industry. Each section concludes with actionable practices and specific recommendations for reforming acquisitions policy, the operation and management of mission systems and their program offices, and their integration into combat units.

[13] Craig Ulsh and Maj Zachary McCarty, "Vignette 2 – F22: DevOps on a Hardware Platform," Defense Innovation Board (DIB) Software Acquisition and Practices (SWAP) Study, May, 2019, https://media.defense.gov/2019/May/01/2002126695/-1/-1/0/VIGNETTE%202%20-%20F22%20DEVOPS%20ON%20A%20HARDWARE%20PLATFORM.PDF.

[14] Dr. Richard Cook, "A Few Observations on the Marvelous Resilience of Bone and Resilience Engineering," REdeploy, January, 2020, video, 36:53, https://re-deploy.io/2019/videos/11-cook.html.

[15] P. Bergmann et al., "Loading and Skeletal Development and Maintenance," *Journal of Osteoporosis*, 2011 (December 2010), accessed July 15, 2020, https://doi.org/10.4061/2011/786752.

1 – Embrace Failure

Mission resilience emphasizes the ability to overcome local failures and uncertainty and to tolerate and adjust to unknown disruptions in order to continue accomplishing a defined set of objectives.

Mission resilience is built on three pillars:

1. **Robustness—the ability of a system to resist or negate the impact of disruption.**
   A robust system is one that persists in pursuing its objectives despite disruption. Like a Nokia 3310 or an AK-47, the system is reliable and functions in a degraded state despite adversity. A robust system does not break easily and gradually degrades before completely ceasing to function.
2. **Responsiveness—the ability of a system to provide feedback on and respond to disruption.**
   A responsive system provides its operators with reliable information on its status and smoothly incorporates their adjustments to maintain maximum functionality in the face of change. Responsiveness is as much about mutual communication and sensitivity as speed—like a car with exquisite steering, the wheels respond quickly to delicate movements of the steering wheel, and the driver can feel the contours of the road across a variety of surfaces.
3. **Adaptability—a system's ability to change itself to continue operating amid disruption over its full life cycle, even when those changes dictate an adjustment of the system's objectives.**
   If responsiveness is a system's capacity to provide feedback on and respond to disruption, adaptability is the system's capacity to be changed over time as warranted by future events and evolving conditions.[16] *Adaptability has a wider mandate than the other pillars. Not only must the components of a system be capable of modification and adjustment, like the modular design of object-oriented programming languages or the replaceability of a World War II Jeep's parts, but the entire system itself must be adaptable in a holistic sense, capable of addressing new and evolving objectives.*

These pillars support each other, fashioning mission resilience into an attribute that is greater than the sum of its parts and intelligible to mission system owners. Robustness allows a system to withstand initial disruption long enough to make adjustments, allowing responsiveness and adaptability. Responsiveness guides the changes made to a system and allows them to proliferate throughout quickly enough to respond to disruption, enabling and augmenting adaptation. Adaptability preserves the long-term utility and functionality of a system, and guides its realignment with fluid mission objectives, preserving its robustness and responsiveness.

There will always be new, unprecedented threats for which no established contingency plan exists and about which no accurate predictions can be made. For this reason, mission resilience, and the ability to communicate its state to mission owners, is critical. The coronavirus pandemic, for example, has forced systemic changes to how mission owners operate.[17] This led to corresponding adaptation in how these systems were maintained and, in some cases, employed.[18] Notably, these responses were led and enabled by people who recognized the signals and adjusted and adapted the systems. The COVID-19 pandemic is just one example of the myriad challenges to mission completion that mission resilience seeks to address.

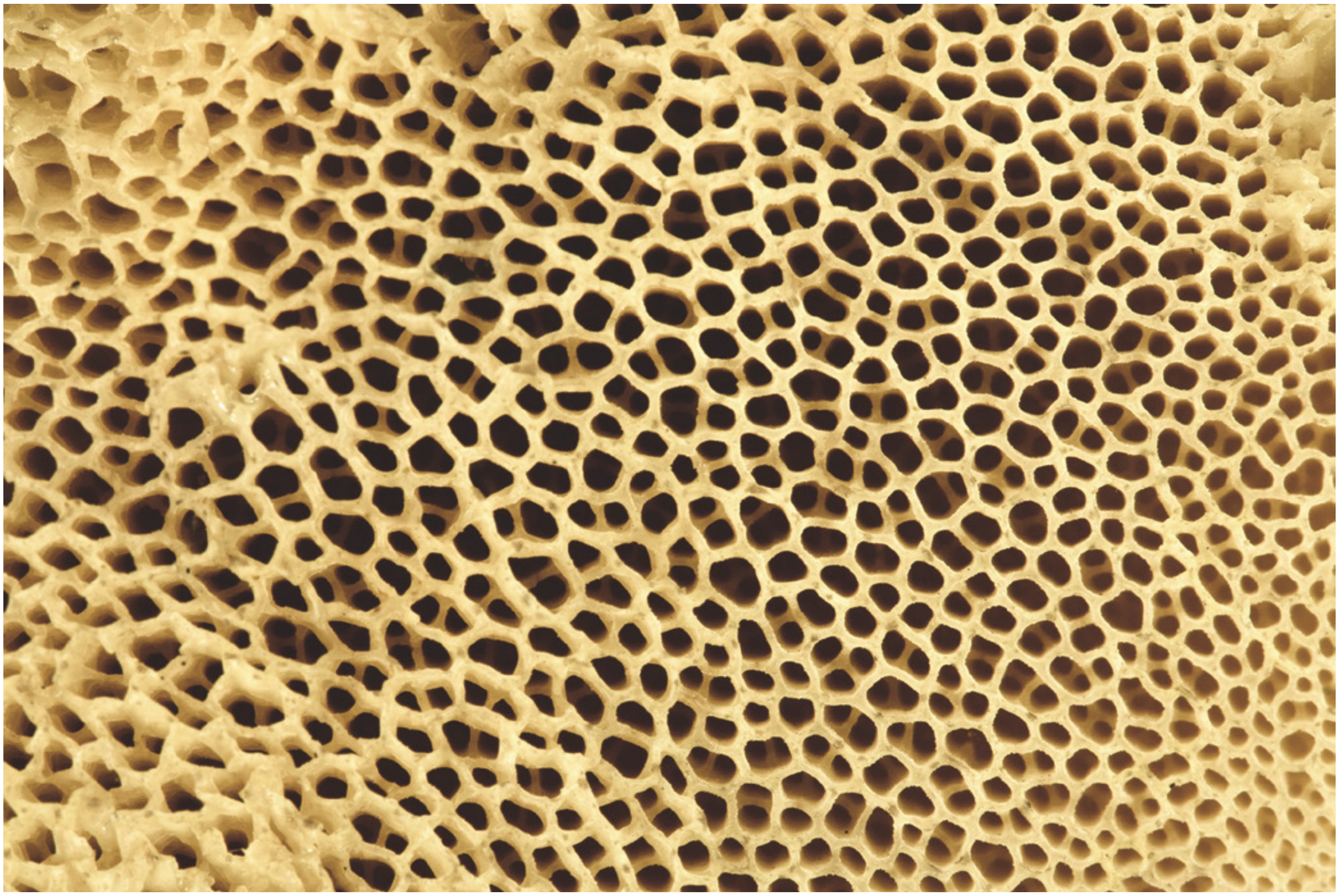
Government perspectives on resilience tend to reduce it to robustness and recovery—emphasizing the technical over the socio-technical and placing responsibility on designers to anticipate the future world, while devaluing the role of operators and maintainers. Improving on those narrow definitions provides a more complete picture of resilience. For example, a previous edition of the *Manual for the Operation of the Joint Capabilities Integration and Development System (JCIDS)* defined resilience as "the ability of the collection of systems to support the functions necessary for mission success in spite of hostile action or under adverse conditions."[19] It enumerated how systems and forces can be made more resilient: through a robust architecture, a networked system ensuring data availability during system compromise, and measures allowing "survival and operation." This focuses on "robustness," while

[16] David D. Woods, "Four concepts for resilience and the implications for the future of resilience engineering." *Reliability Engineering & System Safety* 141 (September 2015): 5-9, https://doi.org/10.1016/j.ress.2015.03.018.

[17] Aaron Boyd, "The Air Force's Platform One Team Thought It Was Agile. Then COVID-19 Hit." Nextgov, May 27, 2020, https://www.nextgov.com/emerging-tech/2020/05/air-forces-platform-one-team-thought-it-was-agile-then-covid-19-hit/165676/.

[18] David Vergun, "DOD Official Details COVID-19 Mitigation Efforts," U.S. Department of Defense, June 22, 2020, https://www.defense.gov/Explore/News/Article/Article/2228330/dod-official-details-covid-19-mitigation-efforts/.

[19] *Manual for the Operation of the Joint Capabilities Integration and Development System (JCIDS)*, December 18, 2015, http://www.acqnotes.com/wp-content/uploads/2014/09/Manual-for-the-Operationsof-the-Joint-Capabilities-Integration-and-Development-System-JCIDS-18-Dec-2015.pdf.



Just as human bones require stressors to prevent skeletal weakening, defense organizations require adequate strain lines in pursuit of mission resilience. *Source:* iStock/Ninell_Art

relegating responsiveness and adaptability to a single side note better understood as flexibility—the bandwidth to provide necessary functions "across a wider range of scenarios, conditions, and threats."

Similarly, a 2011 assessment of the DoD's efforts to engineer resilient systems focuses on resilience in terms of flexibility alone—the property of being "effective in a wide range of situations."[20] The presentation focuses on systems being adaptable, quickly iterated, and better informed. In this way, the DoD's definition best fits our pillar of responsiveness—the speed of change and the ability to incorporate crucial feedback into that change. But there is far more to resilience than that.

Academic versions of resilience tend to be more encompassing than those from government. They are most useful in orienting the definition of resilience, shaping its application, and stretching its usefulness in unconventional directions. Nassim Nicholas Taleb's idea of antifragility illustrates this well (though, for Taleb, the word "resilience" retains its traditional definition as a synonym for robustness). He envisions an axis with fragility at one end, robustness in the middle, and antifragility at the other—"breaks easily," "doesn't break," and "is better for having broken." He indulges the reader with a mythological analogy. Damocles is fragile: his well-being literally hangs on a string, a single-point dependency. A phoenix is robust: it returns to an ideal state reliably. A hydra, however, is antifragile. By regenerating two heads in the place of one, it is better off for having faced disruption—in this case, decapitation. The distinction between hydra and phoenix is a potent expression of the difference between robustness and adaptation. It highlights how resilience, as this paper defines it, is as much about benefitting from uncertainty as it is coping with it. This concept is especially relevant to

[20] Scott Lucero, *Engineered Resilient Systems - DoD Science and Technology Priority*, Office of the Deputy Assistant Secretary of Defense, October 5, 2017, https://sercuarc.org/wp-content/uploads/2018/08/Lucero_ERS_Panel-ASRR-2011-10-05_Final.pdf.

software development, where leaning into failure—failing first and fast—is a form of responsiveness, generating the critical feedback that an adaptive system can use to improve more quickly and accurately than would be possible without the experience of failure.

Netflix's development of chaos engineering[21] illustrates the continuous pursuit of resilience and the importance of real-world failure. Developed to improve the resilience of their applications, chaos engineering offers a controlled methodology to systematically break systems and better learn how to design and operate them. The practice involves unleashing the "chaos monkey" on systems—automated software that creates random disruption in systems, implemented to expose rigidity and fragility particularly in the face of unknown failures. Testing to failure and constructing live systems on which to experiment, rather than protected test cases and sacrificial lambs, sharpens the incentive to adapt both mission systems and supporting cloud infrastructure. Netflix's chaos engineering is the real-world equivalent of Taleb's antifragility—leaning into failure to maximize the benefits of feedback from it.

Failure is inevitable. Organizations need to develop a healthy relationship with failure in order to plan for it, handle it gracefully, rebound from it quickly, and take advantage of lessons learned to become stronger and more successful. Inside every software-intensive system are defects that may put the mission at risk. Some defects may be known but not fixed, while other defects remain hidden until conditions are ripe for failure. No organization is immune to failure. What separates top performers from laggards is their relationship to failure. Laggards fear failure, and that fear can slow their activities to a crawl. This myopic approach to risk creates consequences of inaction that may outweigh the failures they hope to avoid. The practices described below will help organizations embrace failure.

1.1 Chaos Engineering

Chaos engineering is the discipline of "experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production."[22] When Amazon Web Services' (AWS) US-EAST-1 datacenter experienced outages in 2015, Netflix, a customer of the cloud service provider, was able to build on lessons learned from its chaos engineering practices to redirect traffic to an unaffected region and avoid any major service interruptions.[23] By applying chaos engineering, DoD program offices can improve their understanding of how mission systems perform when disruptions occur and learn how to make the mission more resilient, both in terms of the technical implementation and the organizational processes.[24]

1.2 Limit Secrecy

Confidentiality is a necessity when dealing with advanced technologies in the national security context, but the DoD can do far more to orchestrate the interaction of classified programs and plan for loss of secrecy by limiting it to only where necessary. Loss of confidentiality is a type of failure that DoD program managers must accept and plan for in today's contested environment. Examples of things that can be lost are secrets, such as credentials and certificates; architecture, design, and implementation details; and products (e.g., plans). While the DoD invests heavily in ensuring the confidentiality of its products, determined adversaries repeatedly overcome the department's defenses. Thus, DoD programs must plan to work around the loss of confidentiality by reducing the number of secrets, the impact of their loss, and the ability to reprovision new ones quickly where appropriate. This includes more effectively orchestrating and managing classified compartments and their corresponding secrets and permissions, as well as limiting the scale of secrets infrastructure which can fail. As an example from industry, AWS' centralization of access keys limits "secrets sprawl"[25] and makes for a closely audited and controlled environment, so that even in the event of a breach logs can reveal who had access to what and when.[26]

Recommendations

1. **[DoD] Study Mission Resilience in Safety-Critical Systems:** Rapid and continuous development and the organizational incentives and processes to support it largely stem from the commercial and open-source software development worlds. Some of these practices may not be well-suited for engineering resilient mission systems whose failure or abnormal behavior could

[21] Fredric Paul, "Breaking to Learn: Chaos Engineering Explained," New Relic, January 10, 2019, https://blog.newrelic.com/engineering/chaos-engineering-explained.

[22] *Principles of Chaos Engineering*, last updated May, 2018, http://principlesofchaos.org/?lang=ENcontent.

[23] Nick Heath, "AWS outage: How Netflix weathered the storm by preparing for the worst," *TechRepublic*, September 21, 2015, https://www.techrepublic.com/article/aws-outage-how-netflix-weathered-the-storm-by-preparing-for-the-worst/.

[24] Casey Rosenthal and Nora Jones, *Chaos Engineering: System Resiliency in Practice* (California: O'Reilly Media, 2020), https://www.oreilly.com/library/view/chaos-engineering/9781492043850.

[25] Armon Dadgar, "What is 'secret sprawl' and why is it harmful?" HashiCorp, August 14, 2018, https://www.hashicorp.com/resources/what-is-secret-sprawl-why-is-it-harmful.

[26] Jim Scharf, "What to Do If You Inadvertently Expose an AWS Access Key," May 1, 2014, AWS Security Blog, https://aws.amazon.com/blogs/security/what-to-do-if-you-inadvertently-expose-an-aws-access-key.

cause immediate harm to human life. While commercial work in autonomous systems, like self-driving cars, offers valuable insight, the performance environment and tolerances of, for example, military aviation differ significantly. The DoD should charter the Defense Innovation Board in conjunction with appropriate outside partners to study the application of mission resilience practices and principles, including those under the rubric of Agile development, to safety-critical systems. This study should yield at least two outcomes: 1) specific criteria for what differentiates safety-critical systems, including categories of failure and experimentation tolerance and 2) an inventory of common mission resilience practices which are still suitable for systems in each of these categories.

2. **[DoD] Find the Chaos Monkey:** The DoD should seek to implement chaos engineering as a core resilience practice in the testing and evaluation phase of a handful of major mission systems to evaluate the associated benefits and challenges. Intentional disruption of production-ready systems is uncomfortable, but provides tremendous insight on these systems. Tangible gains in security and performance could help clear the path for further organizational reforms. This will require adapting organizational processes and management incentives to embrace purposeful failure as valuable experimentation. Data and signals of system performance and function are king, while compliance with strict timelines and "never-defeated-in-the-lab" testing regimes must fall.

2 – Improve Your Speed

Whereas Taleb provides understanding on the concept of antifragility, Erik Hollnagel provides valuable depth to the scope of resilience. He advises to look broadly and reflect on the power of feedback loops. In the context of engineering, he traces the development of the resilience concept from resilience in terms of a normal and disrupted state to an ability to adjust in response to adverse circumstances. In his argument, resilience is a continuous performance that goes beyond optimizing response to adversity. This includes the ability to capitalize on opportunity as well as on disruption. In the same vein, Hollnagel discusses how systems grow resilient as a function of responding to stimuli with acquired knowledge and informed predictions. He adds, however, that the most resilient systems have the ability to reflect on how the external world reacts to their own adaptation and how to incorporate that feedback cycle, in which a system influences the very cues it informs itself with, into future changes.

These two takeaways—that systems must look at how their resilience drives external change and that resilience also derives value from opportunity—augment the definition of resilience by expanding where it can apply. Widening the aperture of our system definition to include organizational processes like acquisition provides the opportunity for additional capacity to adapt. For example, as hardware reaches end-of-life, a predictable event often put outside the scope of a mission system, procedures for purchasing new devices add resilience by incorporating how a system's internal changes—here, aging—affect broader processes. Further, by looking beyond traditional scope, the system's resilience morphs from something still fundamentally responsive into an opportunistic trait. The chance to influence acquisition provides an opportunity to further increase resilience when seized on.

The speed at which an organization executes software-related processes—patching, resolving bugs and defects, reprovisioning compromised secrets, detecting and remediating, and engineering out classes of vulnerabilities—is a key indicator of the security and resilience of its mission systems.[27] The DoD must make improving speed a key outcome in its transformation toward more resilient mission systems. Improving speed is not about cutting corners to accomplish goals faster, but rather measuring and analyzing everything to understand bottlenecks, then aggressively resolving those bottlenecks to improve speed while still performing due diligence. Improving speed also relies on prioritization, understanding criticality sufficient to differentiate the essential from the important. Oftentimes, this involves finding ways to automate manual, labor-intensive, repetitive tasks so that personnel can focus their energy on more constructive activities. A means to implement this principle is captured in contemporary Agile methodologies, specifically the DevSecOps development philosophy. The following practices help organizations improve their speed.

2.1 Continuous Integration (and Testing)

For many complex applications, it is nearly impossible to predict the vast number of specifications and requirements prior to testing and deployment with real customers or users. Rather than designing, developing, and testing a large application all at once, iterative development is the practice of breaking the software development process into small, manageable cycles. In this way, iterative development can manage complexity and provide a constant feedback loop for defense organizations managing software-intensive systems. Through the broader adoption of development pipelines similar to the Air Force's Kessel Run,[28] the DoD should apply continuous authorities to operate (ATOs) to adopt more continuous and iterative software development practices.[29]

2.2 Continuous Delivery

Where continuous integration advocates developers regularly commit new code to an evolving build, integrating their work with others and testing to ensure the integrity of the resulting program, continuous delivery moves one step further to regularly deliver that program to its end user. Continuous delivery frames code commits and project updates as a push (rather than pull) to the user in an iterative fashion to expose new features and functionality to user feedback as quickly as feasible. This is at odds with conventional milestone-driven acquisition models where products are delivered infrequently and subject to testing and review. Continuous delivery builds in this

[27] Kelly Bissell, Ryan M. Lasalle, and Paolo Dal Cin, *Innovate For Cyber Resilience Lessons from Leaders to Master Cybersecurity Execution*, Accenture Security, 2020, https://www.accenture.com/_acnmedia/PDF-116/Accenture-Cybersecurity-Report-2020.pdf#zoom=40; Molly Crowther, "Faster is Safer: Security in the Enterprise," presented at Velocity Conference, September 30-October 3, 2018, New York, video, 1:00, https://www.oreilly.com/library/view/velocity-conference-/9781492025870/video323218.html.

[28] Kessel Run, "About Us."

[29] Robert Skertic, "Continuous ATO," Defense Acquisition University, August 16, 2019, video, 1:36, https://media.dau.edu/media/Continuous+ATO/1_10jrntl6.

The US Air Force's Kessel Run was designed as a startup software lab within the Defense Innovation Unit, created with the recognition that the DoD's approach to software development is "broken" and threats are outpacing the lengthy acquisition process.[1] Prioritizing continuous delivery, feedback, and learning, Kessel Run's objective is to create software that can be deployed across domains, anytime, anywhere.[2] The resulting agility has allowed the organization to develop software, from idea to the field, in as few as four months—a whopping twenty-four times faster than it takes the rest of the Air Force. It is abundantly clear that a scaled implementation of a Kessel Run-style approach to software development across the services has the potential to deliver more resilience to software-intensive mission systems than ever before. By employing military personnel, the model has filled technical capability gaps that are too often overlooked by contractors competing for large projects.[3] For instance, thanks to Kessel Run, air refueling missions once planned manually on white boards and prone to error and inefficiency are now planned with a special application. By addressing "small" issues ranging from mission planning to reporting systems, the program has saved the Air Force 1,100 man-hours and $13 million every month.[4]

As this success has led to additional investment and growth from twenty-five to one thousand two hundred employees, a burgeoning internal bureaucracy and increasing technical complexity has led to diminishing returns on investment. Rapid growth has led to employee frustrations over hiring, technology development, culture, and programming practices. Scaling up has also revealed limitations in managing the deployment (versus development) of large-scale systems iterated in this model. These frustrations, coupled with concerns over people, process, and technology, make it clear that while the organization is an improvement over the rest of the service, maintaining continuous improvement and rigorous feedback loops with users is not easy to scale.[5]

[1] J. Michael McQuade et al., *Software Is Never Done: Refactoring the Acquisition Code for Competitive Advantage*, Defense Innovation Board, May 3, 2019, https://media.defense.gov/2019/Apr/30/2002124828/-1/-1/0/SOFTWAREISNEVERDONE_REFACTORINGTHEACQUISITIONCODEFORCOMPETITIVEADVANTAGE_FINAL.SWAP.REPORT.PDF.

[2] Kessel Run, "About Us," US Air Force, accessed July 14, 2020, https://kesselrun.af.mil/about.

[3] Frank Konkel, "Pentagon Says It Needs 'More Time' Fixing JEDI Contract," Nextgov, June 19, 2020, https://www.nextgov.com/cio-briefing/2020/06/pentagon-says-it-needs-more-time-fixing-jedi-contract/166292/.

[4] Jim Perkins and James Long, "Software Wins Modern Wars: What the Air Force Learned from Doing the Kessel Run," Modern War Institute, January 17, 2020, https://mwi.usma.edu/software-wins-modern-wars-air-force-learned-kessel-run/.

[5] Tech and Innovation in Government, "Harvard Students Partner with the U.S. Air Force to Help Kessel Run Grow," April 8, 2020, https://innovategovernment.org/usaf-2020-blog/2020/3/30/harvard-students-partner-with-the-us-air-force-to-help-kessel-run-grow.

process of user review and feedback as an ongoing loop to inform the development process. The DoD's default acquisition model is structurally biased toward outdated software development life cycles that involve periodic milestones with code freezes—typically followed by integration, testing, and hardening stages. With continuous delivery, these events occur on a frequent basis, even as often as code is checked in to a build or repository.[30] In its *Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage* study, the Defense Innovation Board identifies helpful metrics to monitor progress toward continuous delivery.[31] For smaller programs, the recent proliferation of integrated DevOps pipelines (e.g., GitLab) can potentially provide a more cost-effective manner to pursue continuous delivery.[32]

2.3 DevSecOps

Continuous integration and delivery tighten the link between developer and user, making changes smaller and more frequent in response to feedback and design goals. These fall under the aegis of the DevOps or development and operations approach which seeks to break down barriers between the development, deployment, and operation of code. DevSecOps is a development philosophy aimed to layer security into that same tightened link between developer and user. Rather than an "insert security here" milestone on a development pathway, DevSecOps approaches security as a similarly continual process—encouraging development within the best practices of secure development life cycles, testing the security of submitted code, dynamically assessing the security of running builds and software

[30] Continuous Delivery, "Introduction," accessed July 15, 2020, https://continuousdelivery.com.

[31] J. Michael McQuade et al., "Defense Innovation Board Metrics for Software Development" in *Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage* (Washington, DC: Defense Innovation Board), May 3, 2019, https://media.defense.gov/2019/May/02/2002127284/-1/-1/0/DEFENSEINNOVATIONBOARDMETRICSFORSOFTWAREDEVELOPMENT.PDF.

[32] GitLab, "Simplify your workflow with GitLab," accessed July 15, 2020, https://about.gitlab.com/stages-devops-lifecycle.

units, and integrating adversarial testing of the entire development pipeline. The goal of DevSecOps is to merge development, security, and operations (in the DoD, traditionally system administration) processes to speed up capability development and delivery and improve the responsiveness of system adaptation. The DoD should take note of private sector efforts, such as Google's Site Reliability Engineering effort intended to involve operations personnel from the beginning of software development to overcome its arduous and lengthy feedback process. The General Services Administration has a helpful DevSecOps guide, including a three-stage maturity model against which organizations can assess their own and vendors' progress.[33]

Recommendations

3. **[White House] Streamline Classification Rules:** The complexity of the current classification system is one of the most structural drivers of complexity in DoD-associated national security systems, including mission system development and program offices. Numerous efforts have been made since the widespread adoption of IT and the Internet to revise how the US national security and intelligence communities designate, handle, store, and declassify sensitive information. The resulting need for special purpose systems and analog markings on otherwise digital-native information lead to costly delays, wastefully specialized and siloed systems, and inefficiencies in information flow and operational collaboration. The Executive Office of the President (EOP) should issue a revision and update to Executive Order 13526, integrating those recommendations of the 1994 Joint Security Commission report, *Redefining Security*, which have not yet been implemented, including moving to a two-tier classification system and common risk determination for inclusion in special access programs or compartments.[34] These changes should be supplemented with the recommendations of the Office of the Director of National Intelligence's (ODNI) 2008 review, *Intelligence Community Classification Guidance Findings and Recommendations Report*, particularly with respect to classification justification, duration, and marking. As noted in the 2008 report, "classification/dissemination/disclosure problems continue … All recent studies acknowledge the need for change in the information-sharing system."[35] A near-term update may be to permit more frequent and frictionless machine-to-machine interaction at common classification levels, with a longer-term effort to simplify the design and application of such rules across the board.

4. **[Congress] Enable the DoD to Buy Services and Capabilities, Not Programs:** Congress should formalize the DoD's pilot Software Acquisition Pathway in the FY22 National Defense Authorization Act (NDAA) with several additional enhancements.[36] The DoD should be empowered to 1) enable micro contracts within larger vehicles, e.g., pay for performance and delivery of working product every two to four-week sprint. 2) Drive embedded resilience performance metrics in contracts and tie evaluation of performance against these metrics, including a) organizational process adherence—DevOps/DevSecOps, b) security—measures of code integrity and patch performance (e.g., vulnerability severity and count), and c) resilience—measures of complexity (against dependency graph), submission of appropriate machine-readable documentation. Congress can build off of its FY21 proposed extension of the pilot program, originally passed in the FY16 NDAA, to streamline auditing processes for innovative technology projects carried out by small businesses.[37] Using the NDAA as a vehicle will give concrete authorities to accelerate awards to nontraditional defense contractors and provide organizational leaders with the tools necessary to change their acquisition culture. The core objective is to drive flexibility in how capabilities are developed and maintained, decoupling a system from a particular vendor or proprietary development pipeline, much as software virtualization decouples software and hardware.

5. **[DoD] You Don't Have to Pick One Software Factory:** The DoD should allow program offices to subdivide design and development work amongst existing DoD Enterprise software factories and rapid acquisition channels, e.g. the Rapid Capabilities Office or Defense Innovation Unit. As the DoD continues its evolution toward Agile development and program management, it must reward success in multiple environments and avoid locking programs into highly adaptive but difficult-to-scale chokepoints. Programs like Kessel Run may specialize in developing particular types or scale of services, perhaps optimizing to the hardware in the

[33] GSA Tech at GSA, *GSA Tech Guides*, accessed August 20, 2020, https://tech.gsa.gov/guides/dev_sec_ops_guide/.

[34] Jeffrey H. Smith, *Redefining Security: A Report to the Secretary of Defense and the Director of Central Intelligence* (Washington, DC: Joint Security Commission, February 24, 1994).

[35] Director of National Intelligence and Chief Information Officer, Intelligence Community Technology Governance, I*ntelligence Community Classification Guidance Findings and Recommendations Report*, Office of the Director of National Intelligence (January 2008): 9. https://fas.org/sgp/othergov/intel/class.pdf.

[36] US Under Secretary of Defense, Department of Defense, "Software Acquisition Pathway Interim Policy and Procedures," Acquisition and Sustainment, memorandum, January 3, 2020, https://www.acq.osd.mil/ae/assets/docs/USA002825-19%20Signed%20Memo%20(Software).pdf.

[37] National Defense Authorization Act for Fiscal Year 2021, S. Rept. 116-236, 116th Congress (2019-2020), https://www.congress.gov/congressional-report/116th-congress/senate-report/236/1.

loop processes or the needs of a class of compartmented programs. The DoD should embrace this diversity and architect policies and technology platforms to support in line. There are natural growing pains in service quality as the DoD Enterprise DevSecOps Initiative (DSOP) and related efforts like PlatformOne build out and real limits in the medium term as service organizations work to match commercial offerings from Microsoft, Amazon, and others. Diversity of options and flexibility is good, especially as a means to drive growth in the user base. Allowing organizations within the DoD to compete to match commercial providers for quality and extensibility is better. Tolerating (and, indeed, rewarding) success in multiple competing organizations within the DoD can facilitate more rapid innovation and maturation of these services.

3 – Always Be Learning

A resilient mission system is designed with the capacity to adapt in mind. This kind of organizational design starts with program offices, program managers, and program executive officers. These groups can work today to better understand their organization, their people, and their technology sufficient to change in the face of anticipated and unanticipated disruptions. The systems these organizations and individuals manage should do more than merely survive and persevere through disturbance—a resilient system gains a comparative advantage in the wake of disruption.

Mission resilience can be understood as the ability of a mission system to continue to operate through disruptions by demonstrating robustness, responsiveness, and adaptability. Disruptions are products of a complex environment, and a system's ability to achieve resilience is a function of its own ability to match the complexity of the outside world to the complexity of its many inner states to achieve its mission objectives. Cloud computing serves as an example of such complexity management. Cloud computing is a set of massively scaled, distributed systems: racks upon racks of servers wired together and subject to overlapping organizational incentives and requirements, all the subject of careful design and oversight choices made, ultimately, by people. Those individual decisions about technology and organizational process create much of the friction associated with large, complex systems, and cloud computing reflects people's centrality in shaping the other two pillars of a system. The mission objective is to provide continuity of service and graceful degradation in the face of disruption, and failure is the collapse of massive swathes of the Internet.

In other literature, resilience is often defined in terms of opposition and reaction. It is discussed as the resistance to

Data center room interior. *Source:* iStock/lovegul external stimuli and the return to equilibrium in their wake. This emphasis on entrenchment and response obscures the importance of the proactive measures undertaken *continuously* and *prior* to catastrophe. Resilience is not static, nor is it reactive. In other words, resilience is *not* just the ability to survive impact and repair the damage—it is a set of principles around monitoring, learning about, and adapting systems to known failure modes, while simultaneously building the capacity to respond to unknown adversity and adapt in the long term. Success is not a product of short-term sprints or overnight changes, and mission success can never be fully guaranteed. Mission resilience can be improved over time through actions taken to enable robustness, responsiveness, and adaptability.

Industry perspectives begin to translate resilience from a broad abstraction to a concrete property of real-world systems by describing how to make something resilient and how to recognize resilience. For instance, Amazon's use of formal methods demonstrates broad application of resilience principles and some of the hurdles to achieving them. To be able to precisely describe system design and architecture, AWS adopted the TLA+ language for formal specification. Broadly, the framework helped shift engineering processes away from asking a one-two combination of questions—"what do we want to happen" and "what might go wrong"—and toward a single inquiry: "what needs to go right?" This drives the continuous integration/delivery model deeply into the organizational process. The new emphasis focused on dependencies, framing development in terms of robustness, requirements, and capabilities all grounded in a precise understanding of what a system is and can do. Engineers had to focus on procedurally verifying the safety of making system changes. This forced them to hone in on responsiveness and adaptive capacity through a deep understanding of the system itself and anticipation of cascading failures. According to Amazon engineers, TLA+ "added significant value, either finding subtle bugs that we are sure we would not have found by other means, or giving us enough understanding and confidence to make aggressive performance optimizations without sacrificing correctness." [38]

Amazon's adoption and use of TLA+ also showcases potential obstacles to fostering resilience. One anecdote recounts how the TLA+ development team pitched the language to AWS engineers. Recognizing that their software engineers were reticent to adopt a formal methods framework, the presenters pitching TLA+ dubbed it "exhaustively testable pseudo-code," described the idea as "Debugging Designs," and were able to carry out a proof-of-concept application. Engineers working on Amazon's Simple Storage Service, which replicates data between AWS cloud regions, used the system to design and debug a fault tolerance algorithm, uncovering subtle bugs that would have gone unnoticed in the process. Following the tangible success, engineers on other projects began adopting the practice independently and discovering bugs in their design phases, prior to implementation. Frequently, TLA+ also revealed bugs in patches for design flaws, and it was consistently able to detect issues even between engineers with different design and coding styles. Getting the personnel component of a system to adapt was instrumental in achieving resilience, vividly depicting a key takeaway: people are the key adaptive agent in mission systems. If they cannot become agile and receptive to change, a system cannot be resilient.

Google's approach tracks this focus on "bend but don't break." In the Heather Adkins-led *Building Secure & Reliable Systems*, resilience is defined strictly as "the ability to protect against attacks and to withstand unusual circumstances that stress [a] system and affect its reliability."[39] Resilient systems tend to keep performing throughout disruption, albeit in a degraded capacity, according to the book. At first, this interpretation appears more limited than other industry standards by focusing on a system's ability to withstand atypical change. As written, its definition is closest to what has previously been called "robustness." However, there is value in more nuanced principles —rapid and informed recovery, easily understandable and analyzable systems, assuming an ever-changing environment, and focusing on personnel and organizational culture as agents of change. Ultimately, many of Adkins' tenets are consistent with our mission resilience framing, just organized under a different vocabulary.

Crucially, resilience is not simply a state of constant change. As Richard Cook and David Woods note, it is being continually poised to adapt—the capacity more than the action, though the latter provides an easier way to measure and quantify the former.

Adversaries evolve, and systems on which the DoD relies are becoming larger, more interconnected, more complex, and harder to understand. As a result, the bar for maintaining advantage over the adversary and achieving mission success is always rising, propelled by both shifting

[38] Chris Newcombe et al. "Use of Formal Methods at Amazon Web Services," Amazon.com, September 26, 2014, https://lamport.azurewebsites.net/tla/formal-methods-amazon.pdf.

[39] Heather Adkins et al., *Building Secure and Reliable Systems: Best Practices for Designing, Implementing and Maintaining Systems* (California: O'Reilly Media, 2020), https://static.googleusercontent.com/media/landing.google.com/en//sre/static/pdf/Building_Secure_and_Reliable_Systems.pdf.

adversary tactics and increasingly complex mission systems. The only way to keep pace is to enable continuous improvement through adaptation, of which learning is a crucial component.[40] The DoD must take advantage of every outcome, be it failure or success, to learn more about itself and its adversaries. The following practices help organizations to always be learning.

3.1 Hypothesis-Driven Development and Experimentation

Hypothesis-driven development (HDD) improves organizational learning by applying the scientific method to building systems to learn how to develop the most useful capability. In the context of system design, HDD involves hypothesizing about the results of a modification to the system; conducting an experiment involving measurement, implementation, and assessment; and learning from the result. In Dave Farley's "Taking Back 'Software Engineering'" lecture, he explains that the scientific method has led to some of the greatest achievements in history—such as NASA landing humans on the moon—and through HDD, organizations can apply it to software development.[41] By applying concepts from the HDD approach, the DoD can improve its ability to produce resilient systems by continuously learning through a data and experiment-driven methodology.

3.2 Observability

Learning from failure in service of improving a mission system depends on capturing as much information as possible on how and why a system failed. This starts with the ability to make correct measurements and interpret them into action. Focused on metrics, tracing, and logs, observability is about having the right instrumentation in place to answer questions about how the internals of computing systems are working. When done well, observability will help developers understand baseline mission system performance, support problem diagnostics, and provide leading indicators for when failure is about to happen. The DoD must modernize its approach to monitoring, as today's systems are increasingly complex and failure modes are multiplying. To do so, the DoD could look to Google's Site Reliability Engineering (SRE) teams' monitoring practices to analyze long-term trends, compare over time, be alert to failure, and conduct retrospective analysis.[42]

3.3 Operational Metrics

Operational metrics enable organizations to measure aspects of resilience across an organization. Distinct from observability, metrics drive measurement toward change, in particular, the measurement of an organization's processes, people, and technology relative to resilience goals. Measure what you want to achieve. Examples of operational metrics include:

- Mean time to patch/update
- Mean time to resolve (for faults, defects, security incidents, bugs, patch release)
- Mean time to deploy
- Number of security events and incidents
- Mission utilization and capacity remaining

Aggregate measures, like averages, can hide atypical patterns that could yield significant insight, so it is valuable for organizations to match these with other descriptive statistics. By capturing operational metrics about the software life cycle and operational processes, DoD program managers can better estimate the risk and resilience of their operational mission systems and how that state is trending. A survey on key performance indicators in cybersecurity by the Digital Guardian's DataInsider blog demonstrates the importance of operational resilience metrics and is helpful in determining what metrics to measure.[43]

Recommendations

Mission systems are subject to exquisite knowledge derived from their use in the field. Capturing that knowledge and working to channel it into the larger development and deployment cycle is the core focus of this set of recommendations. This is true all the way from deployed units in active operations to vendors within the Defense Industrial Base. The DoD will be best positioned to maintain a continuous feedback loop from the field where it has already worked to lower barriers to information sharing between servicemembers and back to mission system program offices. The recommendations here for the DoD focused on the unit level are aimed at one thing: adaptation. These recommendations are intended to position the DoD to leverage a more flexible and tightly integrated development process for new systems together with creative adaptation of legacy systems and capabilities.

[40] Daniel Levinthal and James G. March, "A model of adaptive organizational search," *Journal of Economic Behavior & Organization* 2 (1981): 307–333, https://doi.org/10.1016/0167-2681(81)90012-3.

[41] Dave Farley, "GOTO 2020 • Taking Back 'Software Engineering,'" May 20, 2020, video, 44:51, https://www.youtube.com/watch?v=_N_jlrEBOpw; Launch Darkly Blog, "Hypothesis Driven Development for Software Engineers," March 16, 2018, https://launchdarkly.com/blog/hypothesis-driven-development-for-software-engineers.

[42] Rob Ewaschuk, "Monitoring Distributed Systems" in *Site Reliability Engineering: How Google Runs Production Systems*, ed., Betsy Beyer (California: O'Reilly Media), accessed July 14, 2020, https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems.

[43] Ellen Zhang, "Security and Analytics Experts Share the Most Important Cybersecurity Metrics and KPIS," DataInsider, last updated December 8, 2017, https://digitalguardian.com/blog/what-are-the-most-important-cybersecurity-metrics-kpis.

6. **[Congress] Create the Center of Excellence for Mission Resilience (CEMR) in the DoD:** Congress should create an office dedicated to driving resilience across mission systems and the DoD under the under secretary of defense for acquisition and sustainment. The role of the office is growing and maintaining expertise in mission resilience to share with and support DoD program managers. CEMR is modeled on the US government's digital services support and development agency, 18F, albeit with a focus on resilience and a somewhat more expeditionary mindset, aiming to support programs in situ. The office should embed eighty to one hundred staff within major program offices, bringing dedicated expertise on mission resilience practices and organizational reform. CEMR should also be empowered to designate high-risk program offices and associated mission systems based on performance shortfalls, security threats, or opportunities to pilot mission resilience approaches. These programs would receive additional funds for use on mission resilience reforms, at a fixed percentage of the total target program budget. CEMR should evaluate and update this high-risk list annually, working in conjunction with more narrowly tailored programs like the Air Force's Cyber Resiliency Office for Weapons Systems (CROWS).[44] The goal of CEMR is the rather difficult task of institutionalizing revolution within the DoD, driving change in deeply rooted acquisitions and systems deployment policies. Features like ability to direct budget dollars and a Senate-confirmed position are intended to strengthen that mission.

7. **[DoD] Resilience Rotation:** The DoD should create a nine-month exchange program for technically inclined junior and mid-grade officers and NCOs to "exchange" with staff at the CEMR and with vendors behind their unit's mission systems and associated technology stack. While today there are programs to pull servicemembers into technology vendors, these tend to be short-term rotations (two-eight weeks) or focused more on career transitions. Understanding a unit's mission systems and technology base will better enable these servicemembers to adapt these systems to emergent needs in the field. Providing the services with a slowly expanding pool of tactical and operational talent with social capital across the vendor ecosystem and keen understanding of the technology they deploy with will provide long-term benefits to the officer and NCO pool as well as short-term gains in familiarization and expertise. There is a danger that servicemembers on these rotations may be more likely to leave for industry after their term of service. Managed properly, this phenomenon could create a fabric of veterans in these parts of the private sector who have a clear understanding of operational needs and experience with the DoD's bureaucracy and culture.

8. **[DoD] Measure at Machine Speed:** The DoD's chief information officer (CIO), in conjunction with Acquisition & Sustainment (A&S) and leadership of the DoD Enterprise DevSecOps Services Initiative (DSOP), should develop specific metrics for speed of software development-related processes, including deployment and feedback loops. These metrics should provide program offices more granular and effective means to assess program performance. Metrics should include values like failed deployments, availability, mean time to detect, mean time to deploy, change volume, and automated test pass rates.

9. **[DoD] Tie Users and Developers Together:** DSOP leadership should work with the DoD's CIO and other offices, as appropriate, to lower the barrier to entry for access to the DSOP Mattermost platform, maximizing representation from across the active development community. This platform should also be expanded to serve as a joint-service resource for sharing insights on integrating and employing software-intensive mission systems, modeled on the Army's successful CompanyCommand and PlatoonLeader websites.[45] Army officers used these platforms to exchange information, including some detailed discussion of counter-insurgency tactics. This revised platform, maintained by DSOP, should be the focal point of departmental efforts to implement a core principle of Agile methodologies by bringing user and developer communities tightly together. Deployed servicemembers should be able to share experiences adapting current technologies and mission systems, desired future capabilities and "use-cases," and to flag bottlenecks in design or acquisitions pipelines. The platform would be a resource for units as well as a repository of potential use cases for the novel employment of current systems.

10. **[DoD] BattleLab Initiative:** The DoD should create a permanent organization housed under the Joint Staff and resourced from the Office of the Secretary of Defense (OSD) to collect approximately one hundred and twenty field and staff-grade officers from across the services to complement the SoSITE program managed

[44] Kris Osborn, "New Air Force unit aims to protect weapon systems from cyber attacks," Defense Systems, February 17, 2017, https://defensesystems.com/articles/2017/02/17/crows.aspx.

[45] Dr. Nancy M. Dixon, "CompanyCommand: A Professional Community That Works," Appel Knowledge Services, NASA, July 1, 2007, https://appel.nasa.gov/2007/07/01/companycommand-a-professional-community-that-works/.

by the Defense Advanced Research Projects Agency (DARPA). This BattleLab Initiative should be categorized as a one-year Professional Military Education (PME) opportunity to replace a standard service education opportunity for each officer selected. The BattleLab would be established with a limited staff and largely act as a platform for self-organized activity by each year's cohort to identify new opportunities to employ existing mission systems. Participants would be asked to suspend rank, as in a traditional academic setting, and work with DARPA and BattleLab staff to run wargames, simulations, and limited field trials in conjunction with their parent units as feasible. The DoD and the services should look to this initiative as a source of doctrinal innovation and a means of maximizing the service life and value of legacy mission systems. Relative to the Army Futures Command, BattleLab would be focused on repurposing and better utilizing existing platforms and technology in the US arsenal and provide a natively joint, rather than service-specific, environment.

4 – Manage Trade-Offs and Complexity

Mission resilience is not purchased up front—it must be consciously maintained throughout the entirety of a system's life cycle and is difficult to bolt on retroactively. It is achieved through continuous effort, adjustment, and improvement, and it requires the involvement of all system stakeholders. Resilience requires learning about, purposefully designing, monitoring, and intervening in systems to prevent cascading failures, single points of failure, and information scarcity. While the analogies above focus on systems as hardware, the reality is much more complex as the DoD must account for people, organizational processes, and technologies.

David Woods enumerates four pillars of resilience, adding subtlety to our pillar of adaptation. Two of his pillars clearly correspond: "rebound" is similar to responsiveness, and "robustness" is a perfect match. What Woods calls "sustained adaptability" is a version of adaptation that specifically highlights the importance of the scalability of adaptive capacity—that adaptation in a system should not disrupt dependencies and should account for the larger organizational context in which they occur. Woods discusses resilience as "graceful extensibility," where he considers how a system behaves when its capacity to adapt is stretched to the limit. The ability of a system to continue to adapt and demonstrate resilience even as it operates under abnormal demands is critical. More simply, there must be a robustness in a system's ability to adapt. It needs to be able to change during extreme adversity, not just during normalcy or even anticipated disruption, and it must be able to fail gradually instead of spectacularly and suddenly. Graceful extensibility is a subtle but crucial dimension of adaptation.

Some government perspectives are more abstract, in other cases timelines are widly different. For example, NIST SP 800-160 is more inclusive, describing resilience in terms of anticipation, withstanding, recovery, and adaptation and laying out broad principles and practices for achieving cyber resilience—accounting for a whole-of-system perspective across the full life cycle.[46] However, that focus on anticipation may limit -160's utility. One should anticipate disruption where possible, while integrating that awareness with all pillars and against static response models, even preemptive ones. However, focusing only on what can be anticipated leaves a system vulnerable to what cannot be foreseen. This means that even exceptional efforts to recover may still fail or produce poor system performance if the environment in which that system exists has changed. Only organizations that develop the capacity to change are prepared for unknowable adversity. Tellingly, NIST's report, *Developing Cyber Resilient Systems: A Systems Security Engineering Approach*, focuses on returning to a status quo that in truth may no longer exist—even its vision of adaptation is aspirational, focusing on modifications to adjust to "predicted changes." Accordingly, a holistic view of adaptation encompasses the ability to make changes that can apply to a shifting set of objectives, a greater scope than NIST's focus on preserving "mission or business functions." In short, the capacity for holistic change and the ability to adjust to unpredictable disruption are critical components of resilience not sufficiently dealt with in the NIST report.

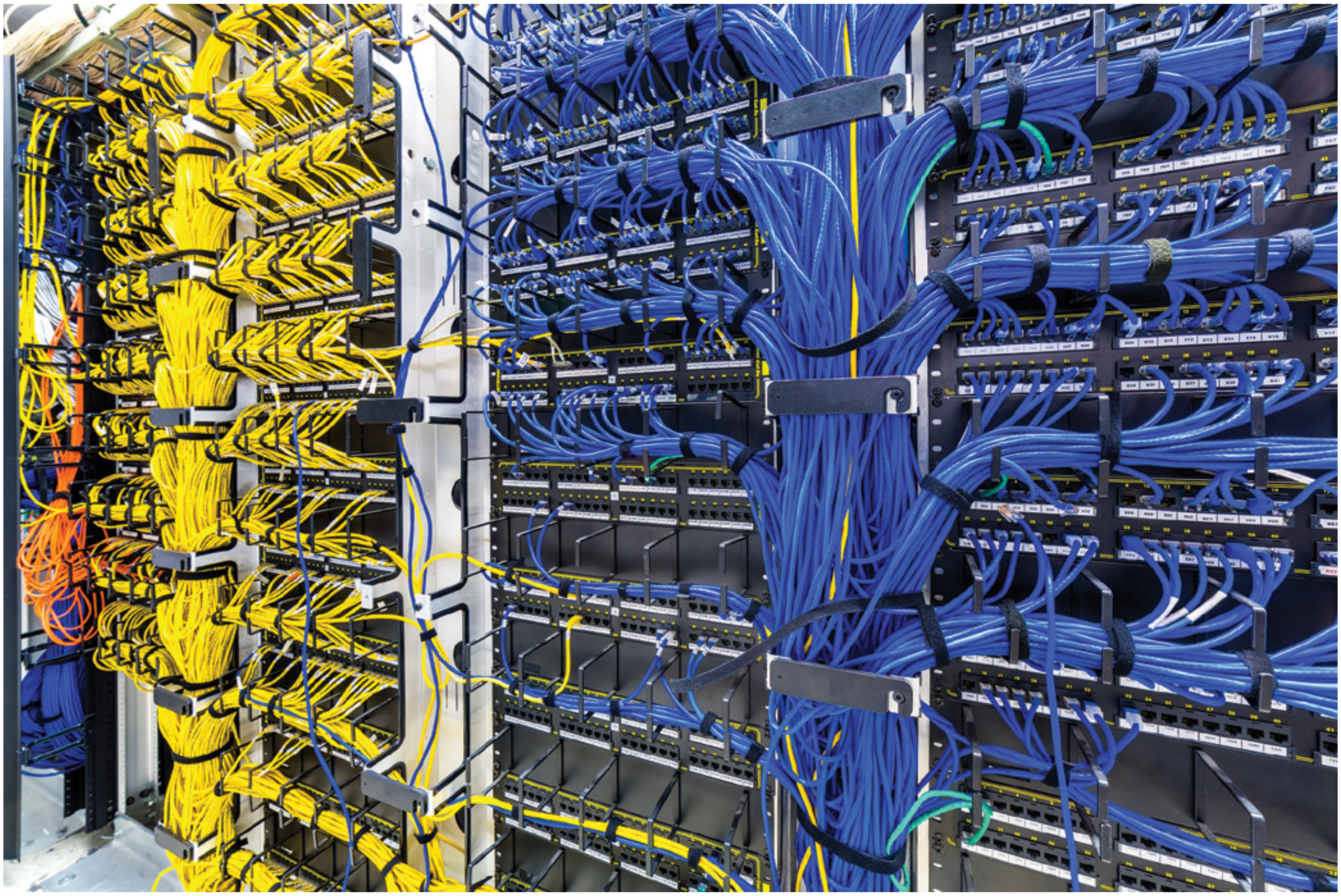
Successful organizations tend to grow over time, and their complexity inevitably increases as a result.[47] They must learn how to cope with seemingly unbounded complexity in the face of growing demand for new features, changing operational contexts, increasing interconnectedness, and evolving adversaries. Making trade-offs is critical to reigning in complexity and effectively managing projects. Traditional project management models, such as the management triangle or waterfall, fix either scope, schedule, or cost against the variable components. In most programs that are developing mission systems, security and resilience are constantly traded off for scope, schedule, or cost causing all three aspects to suffer. The DoD must improve program managers' understanding of what trade-offs are being made and their impacts down the line.

Not all complexity is bad though, and some will prove necessary to adapt to the changing needs and environments. What is the optimal amount? A major concern with complex systems is that they exhibit emergent behaviors that cannot be predicted[48]—thus, problematic behaviors of mission systems cannot be reasoned about until they are observed in operations. To this end, many efforts to address this concern focused on improving our understanding of complex systems through development

[46] Ron Ross et al., *Developing Cyber Resilient Systems: A Systems Security Engineering Approach*, Special Publication 800-160 Vol. 2, National Institute of Standards and Technology, November, 2019, https://csrc.nist.gov/publications/detail/sp/800-160/vol-2/final.

[47] Larry E. Greiner, "Evolution and Revolution as Organizations Grow," *Harvard Business Review* (May-June, 1998), https://hbr.org/1998/05/evolution-and-revolution-as-organizations-grow.

[48] James P. Crutchfield, "The calculi of emergence: computation, dynamics and induction," *Physica D: Nonlinear Phenomena*, 75 (1994):11–54.



Rack with generic Ethernet cat5e cables, part of a large data center. *Source*: iStock/Ultima_Gaina

of more sophisticated and encompassing system models.[49] The academic and management communities have also looked at how to manage or engineer emergence through better design processes[50] as well as organizational structure and adaptation.[51]

Despite great efforts and multiple promising directions, the community has yet to come to a set of tried-and-true practices that can help defense organizations better manage complexity of their missions and mission systems. To manage complexity, the DoD should start by recognizing the need for trade-offs, improving program managers' understanding of what trade-offs are being made, and acknowledging the impact these trade-offs will cause down the line. The DoD should support further interdisciplinary research and innovation to make a breakthrough in this area. The following practices help organizations to manage trade-offs and complexity.

4.1 Primary, Alternate, Contingency, Emergency (PACE) Plans

Originally designed to express an order of precedence for tactical communication, PACE plans—short for primary, alternate, contingency, emergency—outline a methodology for building in redundancies independent of one another

[49] Ales Kubik, "Toward a Formalization of Emergence," *Artificial Life* (February 2003), 9 (1): 41-65; Jeffrey Goldstein, "Emergence as a Construct: History and Issues," *Emergence* 1 (1999): 49-72; Claus Emmeche, Simo Køppe, and Frederik Stjernfelt, "Explaining Emergence: Towards an Ontology of Levels," *Journal for General Philosophy of Science* 28 (1997): 83-119; J. Deguet et al., "Elements About the Emergence Issue: A Survey of Emergence Definitions," ComPlexUs, 3 (1): 24-31, ISSN 1424-8492, 2006.

[50] E. Fricke and A.P. Schulz, "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle," *Systems Engineering*, 4 (2005); R. De Neufville, "Uncertainty Management for Engineering Systems Planning and Design," MIT Engineering Systems Symposium, 2004; D.E. Hastings et al., "Incorporating Uncertainty into Conceptual Design of Space System Architectures," INCOSE International Symposium, 13 (2003): 1,380-1,392.

[51] Olaf Bach, "How to Manage Complexity Through Organizational Structure," Management Kits, November 6, 2019, https://www.managementkits.com/blog/2019/11/6/manage-complexity-through-organizational-structure.

to account for a range of eventualities.[52] The practice has been adopted outside of tactical communications to rank most ideal to last-resort options—an infantryman, for example, may carry a rifle, pistol, grenade, and knife. PACE provides a conceptual framework for developing lines of logic independent of third-party software, allowing a system to operate most of the way in a degraded environment. As explained in Daniel E. Geer, Jr.'s *A Rubicon*, too much interdependency can be more harmful than beneficial as reuse of software already familiar to attackers can cascade into the failure of an entire system.[53]

4.2 Service-Level Objectives

In today's environment where requirements are written years before software is developed, performance and resilience goals are difficult to reconcile and often change over time. Service-level objectives (SLOs) provide a way to determine which behaviors of a mission system matter most and how to measure and evaluate those behaviors in order to manage the trade-off of performance and speed or scope. A target value or range of values measured by carefully defined quantitative measures, an SLO may look like: *Requests Latency SLO: 99% of service requests (measured over 1 hour) will complete in less than 100 ms*. Google's SREs provide a practical framework for DoD programs to choose appropriate metrics that they can reference to improve how they coordinate and manage performance and resilience goals.[54]

4.3 Tabletop Exercises

Tabletop exercises (TTXs) in cyberspace and the physical world[55] can help organizations identify and understand different risk scenarios and prepare for threats and disruptions by bringing together stakeholders to assess an organization's strategic, procedural, and technical capabilities and responses to cyberattacks. In so doing, a TTX can guide mitigations and countermeasures that should be managed along with capability development. For example, through exercises, builders can put themselves in the position of an attacker to gain valuable perspective on how a would-be attacker may exploit their system. Defense organizations would benefit from incorporating cyber impacts to defense-critical infrastructure and operational technology into their TTXs. Organizations have traditionally focused TTXs on Enterprise IT, but due to its poor cyber hygiene, operational technology has been, and will continue to be, a major vector for adversaries to attack defense systems. To its credit, in the last five years, the DoD has worked to improve its incorporation of TTXs into its set of practices to identify, evaluate, and reduce risk across a variety of missions.[56] Industry has embraced the use of cyber TTXs[57] and processes to examine operational readiness of software systems.[58] DoD programs should continue to conduct such exercises on a regular basis.

Recommendations

11. **[DoD] Software-Intensive SLx:** As part of the DSOP, the DoD should create an SLA/SLO/SLI taxonomy with measurable definitions for key Agile methodology concepts, including DevOps and DevSecOps. Utilizing these definitions to measure key metrics, such as development time, deployment frequency, and error rates, program offices will have a language to incentivize and tune vendor behavior. The Defense Innovation Board's suggestion for how to detect Agile BS and the GSA's DevSecOps metrics and maturity model are good starting points for this effort.[59] Importantly, these definitions should be accompanied by parameters—high, medium, and low values. The goal is not to create strict binaries used to categorize "good" or "bad" behavior, but to accurately measure (and map to honest requirements) the performance of software development and deployment.

12. **[DoD] Rethink Perimeter Defense of the DoDIN:** The DoD's cybersecurity practices should be aligned to support the increasing dependence on certified network and computing resources outside the organization's

[52] FireWatch Solutions, "Emergency Communications: What is a PACE Plan?" July 6, 2018, "https://medium.com/firewatch-solutions/emergency-communications-what-is-a-pace-plan-694f14250bd2.

[53] Daniel E. Geer, Jr., *A Rubicon*, Hoover Institution Essay, Aegis Series Paper No. 1801, February 5, 2018, https://lawfareblog.com/rubicon.

[54] Chris Jones et al., "Service Level Objectives" in *Site Reliability Engineering: How Google Runs Production Systems*, ed., Betsy Beyer, (California: O'Reilly Media), accessed July 15, 2020, https://landing.google.com/sre/sre-book/chapters/service-level-objectives/; Adrian Hilton, AJ Ross, and Dave Rensin, "SLOs, SLIs, SLAs, oh my—CRE life lessons," Google Cloud, January 31, 2017, https://cloud.google.com/blog/products/gcp/availability-part-deux-cre-life-lessons.

[55] Community Emergency Response Team, "Tabletop Exercise #1," https://www.fema.gov/media-library-data/20130726-1917-25045-7806/cert_tabletops_combined.pdf.

[56] US Department of Defense*, The Department of Defense Cyber Table Top Guidebook, Version 1.0,* July 2, 2018*;* NAVAIR*, There is a NAVAIR product from 2016 which is Distribution D.*; Lockheed Martin*,* "Built In, Not Bolted On," https://www.lockheedmartin.com/en-us/news/features/2018/built-in-not-bolted-on.html.

[57] FireEye, "Tabletop Exercise: Test your organization's cyber incident response plan with scenario gameplay," accessed July 15, 2020, https://www.fireeye.com/services/tabletop-exercise.html.

[58] Skelton Thatcher, run-book-template, accessed July 15, 2020, http://runbooktemplate.info.

[59] US Department of Defense, "DIB Guide: Detecting Agile BS," October 9, 2018, Version 0.4, https://media.defense.gov/2018/Oct/09/2002049591/-1/-1/0/DIB_DETECTING_AGILE_BS_2018.10.05.PDF; GSA Tech at GSA, *GSA Tech Guides*.

network boundaries. Yet, the DoD's approach to securing the DoD Information Network (DoDIN) imposes meaningful barriers to access to improve the security of this collection of networks and systems by linking to a diverse range of vendors. As the Air Force celebration of a Cloud Native Access Point (CNAP) to its CloudOne network underlined, breaking free of the need to traverse this network boundary is a point of programmatic pride: "… the more we move to the cloud, the more we have to be on the cloud without having to bring anyone back to the DoDIN … The scope of the DoD enterprise appears to exceed the DoDIN and not just in acquisitions and development."[60] In a recent article, CYBERCOM's commander and senior adviser highlighted the command as having incubated a zero-trust approach to network security.[61] The DoD should task the CIO's office to support experimental efforts to facilitate simplified access into DoD networks and organizations like the CNAP. The DoD CIO should also initiate a comparative study of how the DoD treats network boundaries relative to other large computing-intensive organizations like Amazon, Google, and Siemens to identify specific practices and policies to implement for the DoD within the following year. The DoD is unlikely to be ready for a widely adopted zero-trust model, where security policies follow devices more than specified system boundaries. But improvements in these baseline practices would lower the barrier to entry for other services and vendors more easily in future while balancing core security requirements.

[60] Jason Miller, "Air Force's Game-Changing Approach to Cloud Accreditation," Federal News Network, July 30, 2020, https://federalnewsnetwork.com/ask-the-cio/2020/07/air-forces-game-changing-approach-to-cloud-accreditation/.

[61] Paul M. Nakasone and Michael Sulmeyer, "How To Compete In Cyberspace," *Foreign Affairs*, August 25, 2020, https://www.foreignaffairs.com/articles/united-states/2020-08-25/cybersecurity.

Conclusion

Mission resilience is not a product of whizbang technology innovation or novel organizational practice—it is the diligent pursuit of adaptation and iteration in the face of changing circumstances. History provides a rich example in the CORONA program. The Central Intelligence Agency's Project CORONA,[62] which in cooperation with the US Air Force produced the United States' first series of reconnaissance satellites to carry out photographic surveillance of adversarial nations, stood as an embodiment of a resilient organization. Established in 1959, CORONA developed a healthy relationship with failure after experiencing thirteen unsuccessful missions within seventeen months, before its first successful payload capture. Early CORONA attempts were plagued by incessant problems—even solutions to prior problems surfaced new problems for which new solutions were required. Constantly embracing and learning from failure is critical to growing adaptive capacity and building resilience.

Even in the absence of an obvious crisis or major attack, resilient organizations must be relentless in their pursuit of adaptive capacity. For software-intensive mission systems, pressure comes from all directions. Users advocate for new functionality. Power users employ components in unexpected ways, unlocking new value while possibly expanding the attack surface. Users, researchers, and adversaries find bugs in the system, prompting urgent engineering rework and tense forensic investigations. An organization's inability to successfully manage change in any of these areas may lead to overall mission failure. Although regular faults and failures continued throughout CORONA's tenure, until its last mission in 1972, the program was considered to be a success due to its commitment to continuous improvement by learning from failures and quickly adjusting as needed.

But organizations should not wait for crises to present themselves—they must engage in an ongoing campaign of deliberate practice.[63] Even as failures grow scarcer, it is critical for programs to extract valuable insights from their successes. Each day, organizations must consciously and deliberately explore new stressing conditions, pushing themselves just beyond their known limits to learn what failure looks like. In this way, they will deepen their understanding of the performance envelope for the mission system, and they will refine the playbooks to ensure critical capabilities do not collapse under pressure. The CORONA program's commitment to learning and adapting, even as failures grew scarcer, translated to long-term success, as the program went on to successfully capture over one hundred film return capsules.[64] This demands an honest, unflinching assessment of what isn't working, and a willingness to make the modifications necessary. Ultimately, in the midst of a major crisis, the skills nurtured in an organization, and the relationships cultivated among far-flung stakeholders, will deliver returns by reducing the severity and duration of disruptions.

Mounting successes allowed CORONA to methodically grow in complexity. Development through iteration based on learning and constructive feedback allows organizations to manage complexity. Rather than attempting to build in overwhelming complexity and functionality from the outset, resilient organizations, particularly those that make software, should deploy and iterate on a minimum viable product.[65] For CORONA, managing technological complexity meant slowly integrating new technologies to overcome the first phase of the program's operational shortcomings, such as the amount of recoverable film and image quality. The next phase of the program saw the implementation of the Mural camera, an advanced two-camera system capable of adding dimension to images by taking two photos at once from slightly varying angles—analysts were able to stereoscopically gauge structure heights, resulting in groundbreaking intelligence. Later, advancements in booster rocket propulsion allowed for heavier payloads, and thus capacity for more advanced cameras like the J-1 and J-3. By managing the complexity of its people and processes, CORONA was able to grow its team and office space from its original thirteen people and eight hundred square feet to over one thousand five hundred people and four hundred thousand square feet by 1972.[66]

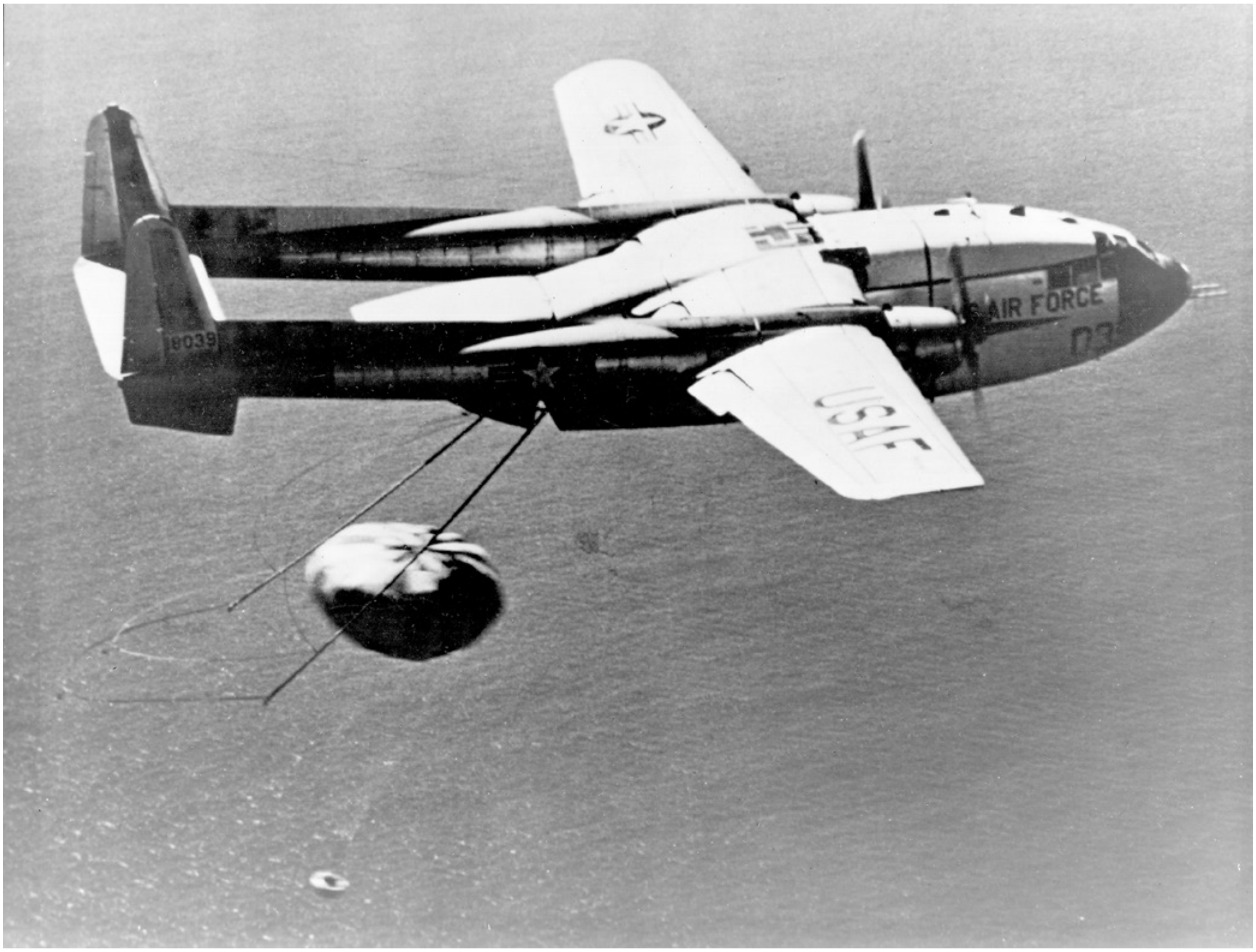
[62] Kevin C. Ruffner, ed., *Corona: America's First Satellite Program* (Washington, D.C.: History Staff, Center for the Study of Intelligence, Central Intelligence Agency, 1995), https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/corona.pdf.

[63] James Clear, "Deliberate Practice: What It Is, What It's Not, and How to Use It," accessed August 4, 2020, https://jamesclear.com/deliberate-practice-theory.

[64] Central Intelligence Agency, "A Look Back … CORONA: The Nation's First Photoreconnaissance Satellite," April 30, 2013, https://www.cia.gov/news-information/featured-story-archive/2010-featured-story-archive/corona-the-nation2019s-first-photoreconnaissance-satellite.html.

[65] Agile Alliance, "Minimum Viable Product (MVP)," accessed September 25, 2019, https://www.agilealliance.org/glossary/mvp/.

[66] *Reel America*, "A Point in Time: The Corona Story," C-SPAN, January 31, 1972, video, 57:59, https://www.c-span.org/video/?321255-1/discussion-cias-corona-satellite-program.



US Air Force C-119J recovers a CORONA capsule returned from space. *Source:* US Air Force

As CORONA's technological and organizational process advancements improved, so too did its mission rates. Originally costly ordeals, capsule recoveries became routine operations, and by 1965, approximately three to four recoveries were made each month. Improving mission speed is critical for resilient organizations, and the faster an organization can integrate feedback and resolve vulnerabilities, the better. By increasing mission frequency and capsule recovery, CORONA provided incalculable strategic value, arming the United States with precise knowledge of the Soviet Union's missile stockpile and the confidence to enter into the Strategic Arms Limitation Talks.[67]

Project CORONA exemplifies a resilient organization. Identifying the principles and practices that underpinned its resilience, and those of organizations like it, is critical to building mission resilience in the DoD. Software may be immortal, but it's not static, and it doesn't create itself. If an organization is going to sustain immortal software indefinitely, as the Defense Innovation Board has advised,[68] then it must engage in concerted, deliberate efforts to plan for and manage change throughout the system life cycle. Where this software implicates safety-critical systems and has direct impact on human life, the stakes for experimentation are higher and barriers to

[67] William J. Broad, "Spy Satellites' Early Role As 'Floodlight' Coming Clear," *New York Times,* September 12, 1995, https://www.nytimes.com/1995/09/12/science/spy-satellites-early-role-as-floodlight-coming-clear.html.

[68] Dr. Craig Fields, chairman, DSB, in memo to USD(R&E) presenting the report of the DSB Task Force on the Design and Acquisition of Software for Defense Systems, February 2018.

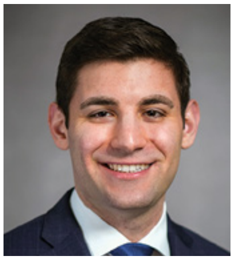
entry for novelty greater. This report developed the concept of mission resilience and applied it to the defense enterprise through four principles: 1) embrace failure, 2) improve your speed, 3) always be learning, and 4) manage trade-offs and complexity. Taking specific practices from each of these principles, the report made recommendations to senior policymakers in the US government, the operational defense enterprise in the DoD, and service and combat unit leadership.

The United States and its allies will be confronted with a challenging security landscape in the coming decades as a new generation of major mission systems is developed amid a changing technology base and rapid evolution in the diversity and capability of adversaries. This report helps to bring long-overdue conversations about innovation in software development and deployment forward to apply to cyber-physical systems and the next generation of technology opportunities. Mission resilience is a long-term goal and the recommendations here are intended as a way to support ongoing reforms and extend them deep into personnel, classification, and procurement policy. Change is painful in any organization, especially one with the internal complexity of the DoD, but it is a necessary paradigm for survival. Defense organizations outside of the United States can also find in this report a framework for evolution. Those entities may find their comparatively smaller size or budgets grant more agility in implementing these reforms. The continued pace of innovation must be embraced as a palliative to fragility and brittleness. Mission resilience offers a model that just might make such an embrace possible. The rest is up to us.

About the Authors

Dr. Trey Herr is the director of the Cyber Statecraft Initiative under the Scowcroft Center for Strategy and Security at the Atlantic Council. His team works on cybersecurity and geopolitics including cloud computing, the security of the internet, supply chain policy, cyber effects on the battlefield, and growing a more capable cybersecurity policy workforce. Previously, he was a senior security strategist with Microsoft handling cloud computing and supply chain security policy as well as a fellow with the Belfer Cybersecurity Project at Harvard Kennedy School and a non-resident fellow with the Hoover Institution at Stanford University. He holds a PhD in Political Science and BS in Musical Theatre and Political Science.

Reed Porada is a security researcher and instructor at BCI focused on the socio-technical facets of system design and operations. His research has focused on helping get to the "so what" of both defensive and offensive cyber measures. To provide decision makers and teams with context for their work, Reed applies skills in systems thinking, communication, technology awareness, and hands on system assessments. At BCI, Reed focuses his research on understanding how attackers redefine system boundaries and use systems in unexpected ways. This work will help inform defensive approaches and develop better tools for reasoning about system security. Reed leads BCI training in cyber systems analysis, focusing on developing systems thinking skills of developers up to managers. Previously, Reed was a staff member at MIT Lincoln Laboratory for ten years. He was responsible for test and evaluation, test automation research, red-teaming of cyber systems, and blue system architectures in support of DoD and other government programs. Prior to joining Lincoln Laboratory, Reed was computer scientist at the Naval Research Laboratory focused on wireless communication systems. He holds an MS in software engineering from Carnegie Mellon University and a BS in Computer Science from University of Maryland, College Park.



Simon P. Handler is the assistant director of the Atlantic Council's Cyber Statecraft Initiative, within the Scowcroft Center for Strategy and Security. In this role, he manages a wide range of projects at the nexus of geopolitics and international security with cyberspace. Prior to joining the Atlantic Council, he served as a special assistant in the United States Senate. During his time on the Hill, he was a congressional fellow with the Wilson Center's Congressional Cybersecurity Lab and Congressional Artificial Intelligence Lab, and completed the East-West Center's Congressional Staff Program on Asia. He holds a BA in International Relations & Global Studies, with a concentration in International Security, and Middle Eastern Languages & Cultures from the University of Texas at Austin.

Orton T. Huang is a technical staff member in the Secure Resilient Systems and Technology Group. He conducts research in building and operating resilient mission systems. His current focus is on adapting and adopting observability, chaos engineering, automation, and machine learning to enable resiliency. In 2003, Mr. Huang joined Lincoln Laboratory as a member of the Advanced Networks and Applications Group. Since then, he has worked on developing and analyzing advanced multipath airborne communications, helped lead the Laboratory's net-centric efforts, and supported various Air Force and Navy C2 modernization programs focusing on security and resiliency. Mr. Huang received his BS and MS degrees from MIT.

Stewart Scott is a program assistant with the Atlantic Council's GeoTech Center. In this role, he manages a wide range of projects at the intersection of emerging technologies and dynamic geopolitical landscapes. He also conducts research and provides written analysis for publication on Atlantic Council platforms and works on joint projects with other centers in the Atlantic Council. Stewart earned his BA from Princeton University at the School of Public and International Affairs along with a minor in Computer Science. His course of study centered on misinformation, social media policy, online extremism, journalism, and American political and economic history. He joined the Atlantic Council after interning with its Cyber Statecraft Initiative in the Scowcroft Center for Strategy and Security.

Dr. Robert D. Lychev is a technical staff member in the Secure Resilient Systems and Technology Group at MIT Lincoln Laboratory. He joined the Laboratory in September 2014. His research focuses on leveraging techniques from systems theory, network science, and cybersecurity to problems in systems analysis and resilience engineering in contested environments. Dr. Lychev earned his BS degrees (summa cum laude) in computer science and physics and an MS degree in computer science from the University of Massachusetts, Amherst, in 2006 and 2008, respectively. He earned his PhD degree in computer science from the Georgia Institute of Technology in 2014. As a PhD student, he focused on evaluating the utility of security-enhanced communication protocols in terms of their provable security and performance guarantees, as well as their deployment challenges in practice. For his work, Dr. Lychev was awarded the Applied Networking Research Prize by the Internet Research Task Force.

Jeremy Mineweaser is a senior staff member in the Secure Resilient Systems and Technology Group. Since joining the Laboratory in 2000, he has conducted research on satellite communications, airborne networks, net-centric operations, information assurance, and cyber operations. His current focus is on the design and execution of wargames that explore the challenges of successfully operating large-scale, software-intensive systems in contested environments. Mr. Mineweaser received a BS degree in computer engineering from Georgia Institute of Technology.

Acknowledgements

This project would not have been possible without the support of Maj. Chris Hocking, Reed Porada, Orton Huang, MIT Lincoln Laboratory, and the United States Air Force. Particular thanks go to Martine Kalke, June Lee, and Lily Liu for their support and feedback.

Atlantic Council

Board of Directors

Atlantic Council

The Atlantic Council is a nonpartisan organization that promotes constructive US leadership and engagement in international affairs based on the central role of the Atlantic community in meeting today's global challenges.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force.

Please direct inquiries to:

Atlantic Council

1030 15th Street, NW, 12th Floor, Washington, DC 20005

(202) 463-7226, www.AtlanticCouncil.org