CrossTalk

The Journal of Defense Software Engineering

November 2023

CyberCrime

CrossTalk: The Journal of Defense Software Engineering

CrossTalk: The Journal of Defense Software Engineering is sponsored by the United States Air Force (USAF) 309 Software Engineering Group (SWEG) at Ogden Air Logistics Complex (ALC). It is also supported by other partners within the Department of Defense (DoD), other USAF systems, and the software engineering community.

The USAF 309 SWEG also publishes CrossTalk, provides editorial oversight, and technical review of the journal. The mission of CrossTalk is to encourage the engineering development and proper management of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

CrossTalk Online: Current and past issues are posted at the following two locations. The All Partners Access Network (APAN) and Defense Technical Information Center (DTIC)

https://community.apan.org/wg/crosstalk/

https://www.dodtechipedia.mil/dodwiki/x/HwDqFQ (Requires .mil domain for full support)

The APAN site requires registration to be a regular member and to interact with others. The DTIC website includes reports designated as unclassified and unlimited information and can be reached at: https://discover.dtic.mil/technical-reports/

Subscriptions: Please send an email to the publisher to receive a notification when each new issue is published online.

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the Technical Review Board (TRB) prior to publication. Please follow the *Author Guidelines,* available at either of the two sites above. CrossTalk does not pay for submissions. Published articles remain the property of the authors and may be submitted to other publications. Security agency releases, clearances, and public affairs office approvals are the sole responsibility of the authors and their organizations. Potential articles can be emailed to: 517SMXS.Crosstalk.Articles@us.af.mil

Reprints: Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with CrossTalk.

Trademarks and Endorsements: CrossTalk is an authorized-publication for members of the DoD. Contents are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, the sponsors, or co-sponsors. All product names referenced in this issue are trademarks of their respective companies.

Publishing Schedule and Back Issues: CrossTalk is currently being published quarterly. Please phone or email us to see if back issues are available, free of charge.

ISSN 2160-1577 (prior print versions); ISSN 2160-1593 (online)

CrossTalk Staff

Sponsor
James L. Diamond Jr.

Managing Director
Christian J. Durain

Assistant Director
Malissa Jones

Managing Publishers
Lennis L. Burton
Siria L. Snounou
Destinie Comeau

Technical Reviewer
Robert DeVincent

Contact us

Phone
Lennis L. Burton, (801) 775-3262

Siria L. Snounou, (801) 777-4734

Destinie Comeau, (801) 775-3246

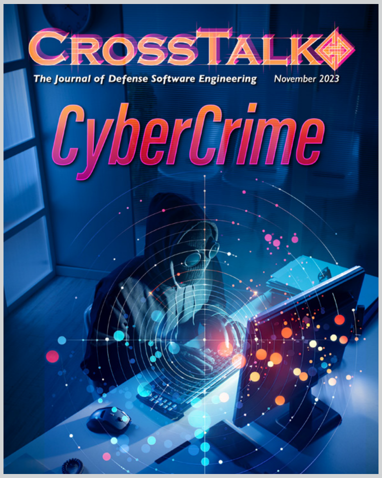
E-Mail
517SMXS.CrossTalk.Articles@us.af.mil

SWEG Socials

Connect with us at the 309th SWEG socials

CrossTalk: The Journal of Defense Software Engineering

Directory



Cover Design by Kent Bingham

CyberCrime

FIGHTING CYBERCRIME: CONFIDENTIALITY, INTEGRITY, AND AVAILABILITY

Joseph F. Bradley Jr.
Director, Cyber Resiliency Office for Weapons Systems

Transnational criminal organizations and malicious cyber actors are constantly seeking new ways to attack everything from our Defense Industrial Base to our critical infrastructure, to local, state, and federal government entities. Moreover, state actors are more than capable of targeting American companies to extract proprietary technology to advance their own military and geopolitical interests. These actions can result in billions of dollars worth of losses, and, in many instances, threaten our national security.

The Department of Defense's 2023 Cybersecurity Strategy addresses some of the challenges we as a nation face from malicious cyberspace actors seeking to exploit our technological vulnerabilities and undermine our competitive edge. These important issues also underscore the need to implement an all-of-government approach to cybercrime by consolidating diplomatic, law-enforcement, military, and intelligence communities to tackle the challenge. We need to build cyber resilience into every facet of our networks, critical infrastructure, and DOD systems – not just as individual components, but rather at the architecture level.

Protecting our Defense Industrial Base is particularly important given that it develops, manufactures, and maintains sensitive technologies, and unlike in authoritarian nations like China and Russia where the government has nationalized the Defense Industrial Base; the United States, as a free-market capitalist nation, often relies on the innovation of our partners in private industry. This is not, coincidentally, why malicious cyber actors routinely target our companies.

Likewise, hacking into a Government Website, Denial of Service Attacks, Identity Theft, Stolen Credit Card information, Phishing campaigns…..what do these all have in common? Irrespective of who the culprits may be, they are all cybercrimes.

Thus, we will examine our actions to deter, prosecute and, most importantly, recover from these malicious deeds.

Why do we, in the Government, care about these crimes? First, let's look at the impact of these actions as they relate to operational activities.

There are three major components to cybersecurity; Confidentiality, Integrity and Availability.

Confidentiality refers to the protection of sensitive information from unauthorized access. This information can include the exploitation of Personally Identifiable Information to either harass people at home, or, in other instances, to steal their identities, their medical information, and beyond.

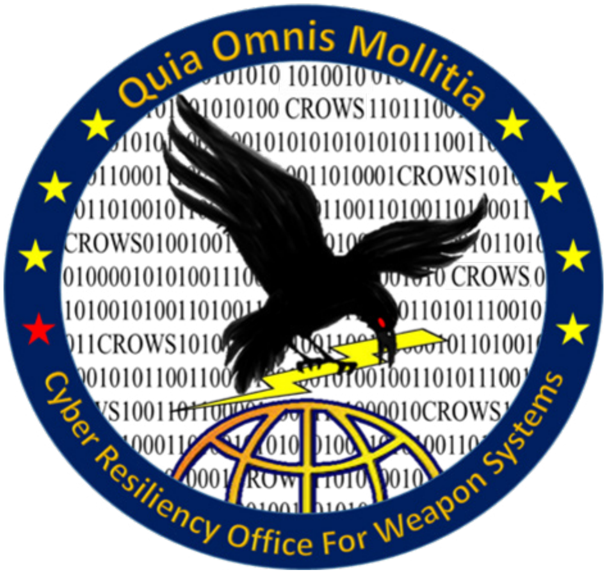


Figure 1. *CROWS Logo.*

What if a leader's food allergies were made public? Knowing this could result in the adulteration of someone's food, which could have a serious impact on their health and their ability to successfully complete their mission.

Integrity refers to information modification or deletion – what if maintenance information was exfiltrated from an online source, modified and reintroduced onto the site? Could this damage equipment when accomplishing "prescribed" maintenance? What about hospital records and situations where actors can change interactions between medicines or known side effects? What happens when we can't trust the data? We still need to accomplish the mission, don't we? Can we rely on institutional expertise to question the data and will people listen?

Availability refers to data being available when needed; what is our tolerance for the "blue circle" on our e-mail or database queries? What frustration can easily be introduced when information access is critical to making timely decisions? How many decisions could be negatively influenced because we do not have timely data, and what would be the impact to mission success?

Adversaries only need to be successful at any one of these activities which can then result in mission disruption and loss of confidence in our automated systems.

When it comes to adversaries, we also need to identify who they are. Whether state sponsored, a transnational terrorist organization, or beyond. We would also need to identify whether they are online disruptors/influencers looking to make a name for themselves, "script kiddies" seeking to show off to their friends, or a completely different set of actors. The important takeaway is that the enemy is ubiquitous which means we must plan for and defend against any or all of them.

Before we get there, we have to be able to understand our own systems; whether we have identified our external and internal interfaces; who should have access to our data; whether we monitor our systems, data, and websites to ensure they still contain the data we think they do; whether we are experiencing any degradation in performance; and a number of other issues. We also need to think about how robust our back-up capabilities are, which means we will need to have backup systems on stand-by for automatic switchover if we detect an issue.

Now, if this is the first time you are reading about any of these issues then you should be, rightfully, concerned. However, rest assured, the nation has been wrestling with these questions since the movie War Games came out in 1983; although the threats are more sophisticated today, the impacts remain as dire. This issue will discuss steps that the Department of the Air Force (to include our Sister Service, Space Force) are taking to improve the Confidentiality, Integrity, and Availability of our war-fighting systems. And if you don't think our business enterprise systems, supply systems, and other support systems are warfighting systems, think again!

- Joseph F. Bradley Jr., Director, Cyber Resiliency Office for Weapons Systems, Hanscom Air Force Base

Call For Articles

If your experience or research has produced information that could be useful to others, Crosstalk can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for the areas of emphasis we are looking for.

Generations in software

February 2024 Issue

Submission Deadline:
October 31, 2023

Artificial Intelligence: transforming the digital Landscape

May 2024 Issue

Submission Deadline:
March 15, 2024

Big Data: Opportunites and Limitations unknown

August 2024 Issue

Submission Deadline:
June 15, 2024

Please follow the Author Guidelines for Crosstalk, available at the APAN or DTIC site.

We accept article submissions on software-related topics at any time, along with Letters to the Editor, Open Forum, and BackTalk. To learn more about the types of articles we're looking for, please visit the above sites or contact us by email or phone

Contact Us

By phone

Lennis L. Burton, (801) 775-3262
Siria L. Snounou, (801) 777-4734
Destinie Comeau, (801) 775-3246

By email

517SMXS.CrossTalk.Articles@us.af.mil

Fighting CyberCrime *with DevSecOps*

Jason Weiss

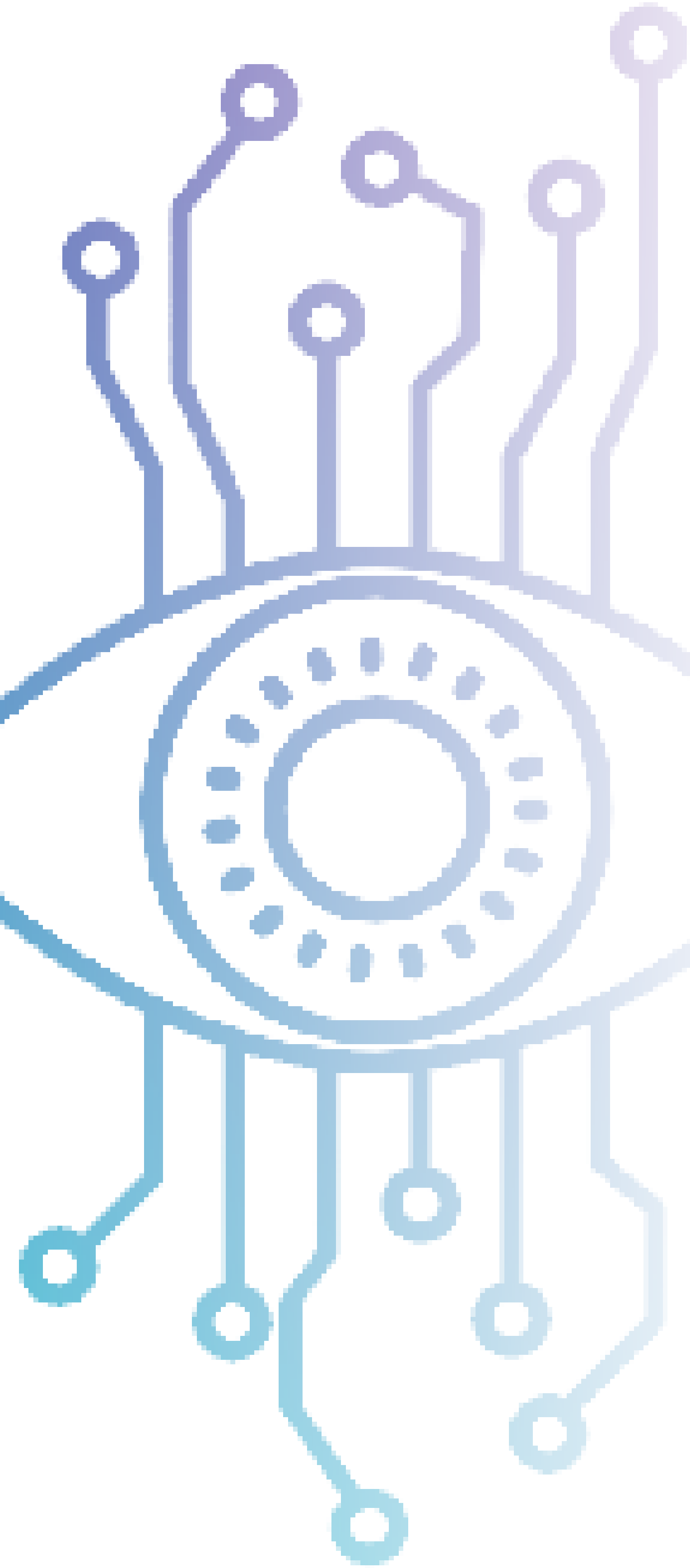
Executive Advisor at Softrams

Cybercrime is so pervasive that there is a universal recognition of the term across every demographic. Media coverage has led to most people describing it as either one of those annoying data thefts or interruptive ransomware attacks. Yet, cybercrime is so much more than an interruptive annoyance, and, as career software professionals, it is imperative that we have a more profound understanding of what constitutes cybercrime and what actions engineers can take to mitigate cybercrime. As software professionals working in or adjacent to national security missions, fighting cybercrime is an explicit non-functional requirement.

Formally, cybercrime has three distinct categories: Crimes against People, Property, and Governments. Several examples in the people category include cyber harassment, credit card fraud, spoofing, identity theft, and online libel/slander. In the national security space, law enforcement and intelligence professionals may be professionally engaged in some of these areas. The most obvious example are the Child Exploitation and Human Trafficking Task Forces (CEHTTFs). Intelligence professionals may be conducting sanctioned operations in, say, cyber stalking or spoofing. Generally, the daily decisions we make as software engineers have marginal impact in this category of cybercrime.

In sharp contrast, our software development lifecycle, our process methodologies, and our daily digital hygiene substantially affects the remaining two cybercrime categories. Crimes against Property range from Distributed Denial-of-Service (DDOS) attacks to hacking, virus transmission to intellectual property rights violations. Furthermore, Crimes against Government also include hacking, as well as cyber warfare, cyber terrorism, and accessing sensitive or classified information.

The broader Defense Industrial Base (DIB) software engineering community is likely not following the developments of the DOD's Cybersecurity Maturity Model Certification (CMMC), but rest assured, DIB companies are acutely aware of its financial ramifications. For our purposes here, we'll depict CMMC using a gross oversimplification. Companies were contractually obligated to adopt a set of cybersecurity controls, defined in National Institute of Standards and Technology (NIST) SP800-171, that are intended to mitigate key risks (but not all risks) in both the Property and Government cybercrime categories. But they didn't… and the DOD knew they didn't. To attempt to rectify this situation,



the DOD has crafted CMMC in a way that requires a third-party to review the implementation and attest to a company's compliance. CMMC is important to the DOD because so much unclassified but sensitive information is being illegally accessed.

As a community, software engineers supporting the DIB must take steps now to familiarize themselves with a different NIST standard, NIST SP800-218, Secure Software Development Framework (SSDF). Like CMMC, the SSDF is intended to help organizations of all shapes and sizes mitigate key risks (but not all risks) across both the Property and Government categories. Unlike the CMMC that explicitly targets the InfoSec community, the SSDF explicitly targets the software engineering community. Every engaging conversation on hacking, regardless of if it is the Property or Government variety, rapidly pivots into a conversation about the software development lifecycle, the process methodologies development teams employ, and the software supply chain. And that brings us to what some will portray as the most controversial statement in this whole issue of Crosstalk:

Software development environments must be more secure than production environments.

For too long, the software development community has operated from a position of entitlement. We expect and thinly justify why, as software practitioners, we must have local admin rights. We use terms like DevSecOps, but too many engineers across the DIB are more interested in spending their weekend learning a new obscure programming language instead of learning critical skills, like threat modeling. We have convinced ourselves that it is not possible to write software at the speed of relevance without persistent Internet connectivity, access to package managers, and integrating a library that saves an hour of coding without understanding anything about the 37 dependencies that library brings into the application, three of which only have API documentation in a foreign language.

Cybersecurity Awareness Month is every year in October. Who among us hasn't made fun of the annual cybersecurity training regimen that reminds us not to pick up a USB stick off the sidewalk and insert it into our government issued laptop and to use extreme caution when clicking hyperlinks in an email? Yet, as professional software engineers, sometimes our com-

munity needs reminded that the “Sec” in DevSecOps stands for security. Its presence is supposed to be a daily reminder that even routine activities should be viewed through a security lens. As software engineers, we need to aspire to better understand what has historically been abstract concepts to us, including things such as risk versus residual risk, how to create an effective business impact analysis, or why capturing and maintaining a table of ports, protocols, and service methods (PPSM) cannot be viewed as merely a paper exercise. This is not an exhaustive list.

Over the last few years, the phrase ‘software bill of materials’ (SBOMs) has become commonplace among software teams. Pause for a moment and ask yourself if your software team is merely a producer or a producer and a consumer of SBOMs. If your team hasn’t integrated SBOM consumption into your Software Developement Life Cycle (SDLC) and associated process methodologies, this illustrates a gap large enough for cybercrime to occur.

It was Aristotle that codified the idea that the whole is greater than the sum of its parts. Unfortunately, this idea also applies to cybercrime. The software development ecosystem is a living organization. Over the last two decades, immense specialization has occurred, and each tool brings its own set of unique benefits, but it also brings its own attackable surface area. Where production should be a predictable and deterministic environment to secure and protect, the development ecosystem is, by its nature, nondeterministic because of the ebb and flow of exploring new languages, new tools, new software components and new open-source software libraries. These very activities are prima facie evidence why development must be more secure than production. That starts with those of us who wake up excited and proud to sling code day in and day out. We must make the whole more secure than the individual pieces and parts that comprise our continuous integration/continuous delivery (CI/CD) pipelines.

It is easy for software engineers to reduce cybercrime to an operations problem or to dismiss the issue as human error (phishing). Combating cybercrime is not easy. Truly embracing and realizing the 42 tasks defined in the SSDF will not be easy. Giving up local admin rights will not be easy. Establishing an evaluation process and requiring every third-party component, commercial or open source, to be evaluated before inclusion in the software you’re creating will not be easy. The challenge and the reward are found in the actions we take as a software engineering community. Let’s not mince words here: if we are going to say DOD software engineering teams and software factories have adopted DevSecOps, then that must mean we collectively realize every day we are actively doing those things that are necessary to fight cybercrime.

About the Author

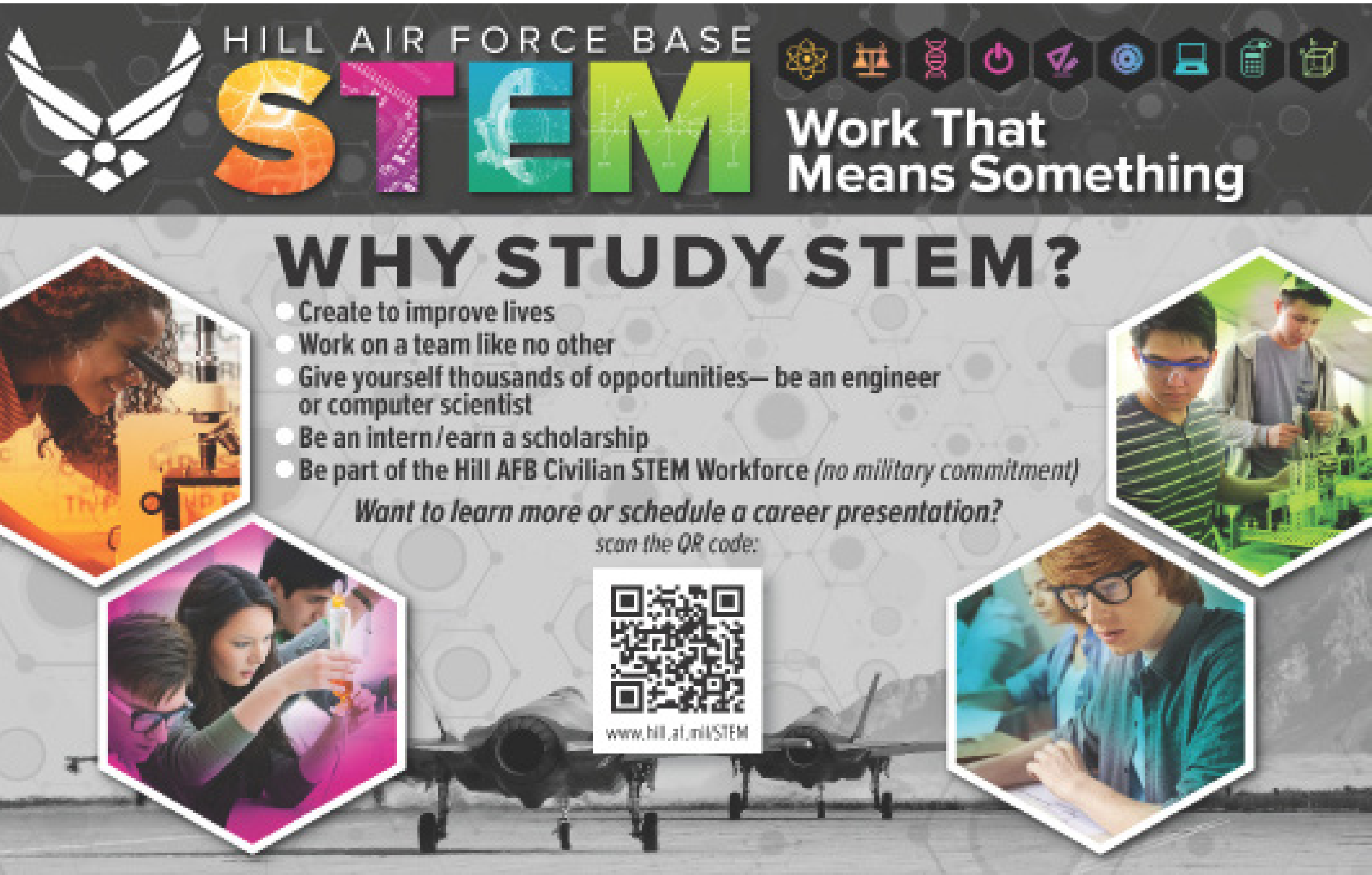
Jason brings an exceptional background in software engineering architecture and cryptology dating back to his service in the US Navy as a cryptologist during the first Gulf War. He was the founding DOD Chief Software Officer and has held software executive leadership positions at both large and small defense contractors. He is presently an executive advisor at Softrams and the co-founder of Digital Triad Group, Inc., a firm specializing in SSDF education.

Mr. Jason Weiss

Executive Advisor

Softrams

jweiss@digitaltriadgroup.com



Building a New Assessment: How to Assess Ransomware Attack Readiness and Recovery

Brett Tucker
CERT Program Technical Manager,
Carnegie Mellon University’s Software Engineering Institute

Abstract

Protection of our nation’s critical infrastructure and those agencies and organizations that support it is of the utmost priority. The Cyber Engineering and Resilience Team (CERT) Division at the Software Engineering Institute (SEI) of Carnegie Mellon University (CMU) aims to provide organizations with recommendations that would both reduce the likelihood of a ransomware attack and mitigate its effects if one was to occur. This article provides practical tips for organizations, so that they may develop their own ransomware assessment to gage their exposure to such risks.

Introduction

In 2021, approximately 37% of global organizations in IDC’s 2021 Ransomware Study reported being the victim of a ransomware attack. TechTarget [1], an industry provider which reported on the study, also noted that in 2021 and 2022, new ransomware trends emerged as attackers realized that certain techniques, such as supply chain attacks and double extortion, yielded better results. To get an appreciation for the scope of these attacks, AAG IT [2], an industry provider, reported that there were 623.3 million attacks in 2021. Ironically, AAG IT also reported a 23% drop of attacks in 2022, which may be an indication of improved defenses. Regardless, ransomware has targeted critical infrastructure. A ransomware attack on a water distribution system in Israel [3], for example, shook executives at American utilities, and one on a petrochemical plant in Saudi Arabia [4] revealed the vulnerability of its oil production.

Protection of our nation’s critical infrastructure and those agencies and organizations that support it is of the utmost priority. The CERT Division at the Software Engineering Institute (SEI) of Carnegie Mellon University (CMU) aims to provide organizations with recommendations that would both reduce the likelihood of a ransomware attack and mitigate its effects if one was to occur.

Methodology and Catalyst

In the wake of the Colonial Pipeline ransomware attack [5], CERT tapped its expertise in cyber risk management and assessment to give organizations an understanding of their security posture and their ability to prevent, detect, and respond to a ransomware attack. The goal of any assessment may include the collection of quantifiable evidence that relates appropriate deployment of control measures to protect their systems and demonstrate resilience in the face of an attack. The collection of such evidence may allow organizations to gain appreciation of their susceptibility to potential ransomware attacks. Unfortunately, assessments of this nature are driven largely by the context of the organization. Demographic considerations for size, resources, mission, and strategy may influence how an organization assesses its risk exposure to ransomware. Therefore, CERT would like to prescribe specific considerations and requirements for proper ransomware assessment development.

In building out such an assessment methodology, CERT recommends that organizations exploit several widely accepted standards and resources to create a robust foundation for generally accepted practices. For example, organizations may seek direction from Cyber & Infrastructure Security Administration (CISA) Cross-Sector Cyber Performance Goals (CPGs) [6] and the controls outlined in the National Institute of Standards and Technology (NIST) 800 series to build recommended control suggestions on gaps in security posture. To help organizations prioritize control selection, some organizations may also rely upon the MITRE ATT&CK Framework [7] for a perspective on attack vec-tors commonly used by ransomware attackers.

Determining which organizational assets are in scope is an important part of any assessment. The CERT Resilience Management Model (CERT-RMM) [8] may help organizations select focus areas (or domains) an organization should consider, beginning at the highest level, with the primary goals and objectives of an organization. OCTAVE FORTE [9] provides a methodology for value stream mapping where overarching objectives can be decomposed into critical services. If an organization operates in the power sector, for example, an overarching objective might be to provide a service to a customer while earning revenue. Breaking that down, the identified critical service might be the delivery of electricity. From that, the organization's critical assets can be further categorized into the following:

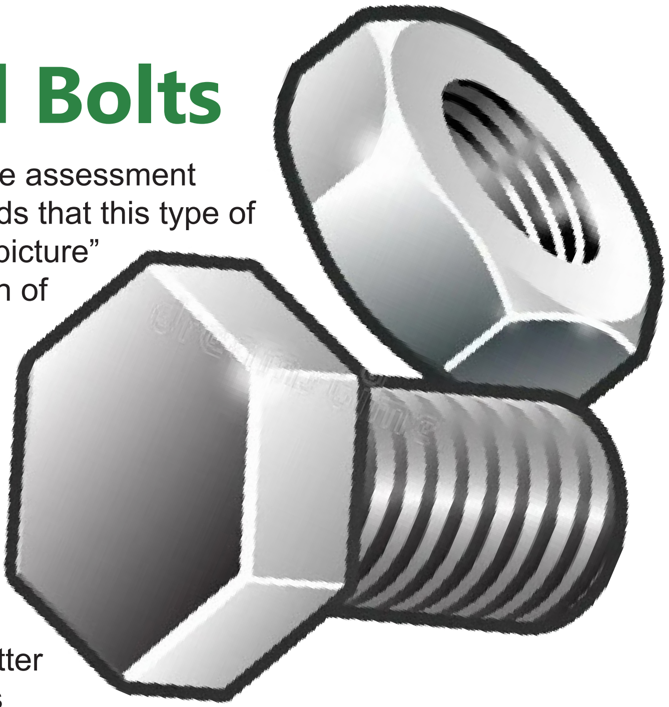
- **People**—those who operate and monitor a service
- **Information**—data associated with that service
- **Technology**—tools and equipment that automate and support the service
- **Facilities**—location or site that contains other assets
- **External Dependencies**—third-party relationships and supply chain

These assets may also be categorized as high-value assets (HVAs), as defined by U.S. Government FIPS 199 [10] because all derive their importance from their ability to meet the organization's mission.

The final step in developing a ransomware assessment is to acknowledge that, depending on available resources, many private organizations can supplement their ransomware resilience with assistance from consulting firms or cyber insurance providers. They can provide not only pre-event services, such as consultation for response strategies, but also services during and after a ransomware attack.

Ultimately, the ransomware assessment should focus on assets that derive their value from their importance in meeting the organization's service mission and assesses ransomware exposure in terms of susceptibility and ability to recover from an attack.

Assessment Nuts and Bolts



The duration of assessments depends upon the scope of the assessment and availability of resources. Regardless, CERT recommends that this type of assessment should initially survey the organization for “big picture” gaps to inform deeper dive research. Ultimately, the duration of the assessment should not be so long that the organization loses visibility and momentum for completion nor diminish the return on investment for gaining insight to ransomware risk exposure.

Preparation includes the initial notification of stakeholders, scheduling meetings, scope planning, and kickoff. A day or two may be scheduled for on-site, facilitated discussions. These facilitated assement meetings provide important interface between assessors and organizational subject matter experts to gain greater clarification on technical issues. This onsite meeting may be followed by 10-15 days of report writing and post assessment. Unless necessary, the assessed organization may lose context and momentum to address critical issues if the post assessment period lingers.

After some research and testing, CERT recommends the use of at least eight subject matter areas, also known as domains, that may help organizations identify critical questions to ask when beginning an assessment:

- **Business Continuity Disaster Recovery (BCDR)**—includes backup systems or strategies, incident response, and backup testing. This domain focuses on ensuring that organizations ask if they have the right testing scenarios for ransomware.
- **Configuration Management**—includes allow/block lists, baselines, restricting permissions, limits to installations, and registry permissions.
- **Endpoint Protection**—addresses cyber hygiene including antivirus, intrusion prevention system (IPS), and web content.
- **Identity Access Management**—includes multifactor authentication (MFA), least privilege enforcement, password management, and user/privileged account management.
- **Incident Management**—focuses on event reporting and escalation, including the type of reporting (i.e., when an incident occurs, how will it be reported across the organization).
- **Network Protection**—includes access limitations (such as remote desktop protocol), email management, and network segmentation.
- **Risk management**—includes insurance and user training.
- **Supply chain**—assessment of third-party provider practices
- **Vulnerability Management**—includes software updates, vulnerability scanning, and audit.

An organization’s capabilities in these domains can be rated as fully implemented, partially implemented, or not implemented, which then correlates to the degree of susceptibility of the organization to ransomware attacks.

Lessons Learned

As alluded to earlier, CERT has developed and conducted a variety of methodologies with a large federal civilian agency, large municipalities, and private organizations. These experiences taught CERT some valuable lessons:

- **Scope** —Trying to narrow the scope to one high-value-asset (HVA) system, such as a payroll, can be challenging. Because ransomware typically infects an entire environment—including people, technology, facilities, and external dependencies—a single-asset focus was not enough. Organizations may begin with an HVA system as an anchor and then expand to all assets touched by that system or its subsystems.
- **Operational tempo**—The organization's operational tempo must be acknowledged because it can greatly influence progress. As an example, an assessment team may realize a need to adapt to the organization's availability of critical resources such as subject matter experts. This might involve breaking up a technical exchange meeting over several days to accommodate different schedules of the stakeholders involved.
- **Domain establishment**—From the outset, it is important to establish the assessment domains to be included in scope to help the organization identify who should participate in the technical exchange meeting. A discussion surrounding BCDR, for example, would involve different employees than a meeting on risk management, which would typically involve an enterprise risk manager.
- **Data quality**—The assessment team must work together to identify the pedigree and quality of information sought. Frequent communications with the assessed organization provided clarifications and context.
- **Source documentation**—In our experience, comprehensive policies must stand behind each organizational procedure, or too often it will not get completed. The assessment should seek to advise on improving source documentation to avoid reliance on culture as a guide to day-to-day activities.
- **Terms and conditions**—Each assessment will likely involve an exchange of a substantial amount of information. This means that agreements must spell out specifics, such as whether information can be shared and the terms and conditions for sharing it. There may also be consideration if the information is shared in a safe environment.
- **Pedigree of information**—during assessment development, organizations should consider whether information may be collected through attestation or validation. Attestation involves subject matter experts declaring the existence of a practice, while validation would demand evidentiary proof of practice. Attestation may enable a quicker assessment with less strain on the workforce, while rigorous validation may take longer for collection. However, validation may provide

greater efficacy and understanding of policy implementation

In general, these lessons learned can be applied to the development of any assessment, not just one about ransomware. Similar relevant assessments for development could include zero trust architecture, mobile devices, and cloud implementation.

Going Beyond Ransomware

Ransomware represents one of the premiere threats to critical infrastructure. Recent events, such as the attack on the Colonial Pipeline, brought the threat of ransomware and its impact on critical infrastructure to the forefront of concerns for our nation's security.

While most CERT assessments focus on generalized cyber ecosystem review, the suggestion of developing a ransomware assessment is different in that it focuses on a specific form of attack rather than a type of asset. Ransomware attacks are common enough that in building the assessment, organizations may be able to draw upon fundamental gaps that may be exploited by other attack vectors.

Like all things in risk management, assessment development has an iterative lifecycle. Organizations must constantly work to improve their methodology in the wake of new assessment opportunities.

References

[1] Kerner, Sean M. "Ransomware Trends, Statistics and Facts in 2023", TechTarget.com, https://www.techtarget.com/searchsecurity/feature/Ransomware-trends-statistics-and-facts/, January 2023.

[2] Griffiths, Charles. "The Latest 2023 Ransomware Statistics (updated August 2023)", AAG, https://aag-it.com/the-latest-ransomware-statistics/, August 2023.

[3] Gross, Judah, A. "After Alleged Iranian Cyberattack, Israel's Water Authority Beefs Up Defens-es", Times of Israel, https://www.timesofisrael.com/after-alleged-iranian-cyberattack-israels-water-authority-beefs-up-defenses/, July 2021.

[4] New York Times. "A Cyber Attack in Saudi Arabia Had Deadly Had a Deadly Goal, Experts Fear Another Try", New York Times, New York, https://www.nytimes.com/2018/03/15/technology/saudi-arabia-hacks-cyberattacks.html/.

[5] Easterly, Jen. "The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years | CISA", Department of Homeland Security – CISA, Washington DC, https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years/, May 2023.

[6] CISA. "Cross-Sector Cybersecurity Performance Goals 1.0.1", Department of Homeland Secu-ri-ty, Washington DC., https://www.cisa.gov/sites/default/files/2023-03/CISA_CPG_REPORT_v1.0.1_FINAL.pdf/, March 2023.

[7] Strom, Blake E., Applebaum, Andy, Miller, Doug P., Nickels, Kathryn C., Pennington, Adam G., Thomas, Cody B. "MITRE ATT&CK: Design and Philosophy", MITRE Corporation, McLean, VA., https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf, July 2018.

[8] Caralli, Richard A., Allen, Julia H., White, David W. "CERT Resilience Management Model for Managing Operational Resilience", Addison-Wesley Professional, Philadelphia, PA., https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30375/, July 2016.

[9] Tucker, Brett A. “Advancing Risk Management Capability Using the OCTAVE FORTE Pro-cess”, Software Engineering Institute, Pittsburgh, PA., https://resources.sei.cmu.edu/asset_files/TechnicalNote/2020_004_001_644641.pdf/, November 2020.

[10] Computer Security Division, Department of Commerce National Institute of Standards and Technology, “FIPS PUB 199 Standards for Security Categorization of Federal Information and In-formation Systems”, Department of Commerce, Washington DC, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.199.pdf/, February 2004.

About the Author

Brett Tucker is the Technical Manager of Cyber Risk in the CERT Program at Carnegie Mellon University’s (CMU) Software Engineering Institute (SEI). Brett is responsible for a research and development portfolio focused on improving the security and resilience of the Nation’s critical infrastructure and assets with specific focus on risk management and resilience. Brett is also adjunct CERT Faculty at the Heinz College.

Mr. Brett Tucker

Technical Manager

Carnegie Mellon University

batucker@cert.org

December 18-20
San Diego, CA

Weapon Systems Software Summit

@ the DoD Maintenance Symposium

Explore solutions to the most difficult and complex software issues and challenges impacting the Department of Defense.

Summit Topics:

- DevSecOps
- Software Assurance
- Weapon System Software Development on the Cloud

Register for the Summit and book your Hotel below!

CYBERSECURITY: THE DRIVE FOR CONTINUING INNOVATION

REBEL BROWN
CEO,
COGNOSCENTI,
INC.

STEVE FOOTE
CTO,
PHENOMENATI

Introduction

Cybersecurity is the critical link in protecting networks, devices, and data from unauthorized access or criminal use. It also ensures confidentiality, integrity, and availability of information to the correct users.

Cybersecurity technology has continuously delivered significant innovation to stay ahead of the constant potential for new and more ingenious attacks. From its inception in the early 1970s, when the first computer viruses emerged (like the Creeper and Reaper programs), cybersecurity has faced a continuous wave of technologies and actors whose only focus is to steal critical information. As information technology advances to create new and better solutions, so too do the threats to our critical information and infrastructures. It's a constant game of leapfrog, with extremely high stakes for all of us.

In the 1980s, the advent of personal computers led to the development of the first antivirus software, such as McAfee's pioneering products. The 1990s saw the rise of firewalls, encryption protocols, and the introduction of public-key infrastructure (PKI), which improved data protection. As the internet gained popularity, the early 2000s witnessed the spread of worms and Distributed Denial of Service (DDoS) attacks, prompting the use of intrusion detection and prevention systems.

The 2010s brought advancements in AI-based threat detection, machine learning, and behavioral analysis to combat increasingly sophisticated cyber threats. Additionally, the growing adoption of cloud computing and mobile devices fueled innovations in cloud security and mobile security solutions.

Specific recent technological advancements in cybersecurity include:

- Artificial intelligence (AI) and machine learning are increasingly applied in cybersecurity to enhance threat detection and response
- Zero Trust security models to provide more granular access controls, reducing the risk of unauthorized access to critical resources
- Blockchain technology is being used for creating secure and tamper-proof audit logs, manag-

ing digital identities, and enabling secure data sharing across multiple parties without the need for a central authority

- Hardware-based security measures, such as hardware security modules (HSMs) and secure enclaves (e.g., Intel SGX), create isolated environments that are resistant to external tampering and attacks
- The growing adoption of cloud services, various tools for cloud-native security monitoring, data encryption, and identity and access management (IAM)

Today, cybersecurity faces a renewed and critical demand to evolve thanks to threats from quantum computing, artificial intelligence, and the ubiquitous deployment of devices thanks to the Internet of Things (IoT). Today's focus must be on advancements in quantum-resistant cryptography, secure IoT implementations, and bolstering defenses against AI-powered attacks. This article places heavy focus on the pros and cons of quantum computing in the cybersecurity space, while touching briefly on other mentioned innovations and threat factors.

Enter the Latest and Greatest Threat: Quantum Computing

Most of our encryption techniques feature mathematical computations that take classical computers a very long time to compute if they can solve them at all. Specifically, most encryption leverages factorization applied to complex security keys.

In simple math terms, imagine multiplying two 10-digit numbers by each other. You can quickly calculate that answer with a classical computer. When you attempt to reverse the computation to identify the two original numbers, the classical computer falters. It can't finish the computation in any reasonable time frame.

A quantum computer can easily factor the prime numbers and break the key.

How? Quantum computing leverages the principles of quantum mechanics to perform computations in a fundamentally different way than classical computing. While classical computers use bits (0s and 1s) as the fundamental unit of data, quantum computers use quantum bits, or qubits.

Qubits can exist in a superposition of states, meaning they can represent both 0 and 1 simultaneously. This allows quantum computers to perform complex calculations exponentially faster than classical computers for specific problems. Quantum computers also leverage entanglement, a phenomenon where the state of one qubit is correlated with the state of another, even if they are physically separated (Refer to **Further Readings** for more information).

Consequently, quantum computers can solve complex problems using quantum algorithms, like Shor's algorithm, (See Page 21) for factoring large numbers (relevant for breaking certain cryptographic algorithms), and Grover's algorithm, for searching unsorted databases with a quadratic speedup. Grover's algorithm is depicted in **Figure 1.**

Quantum computing could also attack AI-driven security systems, such as breaking encryption-protecting AI models, manipulating data or models, or exploiting vulnerabilities specific to these AI security architectures.

What are the top potential weaknesses in cybersecurity with the advent of quantum computing and AI technologies?

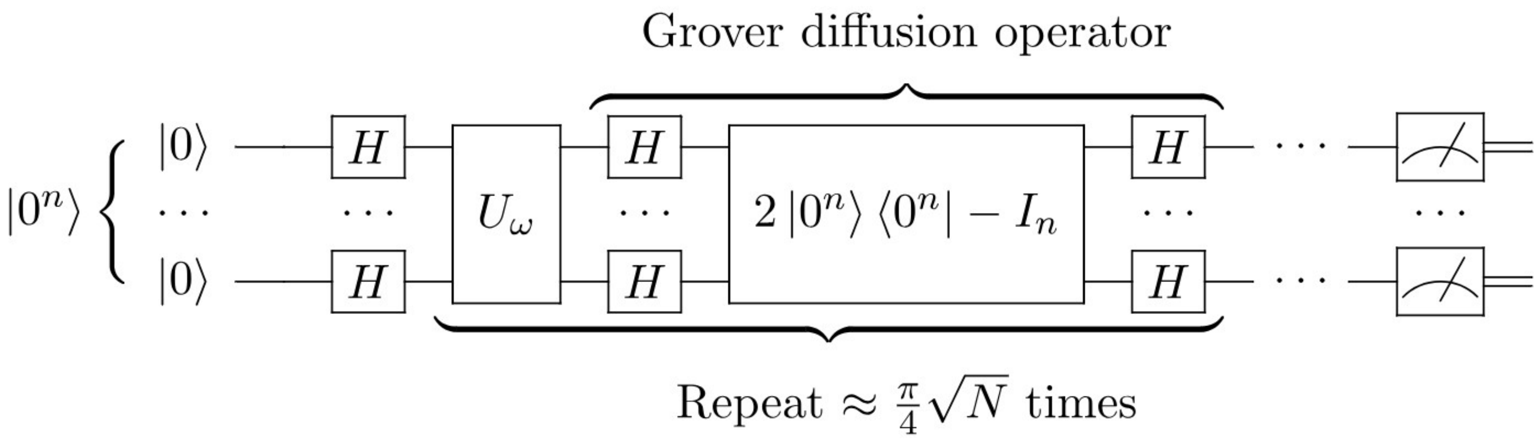

Grover diffusion operator
$|0^n\rangle$
$|0\rangle$
$|0\rangle$
H
U_ω
$2|0^n\rangle\langle 0^n| - I_n$
Repeat $\approx \frac{\pi}{4}\sqrt{N}$ times

Figure 1. *Grover's Algorithm [1].*

Cryptographic: Quantum computing has the potential to break many widely used cryptographic algorithms that currently underpin the security of digital communication and data storage, such as RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography). ECC is considered more secure than RSA, because RSA is based on factoring large numbers, a problem that classical computers can potentially solve. In contrast, ECC is based on the discrete logarithm problem, which is much harder to solve.

Still, algorithms like RSA and ECC are vulnerable to attacks from quantum computers using Shor's algorithm. Such attacks would compromise the confidentiality and integrity of protected information.

Password Cracking and Brute Force Attacks: Quantum computing can significantly accelerate password cracking and brute force attacks, simply due to their dimensional and rapid computational power. Indeed, traditional encryption methods, based on long and complex passwords, might become obsolete. A quantum computer will quickly find access.

The best password crackers often attempt to guess, using more likely passwords first as well. Thus, a quantum-enabled computer that can predict human passwords based on previous behavior would be expected to crack password hashes, as well as manual passwords.

AI-powered Attacks: AI algorithms are everywhere. They underpin nearly all autonomous and robotic systems deployed in security applications. This includes facial recognition, biometrics, drones, and autonomous vehicles used in combat surveillance and military targeting applications.

Unfortunately, AI advancements can also be used against AI-driven security. As AI and machine learning technologies advance, they can be used to enhance the sophistication of cyber-attacks, fueled by the computational power of quantum computing. For example, AI-powered malware and phishing attacks can become more convincing and difficult to detect by traditional security systems, increasing the success rate of such attacks.

Data Integrity Issues: With quantum computing, there is a risk of data integrity being compromised. Verifying the authenticity and integrity of data may become challenging as quantum computers could manipulate data in ways that are hard to detect using classical methods.

IoT Security: The Internet of Things (IoT) has grown rapidly, and many IoT devices lack robust security measures. Quantum computing and AI could be used to exploit vulnerabilities in IoT devices, potentially leading to large-scale attacks on critical infrastructure or privacy breaches.

Harvest Now, Decrypt Later (HNDL): Threat actors can collect encrypted data from organizations now, with the expectation that such data can be decrypted later when quantum computing reaches a maturity level capable of breaking current encryption. HNDL poses a risk to all organizations. Such attacks have most likely already taken place, and the targeted organizations lack the sophisticated capabilities to detect them. HNDL attacks are an inevitable threat to all enterprises in a post-quantum world.

Cybersecurity and computing researchers and experts are actively working on developing post-quantum cryptographic algorithms to address these vulnerabilities. Moreover, AI and machine learning are also being leveraged to improve cybersecurity defenses, using advanced anomaly detection and behavioral analysis techniques.

The good news is that quantum computers can also advance cybersecurity, offering defenses against advanced threats. For example, quantum cryptography leverages the power of quantum particles to create unbreakable encryption keys. Quantum error correction algorithms, when they become available at scale, will be able to detect and correct threats, minimizing the risk of data corruption.

That said, the cybersecurity landscape will continue to evolve dynamically as it always has. It's essential to stay up to date with the latest advancements and vulnerabilities to build resilient systems.

The Quantum Threat to Cybersecurity: An Example

To dive into how a quantum algorithm can break security, let us look at a specific and well-publicized threat in action: Shor's algorithm. This algorithm is designed to efficiently factorize large integers. It is one of the most well-known and significant quantum algorithms, as it demonstrates a substantial speedup over classical algorithms for solving this specific problem.

The factorization of large integers into their prime factors is a crucial task in classical cryptography. Many cryptographic algorithms, such as RSA, rely on the computational difficulty of factoring large numbers for their security. However, Shor's algorithm, when executed on a sufficiently powerful quantum computer, can efficiently factorize large integers, rendering traditional factorization-based cryptographic schemes vulnerable to attacks.

Quantum Fourier Transform: Shor's algorithm leverages a quantum-capable version of the Fourier transform called the Quantum Fourier Transform (QFT). Designed to leverage quantum performance, the QFT efficiently finds periodicity in quantum states.

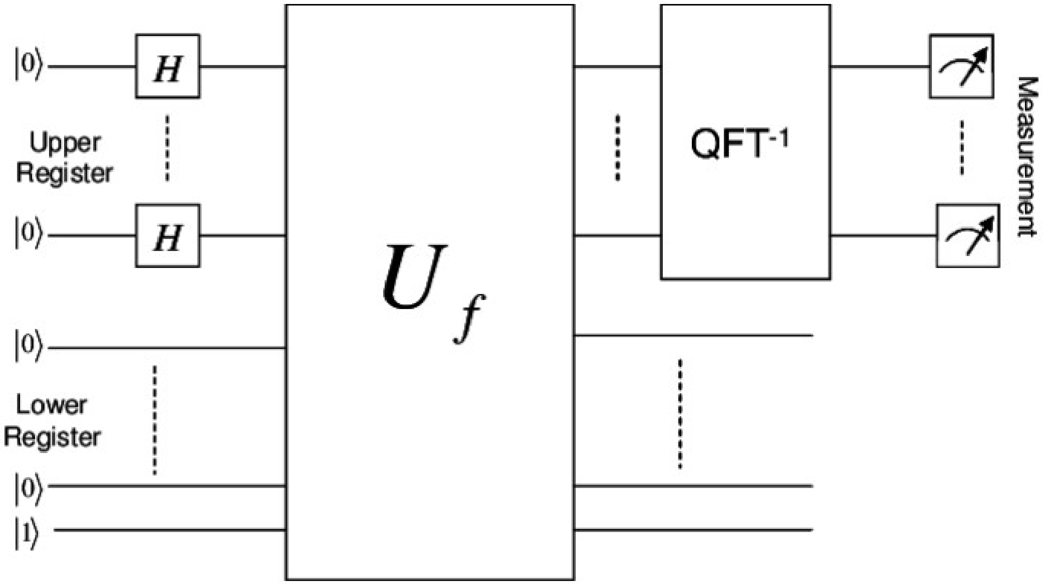


Figure 2. *Shor's Algorithm. In this figure, the upper register consists of 2n qubits and holds the superposition of integers 0. N 2 −1; lower register consists of n qubits and holds the superposition of values a x mod N after computed by U f block. Classical postprocessing of the measurement in the computational basis after the QFT block gives with high probability the period of the function f (x) = a x mod N [2].*

Quantum Period Finding: The core idea of Shor's algorithm lies in its ability to find the period of a modular exponentiation function using the Quantum Period Finding subroutine. The function takes an input (x) and computes the value f(x) = a^x mod N, where a is a random integer less than N, and N is the number to be factorized.

Quantum Superposition: As noted, quantum computing takes advantage of superposition, which allows qubits to exist in multiple states simultaneously. Shor's takes advantage of superposition, to evaluate f(x) for a range of inputs in parallel, dramatically accelerating its ability to break today's encryption.

Period Detection: The Quantum Period Finding subroutine efficiently finds the period r of the function f(x) = a^x mod N. The period r is crucial because if r is even, then (a^(r/2) - 1) and (a^(r/2) + 1) are non-trivial factors of N. If r is odd, the algorithm repeats the process with a different random a until an even period is found.

Classical Post-Processing: After obtaining the period r, classical post-processing is used to compute the factors of N using the factors derived from (a^(r/2) - 1) and (a^(r/2) + 1).

It's important to note that Shor's algorithm requires a fully functioning, error-corrected quantum computer with enough qubits to achieve its theoretical exponential speedup. As of today, practical, large-scale quantum computers remain a significant technological challenge. However, the potential impact of Shor's algorithm on public-key cryptography has driven extensive research into post-quantum cryptographic algorithms that are resistant to quantum attacks.

What are Quantum-Resistant Algorithms?

As quantum computing technologies progress, traditional cryptographic algorithms, which rely on the computational difficulty of certain mathematical problems, may become vulnerable to quantum attacks. Quantum-resistant algorithms aim to provide a level of security that remains effective even in the presence of powerful quantum computers.

Quantum-resistant algorithms, also known as post-quantum algorithms or quantum-safe algorithms, are cryptographic algorithms designed to be secure against attacks from quantum computers.

The following **Table 1** shares examples of the types and specifics of current quantum-resistant algorithms:

Table 1. *Examples of quantum-resistant algorithms.*

Algorithm	Technology	Description
Hash-Based Algorithms	Rely on cryptographic hash functions to provide security against quantum attacks.	Some cryptographic hash functions based on number-theoretical problems, can be exponentially broken with a quantum computer. Knowledge of the secret pair can invert, or break, the hash. A quantum computer could run Shor's algorithm to detect that pair, opening the door to security breaches. That said, most common cryptographic hash functions, such as SHA 256, have only been susceptible to brute force attacks to date. Examples include the Merkle signature scheme [3] and the Lamport signature scheme [4].
Code-Based Cryptography	Employs error-correcting codes to create cryptographic schemes.	Breaking these schemes with quantum computers is believed to be computationally hard due to the underlying coding theory. Examples include the McEliece cryptosystem [5], Niederreiter cryptosystem [6], and the Stern cryptosystem [7]
Lattice-Based Cryptography	Uses the mathematical structures known as lattices to create cryptographic primitives.	Lattice-based schemes are considered quantum-resistant because quantum computers have not shown a significant advantage in solving lattice problems efficiently. The secret key for lattice-based cryptography is a set of points that are close to each other. The public key is a set of points that are far apart. Finding the secret key from the public key is difficult, even for quantum computers. The search requires some brute force review of every option. Quantum computers have the potential to process faster, but such acceleration is far smaller than for other public key cryptography. Popular lattice-based schemes include the Ring Learning with Errors (RLWE) [8] and the N-th degree Truncated Polynomial Ring Units (NTRU) encryption and signature schemes [9].

Algorithm	Technology	Description
Multivariate Quadratic Equations (MQ)	A class of public-key cryptographic schemes that use multivariate polynomials over a finite field.	Solving systems of multivariate polynomials is known to be NP-complete, thus multivariate constructions are top contenders for post-quantum cryptography standards. Examples of MQ-based schemes include the Unbalanced Oil and Vinegar (UOV) signature scheme [10] and the Rainbow signature scheme [11].
Isogeny-Based Cryptography	Built on the mathematical properties of isogenies between elliptic curves.	Isogeny-based schemes use the shortest keys of any proposed post-quantum encryption methods, even though the math behind them is complex [12]. Schemes like SIDH (Supersingular Isogeny Diffie-Hellman) [13] and SIKE (Supersingular Isogeny Key Encapsulation) [14] are quantum-resistant and used for key exchange and encryption.

Table 1. *Examples of quantum-resistant algorithms cont.*

The transition from traditional cryptographic algorithms to quantum-resistant ones is a complex process and requires careful consideration and standardization efforts to ensure widespread adoption and security in a post-quantum world.

A Post-Quantum World Demands New Standards

To protect global information and infrastructure from these new threats, global standards for cybersecurity must be defined and deployed. Standardization bodies, such as the National Institute of Standards and Technology (NIST), are leading the effort to identify and standardize quantum-resistant cryptographic algorithms.

Specifically, NIST is actively working to identify and propagate a suite of next generation standard quantum-resistant algorithms. The algorithms are designed for two main tasks for which encryption is typically used: general encryption, used to protect information exchanged across a public network; and digital signatures, used for identity authentication. All four of the algorithms were created by experts collaborating from multiple countries and institutions.

In July of 2022, NIST announced the first four quantum-resistant algorithms that will become part of the post-quantum-cryptographic standard [15][16]. The chosen algorithms are CRYSTALS-Kyber, for general encryption to access secure websites [17], and CRYSTALS-Dilithium, FALCON, and SPHINCS+ for digital signatures [18][19][20]. The post-quantum cryptographic standard, expected to be finalized around 2024, will help enterprises prepare their environments for the time when quantum computers will be powerful — and readily available — enough that they will be able to break present-day encryption.

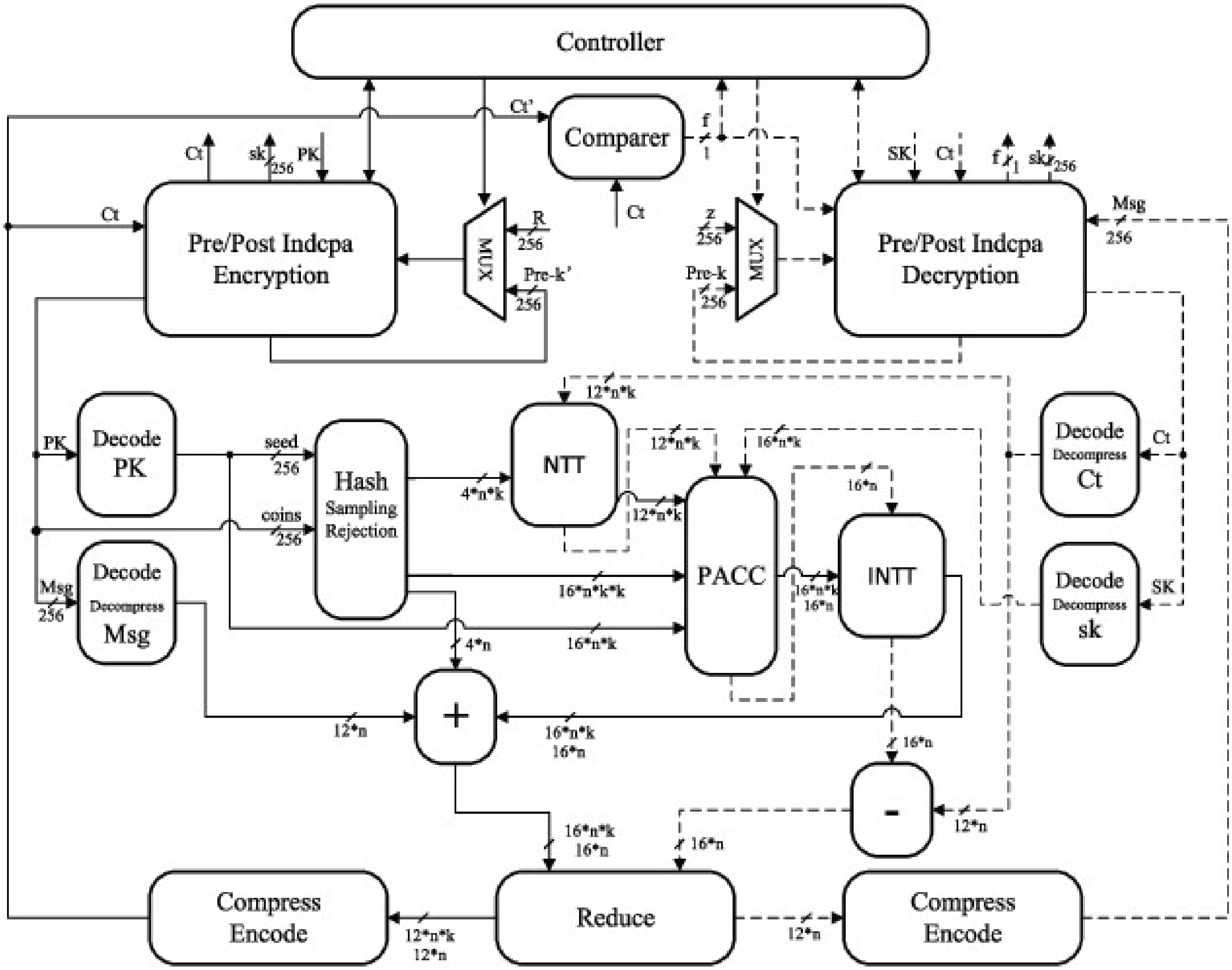
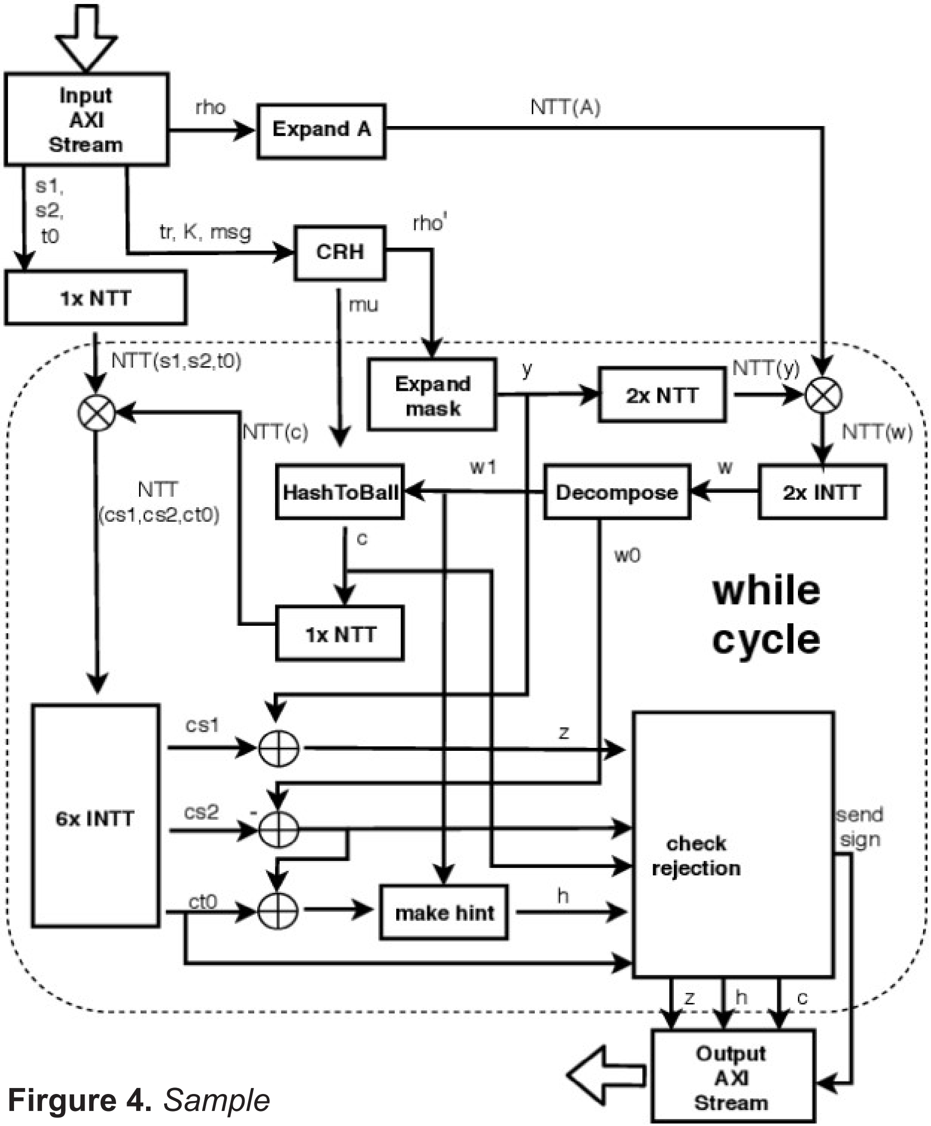


Table 3. *Sample CRYSTALS-Kyber hardware implementation [21].*

CRYSTALS-Kyber: Kyber is a key encapsulation method (KEM) designed to be resistant to cryptanalytic attacks with future powerful quantum computers. It is used to establish a shared secret between two communicating parties without an (IND-CCA2) attacker in the transmission system being able to decrypt it.

CRYSTALS-Dilithium: Dilithium is a lattice-based digital signature scheme whose security is based on the hardness of finding short vectors in lattices. The strength of a CRYSTALS-Dilithium key is represented by the size of its matrix of polynomials. For example, CRYSTALS-Dilithium (6,5) has a matrix size of 6x5. The larger the matrix size, the stronger the key. CRYSTALS-Dilithium keys can only be used for Digital Signature Generation and Verification.



Firgure 4. *Sample CRYSTALS-Dilithium implementation [22] (above).*

FALCON: FALCON is also a lattice-based post-quantum signature scheme. While Dilithium was chosen as the preferred signature scheme, due to its use of fewer different mathematical operations during key and signature generation, there are situations where Dilithium signatures may be larger than acceptable. For these situations, FALCON provides the smallest signatures and the smallest combined size of signature and public key. FALCON also provides (EUF-CMA) security.

SPHINCS+: SPHINCS+ was selected as an alternative for digital signature. Unlike other hash-based methods, it is a stateless hash-based signature scheme. Hash-based methods require that the private keys which have signed previous messages are remembered. SPHINCS+ avoids that requirement, enhancing overall security. Additionally, the

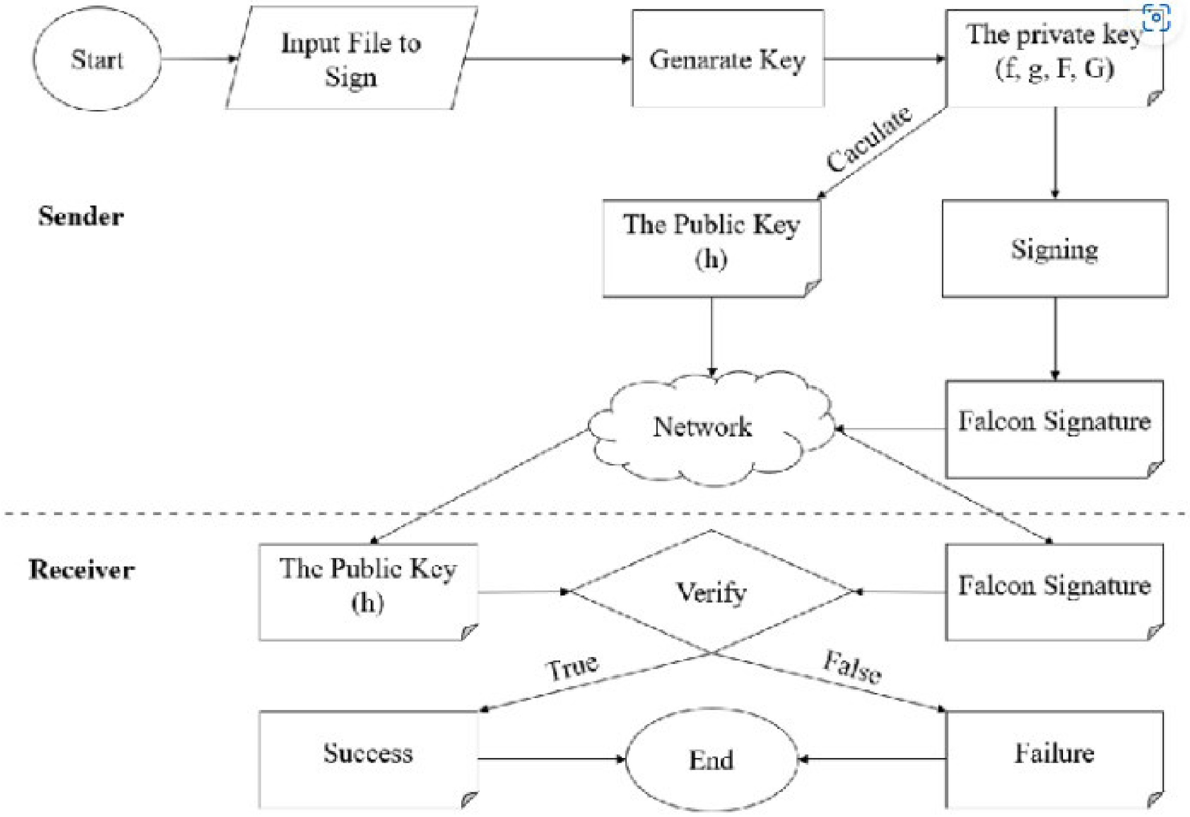


Figure 5. *Sample FALCON implementation [23] (right).*

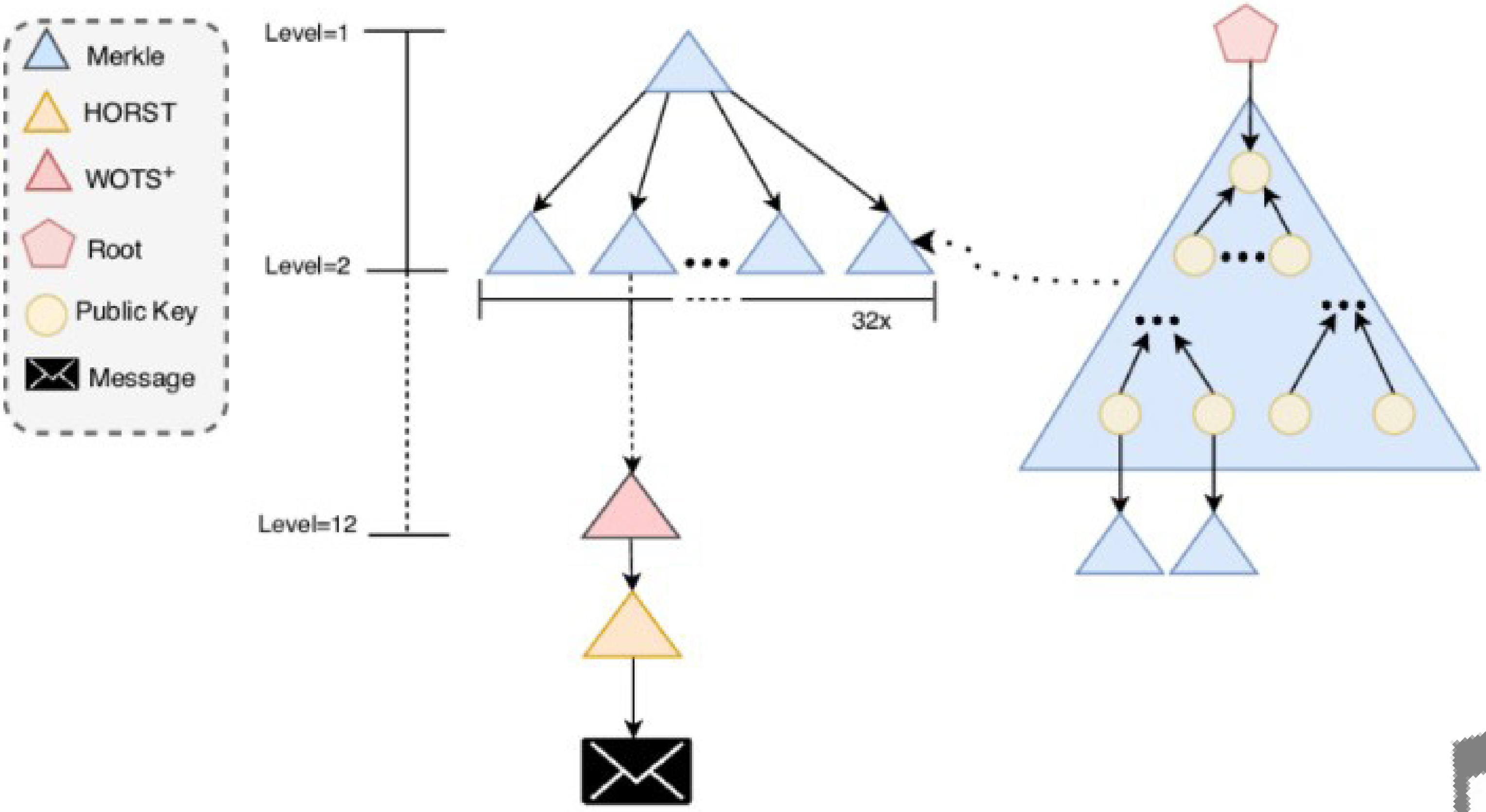


Figure 6. *Sample SPHINCS stateless hierarchical structure [24].*

relatively small size of SPHINCS+ keys offer processing advantages. For example, SPHINCS+ 256 128-bit has a public key size of 32 bytes, a private key size of 64 bytes, and a signature of 17KB. Compare that to other similar hashing methods for 128-bit, 192-bit, and 256-bit versions.

NIST will continue to solidify its quantum-resistant standards, with a final first phase expected in 2024. It's critical to all organizations that these standards be implemented. Harvest Now, Decrypt Later is already happening, and targets often do not know that the information has been penetrated. We need to be prepared for the coming threats of quantum computing. Advancements in technology are rapidly coming to market. We may not know until after the damage is done that quantum computing has breached our current security.

Key Use Cases for Cybersecurity in the U.S. Government

Governments around the world understand the critical impact of cybersecurity protections. A wide array of government data and operations include classified data and information.

Key use cases for cybersecurity in the U.S. government are shown in **Table 2** below.

Use Case	Description
National Defense and Intelligence	The U.S. government invests heavily in cybersecurity to protect national defense and intelligence assets. They use cybersecurity to secure sensitive military information, classified data, and critical infrastructure - such as power grids, communication networks, and transportation systems - from cyber threats and potential attacks by hostile entities.
Protecting Government Networks and Data	The U.S. government manages vast amounts of sensitive data, including citizen information, financial records, and diplomatic communications. Cybersecurity is crucial in safeguarding U.S. government networks and systems from cyber-attacks, data breaches, and espionage attempts.
Preventing Cybercrime	The U.S. government works to combat cybercrime and protect citizens from online threats, including cyber fraud, identity theft, and online scams. Cybersecurity initiatives involve law enforcement agencies collaborating with cybersecurity experts to investigate and apprehend cybercriminals.
Critical Infrastructure Protection	The U.S. government focuses on securing critical infrastructure, such as power plants, water supply systems, transportation networks, and healthcare facilities. Cybersecurity measures are put in place to defend against potential cyber-attacks that could disrupt essential services and cause widespread damage.
Cyber Diplomacy and International Relations	In the realm of international relations, many governments engage in cyber diplomacy to foster cooperation on cybersecurity issues, establish cyber norms, and deter malicious cyber activities. This involves bilateral and multilateral efforts to address cyber threats and promote responsible behavior in cyberspace.

Table 2. *Cybersecurity use cases for the U.S. government.*

Around the globe, governments play a significant role in funding and supporting cybersecurity research and development, promoting cybersecurity education and awareness, and collaborating with the private sector and academia to strengthen overall cybersecurity capabilities within the country. The specific use cases may vary based on each government's unique cybersecurity strategy and priorities.

What Can You Do to Protect the Future?

First, organizations must begin planning for the deployment of the new NIST standards for quantum-resistant cybersecurity and other advanced technologies as they come to play.

To prevent cybersecurity hacks and failures in the top government use cases mentioned earlier, IT departments need to be sure they are taking the following steps, regardless of the threats from AI and Quantum Computing.

Use Case	Key Actions for Cybersecurity Today
National Defense and Intelligence	• Implement strong access controls and multi-factor authentication (MFA) to protect classified information and critical systems • Regularly update and patch software and systems to address known vulnerabilities • Deploy advanced threat detection systems and conduct regular security assessments to identify potential weaknesses • Foster a culture of cybersecurity awareness and training among personnel to prevent social engineering attacks and insider threats • Establish incident response plans to respond swiftly and effectively to any security breaches or cyber incidents
Protecting Government Networks and Data	• Encrypt sensitive data both at rest and in transit to safeguard against unauthorized access • Employ network segmentation to limit the impact of breaches and contain potential threats • Conduct regular security audits and vulnerability assessments to identify and remediate weaknesses • Implement robust backup and disaster recovery strategies to ensure data integrity and availability in case of an attack or system failure • Utilize advanced endpoint protection and next-generation firewalls to monitor and control network traffic effectively
Preventing Cybercrime	• Deploy cutting-edge anti-malware solutions and intrusion prevention systems to detect and block cyber threats • Promote secure online practices among citizens, such as using strong passwords, avoiding phishing emails, and updating software regularly • Collaborate with law enforcement agencies to share threat intelligence and coordinate cybercrime investigations • Engage in public-private partnerships to tackle cybercrime collectively • Establish cybercrime response teams to handle reported incidents and support victims

Table 3. *Key actions for cybersecurity.*

Use Case	Key Actions for Cybersecurity Today
Critical Infrastructure Protection	- Conduct regular risk assessments and security audits for critical infrastructure systems - Isolate critical systems from public networks and maintain a separate network for control systems - Implement real-time monitoring and anomaly detection to identify potential threats or unusual behavior - Institute comprehensive physical security measures to protect critical infrastructure sites from physical attacks - Establish clear incident response plans, including coordination with relevant government agencies and private sector partners
Cyber Diplomacy and International Relations	- Engage in international forums to promote the development of international cyber norms and cooperation on cybersecurity issues - Strengthen partnerships with like-minded nations to share threat intelligence and collaborate on cyber defense strategies - Establish clear policies for responding to cyber incidents originating from foreign entities - Participate actively in global cybersecurity standardization efforts and promote the adoption of secure practices worldwide - Foster transparency and open communication with other nations to build trust and confidence in cyber relations

Table 3. *Key actions for cybersecurity cont.*

Table 3 shares current best practices for cybersecurity. Then there's the quantum threat. What should be the focus regarding that known issue?

1. Rethink assumptions regarding IT assets.

All authentication should be multi-factor (something you know, something you have, something you are), and leverage short-lived tokens (for "something you have") in addition to strong passphrases (for "something you know") because that forces thieves using quantum-enabled cracking to have to crack the "something you have" over and over again for any compromised identity.

Give serious consideration to adding in a third factor, like biometrics. because that introduces another order of complexity for quantum-based attacks to have to defeat.

Virtualize as many computing resources as possible (even those physically "on premise") and re-generate the virtual computers/servers at least daily - assume that they have been compromised. As part of the re-generation, vary the operating system type and version in order to vary the attack surface. This can be accomplished using Infrastructure-as-Code and tools like Terraform scripts. A constantly changing attack surface increases attack complexity (and resources required) even for quantum-enabled attackers.

Expire your data storage frequently. Every night, decrypt and re-encrypt your persistent data with new encryption keys - assume the encryption keys have been compromised. Backup encryption keys separately from data backups in a fashion where the keys are easily recovered if needed in an emergency.

Vary the cryptographic algorithms employed wherever technically feasible and look for IT infrastructure with swappable cryptography when making future purchases. This increases the complexity of your attack surface for adversaries and attackers.

2. Distribute information assets.

Common wisdom leads organizations to centralize their information assets because, in theory, that makes it easier to manage. But that presents a singular target for adversaries, and if compromised, readily leads to the loss of all the information (due to ransoming, storage failures, etc.).

Instead of moving all data & information to a central storage location for processing, deliberately plan to spread your processing out closer to where the data is generated. Modern computing facilitates this type of “distributed processing.”

The distribution can be geographically based, organizationally based, customer demographically based, etc.

By spreading out the processing of data, the size of the attack surface increases geometrically, and mitigates the scope of any successful attack. AI-enabled attacks and the use of quantum computing to crack cryptography both get geometrically more challenging for adversaries when an attack surface is spread out.

This strategy is very similar to Starlink’s business strategy. Instead of putting a few, very expensive, large satellites into orbit (like most of the existing telecommunications companies have done), which make easy targets for nation-state adversaries, Starlink launches large numbers of small, disposable satellites. It is orders of magnitude more difficult for an adversary to jam sufficient numbers of Starlink satellites’ transmissions, or to leave space debris in the path of sufficient numbers of Starlink satellites in order for them to effectively degrade Starlink’s telecommunications services.

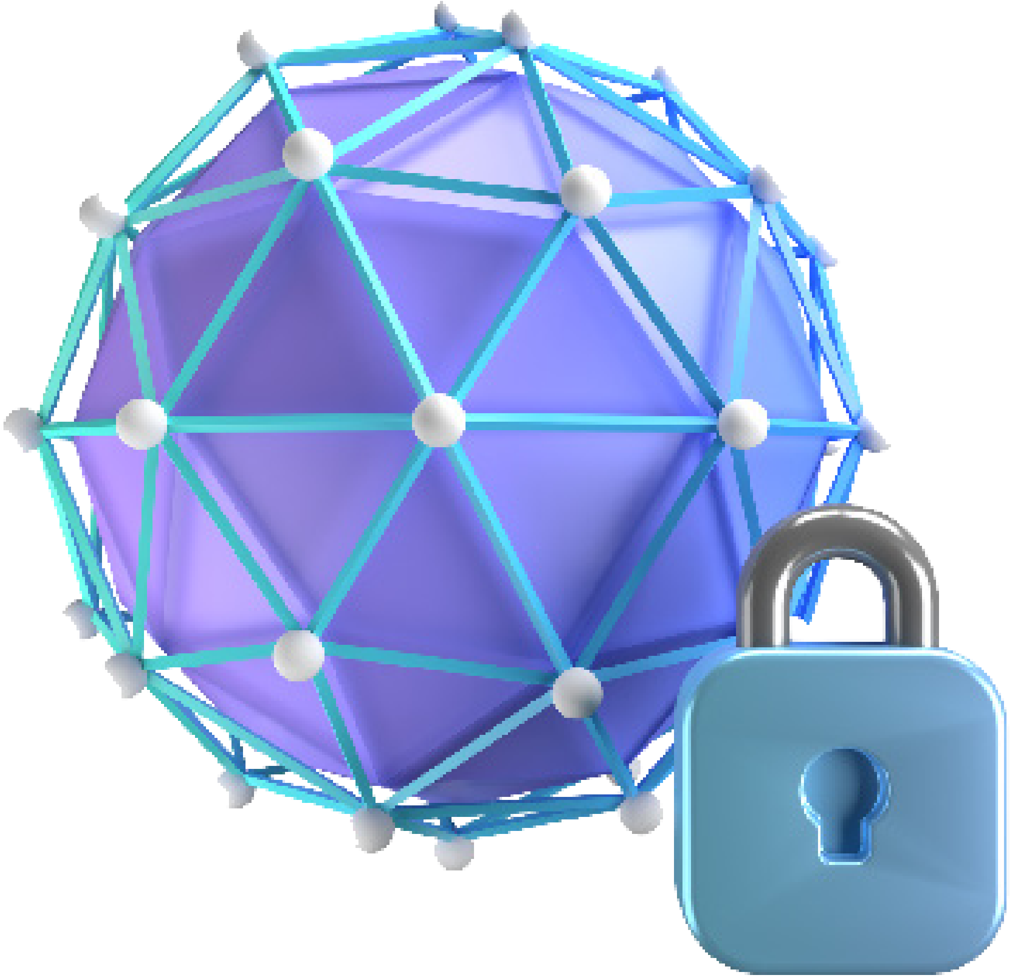
3. Plan to adopt Quantum Teleportation in network infrastructure.

While quantum computing has grabbed most of the IT-related and security-related press over the past decade, advancements in the field of quantum teleportation of information have been making relatively quiet, but steady progress.

It may sound like science fiction, but it is not. And it is closer to becoming a reality than quantum computing, because the physics involved in the commercialization of quantum teleportation are far less challenging.

With quantum teleportation, information transmissions are instantaneous and undetectable to other parties, which means they are also inherently secure (e.g. confidentiality and integrity).

Quantum teleportation will soon become a viable medium for “physical transmission” (ISO Layer 1) of



network packets. And all the existing security technologies above ISO Layer 1 will still be available for additional security protections.

Initially, while quantum teleportation only supports small information transmissions, it may be only secrets and shared crypto keys that get transmitted via quantum teleportation for economic reasons. But as the technology evolves to support cheaper implementations and larger data transmission volumes, quantum teleportation will be used for full data transmissions, replacing archaic physical mediums (e.g. copper, fiber, wi-fi) by comparison.

As the network infrastructure continues to evolve, plan to adopt quantum teleportation for the longest transmission network connections first (likely due to initial pricing of the new technology offerings). This typically will involve communications that either touch/transit the Internet or that transit physical communication channels to which access is not directly controlled (e.g., leased lines, satellite uplinks, etc.) - in other words, the external physical network attack surface.

And finally, do not presume that TCP/IP will be the network protocol stack most suitable for transmissions over “quantum teleportation”-based network segments. The TCP/IP protocols were based upon assumptions regarding network communication mediums that will no longer be valid or relevant with the introduction of quantum teleportation.

Plan to adopt new network protocols (that do not yet exist) and non-TCP/IP protocols that employ less packet overhead, less routing processing, and substantially faster (and more reliable) throughputs overall as offered by quantum teleportation.

Finally, staff must be trained on the advancements in **quantum teleportation** information as soon as possible.

Overall, a comprehensive and proactive cybersecurity strategy that includes a combination of technical measures, employee training, collaboration with partners, and clear incident response plans is essential to safeguard government systems and critical infrastructure from cyber threats and potential failures. Regular assessment, continuous improvement, and staying up to date with emerging threats are critical aspects of maintaining robust cybersecurity in the face of evolving challenges.

Conclusion

The potential impact of quantum computing on today's cybersecurity is significant and potentially world changing.

We all know that quantum computing has significant potential to shift our computational and predictive capabilities.

This power also presents a serious threat to cybersecurity, requiring a change in how we encrypt our data. Even though quantum computers aren't yet technically scalable or powerful enough to break most of our current encryption, it's critical to change our encryption methods to stay ahead of the coming threat. We need quantum-proof solutions now, especially given the threats of harvest now, decrypt later techniques. We simply do not know when our encryption will succumb. If we wait until those powerful quantum computers start breaking our encryption, it will be too late to protect critical information and infrastructure.

The powerful news is that global governments, research institutions, universities, and quantum computing leaders are focused on creating quantum-resistant algorithms that can and will revolutionize cybersecurity techniques, making them resilient in the face of ever more powerful attacks.

Additional Readings

Brown, Rebel. "Quantum Computing: Redefining Technology, Science, & Information." Crosstalk: The Journal of Defense Software Engineering, August 2023.

References

[1] Wikipedia contributors. "Grover's Algorithm." Wikipedia, Aug. 2023, en.wikipedia.org/wiki/Grover%27s_algorithm.

[2] "Fig. 1. High Level Diagram of Shor's Algorithm. Upper Register Consists..." ResearchGate, www.researchgate.net/figure/High-level-diagram-of-Shors-algorithm-Upper-register-consists-of-2n-qubits-and-holds_fig1_228102587.

[3] Komal. "Merkle Signature Scheme." Coding Ninjas Studio, www.codingninjas.com/studio/library/merkle-signature-scheme.

[4] Mishra, Ayush. "Lamport Signature Scheme." Coding Ninjas Studio, www.codingninjas.com/studio/library/lamport-signature-scheme.

[5] Classic McEliece: Intro. classic.mceliece.org.

[6] Sendrier, Nicolas. "Niederreiter Encryption Scheme." Springer eBooks, 2011, pp. 842–43. https://doi.org/10.1007/978-1-4419-5906-5_385.

[7] Wiki, Contributors to Crypto. "Naccache–Stern Cryptosystem." Crypto Wiki, cryptography.fandom.com/wiki/Naccache%E2%80%93Stern_cryptosystem.

[8] Pedrouzo-Ulloa, Alberto, et al. "Revisiting Multivariate Ring Learning With Errors and Its Applications on Lattice-Based Cryptography." Mathematics, vol. 9, no. 8, Multidisciplinary Digital Publishing Institute, Apr. 2021, p. 858. https://doi.org/10.3390/math9080858.

[9] Hoffstein, Jeffrey, et al. “NTRU: A Ring-Based Public Key Cryptosystem.” Disa. https://safe.menlosecurity.com/doc/docview/viewer/docN8E4631E1449D83b2bb2aa82b41ff3bcb74d6ba2b04095bffcc84f942c62a52838685adb534da.

[10] Kipnis, Aviad, et al. “Unbalanced Oil and Vinegar Signature Schemes.” ResearchGate, Jan. 2000, www.researchgate.net/publication/2577687_Unbalanced_Oil_and_Vinegar_Signature_Schemes.

[11] “Rainbow Signature: One of the Three NIST Post-quantum Signature Finalists.” PQCRainbow. www.pqcrainbow.org.

[12] Velon, Javier Silva. “Zero-knowledge proofs and isogeny-based cryptosystem.” Disa. https://safe.menlosecurity.com/doc/docview/viewer/docN21AD73C75F88ddfa158b19506f1c9d227247bc3d9fd68db485f44880d2c212885f7d8a60423d.

[13] Jao, David, et al. “Supersingular Isogeny Key Encapsulation.” CSRC. csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/SIKE-spec.pdfhttps://safe.menlosecurity.com/doc/docview/viewer/docN21AD73C75F88ddfa158b19506f1c9d227247bc3d9fd68db485f44880d2c212885f7d8a60423d.

[14] “SIKE – Supersingular Isogeny Key Encapsulation.” SIKE – Supersingular Isogeny Key Encapsulation, sike.org.

[15] “NIST Announces First Four Quantum-Resistant Cryptographic Algorithms | NIST.” NIST, July 2022, www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms.

[16] Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce. Post-Quantum Cryptography Standardization - Post-Quantum Cryptography | CSRC | CSRC. csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

[17] Kyber. pq-crystals.org/kyber/index.shtml.

[18] Dilithium. pq-crystals.org/dilithium/index.shtml.

[19] Falcon. falcon-sign.info.

[20] SPHINCS+. sphincs.org.

[21] Huang, Yiming. “A Pure Hardware Implementation of CRYSTALS-KYBER PQC Algorithm Through Resource Reuse.” 2020, www.semanticscholar.org/paper/A-pure-hardware-implementation-of-CRYSTALS-KYBER-Huang-Huang/387e2a3ad2e8fc0f17627d964e100e2aa9e3994d.

[22] Ricci, Sara. “Implementing CRYSTALS-Dilithium Signature Scheme on FPGAs.” 2021, www.semanticscholar.org/paper/Implementing-CRYSTALS-Dilithium-Signature-Scheme-on-Ricci-Malina/947af37669495baca8a7f96b12dacdd0d9fee604.

[23] “Fig. 1. Falcon Post-quantum Digital Signature Operation Model.” ResearchGate, www.researchgate.net/figure/Falcon-Post-quantum-digital-signature-operation-model_fig1_369739938.

[24] “Fig. 3: Hypertree Structure Used in SPHINCS: An Illustration Of...” ResearchGate, www.researchgate.net/figure/Hypertree-structure-used-in-SPHINCS-An-illustration-of-stateless-Hierarchical-Signature_fig2_340859654.

About the Author

As a recognized technology strategist, Rebel Brown guides companies to profitably define, launch, and grow their bottom lines. She is a go-to-market expert whose strategies, positioning, and launches have led to dramatic and successful results for over 75 startups and 300 high-tech and complex B2B technology companies globally. Her current passions include quantum computing, artificial intelligence (AI/XAI) and machine learning (ML).

Ms. Rebel Brown

CEO

Cognoscenti, Inc.

rebel@rebelbrown.com

Steve Foote is an accomplished software engineering executive, designing and implementing enterprise applications and development teams which provide a competitive advantage for his clients. As the Director of MITRE's Software Engineering Technology Center comprised of 600+ software architects, engineers, and scientists, Steve applied his knowledge and experience in agile software engineering, computer architecture, cyber security, mobile technologies, web services, and enterprise applications to a many Departments and Agencies within the United States' Federal Government.

Mr. Steve Foote

CTO

Phenomentai

steve@phenomenati.com

LESSONS LEARNED FROM CLOUD ONE MIGRATION

Joseph S. Moore IV
Cost Analyst
United States Air Force

Brandon M. Lucas
Assistant Professor
AFIT, United States Air Force

Robert D. Fass
Assistant Professor
AFIT, United States Air Force

Jonathan D. Ritschel
Associate Professor
AFIT, United States Air Force

Vincent J. Papia
Technical Director
PEO C3I&N, United States Air Force

"If we fail to adapt… at the speed of relevance, then our military forces… will lose the very technical and tactical advantages we've enjoyed since World War II"

-Secretary of Defense James N. Mattis

The United States Air Force (USAF) initiated Cloud One in 2017 [1]. The impetus to migrate toward cloud computing was clear. Numerous policies, directives, and legislation from the preceding decade advocated for a cloud-first military [2][3]. The stand-up of Cloud One appears to be a watershed moment towards that goal. Since 2017, the Air Force has migrated over 100 applications to Cloud One [1]. It has become the leading USAF provider for cloud computing platforms, technologies, approaches, and solutions.

Much has been researched and published about cloud migrations and operations in the commercial sector [4][5]. However, there is a dearth of information discussing public sector adoption. More specifically, the process and performance of Cloud One has yet to be studied in detail for Air Force mission applications. Given that cloud computing is continuing to be adopted throughout the military, there are valuable lessons that can be learned from current Cloud One migration and operation experiences.

This paper seeks to illuminate those insights. We analyze personnel requirements, application technical performance, application requirements fulfillment, operational security risks, and cost performance to discern implementation results versus predicted benefits. Surveys and follow-up interviews with mission application owners are utilized to collect this information. The goal is to identify existing issues in the process and highlight possible improvements. These results can inform future efforts to migrate USAF mission applications to Cloud One.

Cloud Computing and Cloud One

Cloud computing is characterized by the delivery of on-demand computing services (e.g. storage, management, and processing of data) over the internet, rather than via local infrastructure [6]. The National Institute of Standards and Technology (NIST) states that cloud models are comprised of five essential characteristics, three service models, and four deployment models. See **Table 1**.

Essential Characteristics	Service Models	Deployment Models
On-Demand Self Service	Software as a Service (SaaS)	Private Cloud
Broad Network Access	Platform as a Service (PaaS)	Community Cloud
Resource Pooling	Infrastructure as a Service (IaaS)	Public Cloud
Rapid Elasticity		Hybrid Cloud
Measured Service		

Table 1. *NIST Cloud Model Taxonomy [6].*

Cloud One provides cloud computing options for military applications. Cloud One's mission is to provide common secure computing environments, standardized platforms, application migration and support services, and data management. In other words, Cloud One grants government applications the ability to enjoy the cloud computing benefits that are available to commercial cloud consumers. Cloud One utilizes the PaaS model, which allows the Cloud User to launch their own created or procured applications supported by the cloud provider. Cloud One's website states their PaaS model "offers an ideal balance of mission application self-management and best-practice, DISA-approved 'guardrails,' allowing your team to focus on your application, instead of spending valuable time managing hosting environment and underlying infrastructure" [1].

Cloud One provides a subset of commercial cloud services. Mission application owners have access to auto-scaling to meet demand. Cloud One offers data backup and recovery, system updates, and patch support. Application responsiveness and downtime are mitigated through load balancing of traffic while applications are monitored automatically and provide automated alerting. Each of these services is DISA-approved and may offer optimized performance at the lowest possible cost (pay-as-you-use). Cloud One also offers tailored services that are common for USAF/DoD requirements and environments. These services include compliance and accreditation, Cyber Security Service Provider (CSSP) integration, monitoring/logging, operating analytics, DevSecOps (software development, security, and information technology operations), automated security and vulnerability management, identity/access management, collaboration, and support [1].

Pros and Cons of Cloud Computing

Cloud computing offers several benefits. The web-based services provide flexibility. Applications and resources can be accessed from anywhere, at any time [7]. Collaboration is also improved via shared access to data and documents. The cloud allows real-time collaboration between organizations in disparate locations [8].

Scalability is another advantage of cloud computing. Computing resources are adjusted based upon demand levels. This can lead to cost savings [6]. Additional cost savings may occur during configuration. Cloud computing has a relatively low initial configuration fee compared to many services that require licenses [9]. Similarly, the cloud may reduce the need for customers to maintain Subject Matter Experts (SMEs), reducing customer costs [9].

Cloud computing also has disadvantages. Since customers must use the web for their services, any internet access impediments will slow down or halt access to those services. With mission-critical services, this can be a problem [7]. Similarly, cloud computing, in comparison to locally hosted software applications, is dependent day-to-day on the cloud supplier for access to the IT services. This lack of control over infrastructure can affect performance monitoring, customization, and compiance with local regulations [6].

However, perhaps the most important concern to military organizations is data protection and security, since data breaches, unauthorized access, and loss of control over sensitive information are major concerns [10]. The DoD recognizes these risks, and illuminated its response in the DoD Cloud Strategy. This document outlines the vision towards an enterprise cloud environment for the DoD, while addressing data and security concerns [11].

In summary, the existing literature identifies flexibility, collaboration, scalability, and cost savings as potential benefits of cloud computing. At the same time, it cautions that there may be access, infrastructure control, data protection, and security concerns. On balance, the commercial sector experience shows the benefits typically outweigh the disadvantages. However, the questions remain: "what lessons have the USAF Cloud One migration efforts revealed?" and "has the migration and operations provided the same net-positive results as the commercial sector?" The remainder of this article provides an empirical examination of Cloud One migrations to unveil those insights.

Study Design

Surveys and follow-up interviews with mission application representatives were our primary source of data. Five characteristics were investigated pertaining to Cloud One migration. These characteristics were chosen based on the top risks to organizations utilizing cloud computing identified by Dutta et al. [12] and Bhat et al. [13]. **Table 2** shows the five characteristics, their reason for inclusion, and the number of questions asked about each characteristic in the survey.

Characteristic	Description	Number of Questions
Personnel Requirements	Personnel counts pre- and post-migration; skillset and labor category requirements	5
Application Technical Performance	Average uptime, user satisfaction, and performance (latency, app crashing, etc.) pre- and post-migration	6
Requirements Fulfillment	Ease of requirement fulfillment pre- and post-migration; Information or changes needed to facilitate migration	4
Operational & Security Risks	Identifies operational and security risks pre- and post-migration	6
Cost	Identifies projected and realized cost impacts	5

Table 2. *Survey Characteristics.*

The Cloud One program provided a list of mission application representatives as potential participants in the study. The list of representatives included mission applications that were already migrated, currently migrating, and scheduled to migrate. An initial pool of 115 representatives were first asked to confirm their interest in participating in the study: 14 accepted, 6 denied, 95 did not reply. This initial invitation was sent twice by the research team.

The 14 representatives that accepted the invitation were then sent the survey questions. They were given three weeks to consult with their team and provide answers to the questions via e-ail. 6 of 14 representatives provided answers to the survey. Next, individual interviews were conducted with the six respondents (and their team). The respondents were comprised of both DoD civilians and military with skillsets that included software engineers, program managers, and members of their organizational leadership team. Clarifying questions from the survey and additional information to correspond with other representatives answers was collected. Any experiences that were common between more than one participant were used as a future or follow-up question to the other participants to gauge the frequency of the experience. All interviews were recorded to ensure accuracy and transcribed (Note: due to technical issues, one interview was not properly recorded; study team notes were referenced instead).

Due to the nature of the data collected, we used Grounded Theory to generate insights from the data. Grounded Theory is a strategy for systematically analyzing data in an exploratory manner for the development of theory [14]. It allows for the identification of a pattern within the data, and from that pattern, the discovery of the core category or foundation of the theory. The guiding principle is to let the data derive the theory, as opposed to fitting data to a predisposed assumption. Because Grounded Theory is a manual process, it explicitly incorporates the “human brain” in the process.

Grounded Theory’s constant comparative method involves multiple phases of coding data. This process involves assigning codes or categories to each line of data, and constantly comparing those codes to related codes across the document [14]. The process of coding continues until core categories and related concepts emerge, and all possible categories are exhausted [15]. Grounded Theory has been used successfully in many fields, including information technology [16]. We utilized it for both the initial survey and follow-up interview transcripts.

Results

Three separate analyses were conducted. The first investigated individual codes, the second group characteristics, and the last was a time-phased analysis. To facilitate these, Grounded Theory was applied to the survey and interview comments. In total, 25 different codes were mapped to 217 cells of text.

The first analysis explored the 25 individual codes data via frequency distributions. We found uptime (10.66%), security risks (9.14%), and operational risks (8.63%) to be the three most frequently used codes. Uptime was improved in 75% of the applications after migrating to Cloud One. Similarly, three of the six applications noted that security risks decreased due to Cloud One migration. However, two of the six applications indicated that security risks are a tradeoff between increased risk envelope and increased risk mitigation tools.

In addition to the high frequency codes, the individual code investigation also revealed insights in low-frequency codes. For example, the code “port issue” was only found in 3.05% of the comments. However, a consistent message from four of the six applications needing more ports opened (as opposed to being restricted to just the standard 443 port) was an important finding.

The second stage of analysis sorted the individual codes into the five characteristics (or subject matters) from **Table 2**. By utilizing this higher hierarchy of categorization, we were able to compare and analyze the same subject matter between participants. See **Table 3.**

Characteristic	Key Findings
Personnel Requirements	• Three of the six applications added personnel due to Cloud One migration. The capability adds the requirement for technical expertise that is not native to the average organization.
Application Technical Performance	• Three of the six applications noted increased uptime while one stated uptime as a negative issue. The remaining two could not provide statistics because the migration was not being fully completed. The application with uptime issues is an entirely new application. This outlier could be due to an application development issue rather than a Cloud One migration issue. • Three of the six applications note significant performance issues prior to Cloud One migration, including numerous downtimes, network instability/outage, and insufficient storage space. Cloud Migration solved two; the other could not answer due to migration in progress.
Requirements Fulfillment	• Three of six applications record requirements that were easier to fulfill before Cloud One Migration. Respondents noted better accessibility to the development environment for users, better access to servers to maintain the software, and easier data transfer between networks before migration. • Four of the six applications record requirements that are easier to fulfill after Cloud One Migration, including application update and enhancement, redundancy, remote access, and storage. Two of the three applications that report requirements that were easier to fulfill before Cloud One migration have a tradeoff after migration since they also reported requirements that were easier to fill after Cloud One Migration. • One application stated Cloud One cannot fill some of its requirements • All six applications listed information that would have better prepared the organization for migration. Applications highlight the need for training (data analytics and virtual networking), transparency on responsibility separation, and updates on migration policies.
Operational & Security Risks	• Operational risks results were mixed with three positive and three negative responses. Only two of the three applications that noted increased operational risks cited specifics. The first was access to Cloud One risks. The second risk was cited due to the disparate development and production environments. • Two applications state that migration has increased security risks and four state that migration has decreased security risks. The two applications that state increased also indicate a decrease in security risks due to a risk transfer to the user's organization. These are not new risks due to Cloud One Migration; instead, the risk may have previously been owned by an IT unit. Two of those that indicated decreased security risk cited that the new tools/options that Cloud One provides help to mitigate the risk.
Cost	• None of the applications indicated any empirical cost savings.

Table 3. *Findings from Characteristic Analysis.*

The third analysis was a time-phased examination. The migration process was divided into three time phases: pre (orange boxes), during (purple boxes), and post (blue boxes). See **Figure 1** for the topics covered by phase.

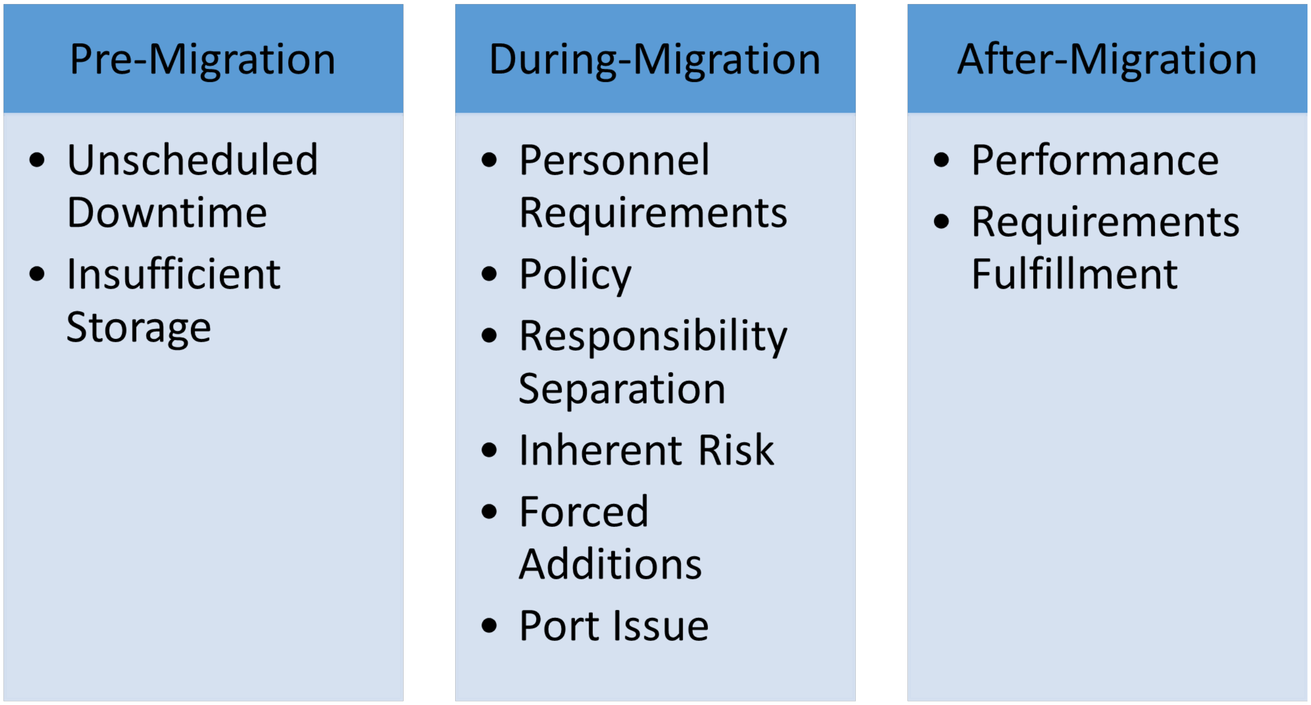


Figure 1. *Timeline of Migration Topics.*

Pre-migration was characterized by unscheduled downtime, network instability, and lack of storage. One application even cited 100+ unscheduled downtime events over two years. These pre-migration issues were motivating factors to migrate to Cloud One.

During migration several themes emerged. Adding new personnel with cloud expertise was a common need. Similarly, most organizations noted the need for more training of current personnel to perform new tasks. Revision of current policies and separation of responsibilities was also a recurrent theme. Prior to migration, the IT for Air Force organizations is often primarily controlled centrally instead of inherently within the organization that owns an application. Converting to the cloud naturally places more ownership on the organization. These concerns are echoed in mission applications requesting “more information on what mission partners are expected to handle versus previous environment owners (cybersecurity, network/comm., and system administration).”

Two additional themes emerged during migration. Risks transfer is a concern. Not only are previous risks inherited by the mission application, but so are new risks due to the new environment. The last theme is issues due to migration of apps that were not originally intended to run in the cloud. Respondents opined that Cloud One is built to host applications designed for the cloud, but many apps are not. This leads to concerns regarding “forced additions” and “port issues.”

Post-migration analysis noted improvements in requirements fulfillment such as accessibility to the development environment, better access to servers to maintain the software, and data transfer between networks. Other requirements stated as easier to fulfill after migration were application updates and enhancements, redundancy, and remote access. Performance was also improved with increased uptime, increased storage space, and better accessibility. Conversely, significant cost savings were not reported in the data.

Conclusion

"I don't need a hard disk in my computer if I can get to the server faster... carrying around these non-connected computers is byzantine by comparison."
- Steve Jobs

Cloud migration from local servers and computers is well underway in the DoD. Much like the literature on civilian sector experiences, our analysis revealed cloud migration to be a net-positive. While not all the promised benefits (e.g. cost savings) are yet to be realized in our data sample, there were numerous requirements fulfillment, security, and performance improvements realized. Despite these positive results, there are lessons to be learned for future migrations. We summarize our findings into three areas for Cloud One migration improvement: 1) organizations should hire cloud-specific personnel to assist with migration and post-migration, 2) Cloud One should collaborate early with mission applications to determine access to their particular required communication endpoints (ports) outside of their default communication endpoint (port 443), 3) and stake holders should discuss the separation of responsibilities and ownership of risk before migration.

References

[1] Cloud One Website, US Air Force (2023). https://cloudone.af.mil/#/

[2] Kundra, Vivek. "Federal Cloud Computing Strategy." The White House, 8 February 2011. https://www.dhs.gov/sites/default/files/publications /digital-strategy/federal-cloud-computing-strategy.pdf

[3] The National Defense Authorization Act for Fiscal Year 2012. S.1867, 112th Congress 1st Session. http://www.gpo.gov/fdsys/pkg /BILLS-112s1867es/pdf/BILLS-112s1867es.pdf. (Section 703-7)

[4] Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58.

[5] Qi, Wenhao, Meng Sun, and Seyed Reza Aghaseyed Hosseini. "Facilitating big-data management in modern business and organizations using cloud computing: a comprehensive study." Journal of Management & Organization (2022): 1-27.

[6] Mell, P. and Grance, T. (2011). "The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology," NIST SP 800-145, Gaithersburg, MD. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[7] Hashem, Ibrahim Abaker Targio, et al. "The rise of "big data" on cloud computing: Review and open research issues." Information systems 47 (2015): 98-115.

[8] Rittinghouse, J.W., & Ransome, J.F. (2009). Cloud Computing: Implementation, Management, and Security (1st ed.). CRC Press. https://doi.org/10.1201/9781439806814

[9] British Computing Society (BCS) (2012). Cloud Computing: Moving IT Out of the Office. BCS, The Chartered Institute for IT.

[10] Pearson, Siani, and Azzedine Benameur. “Privacy, security and trust issues arising from cloud computing.” 2010 IEEE Second International Conference on Cloud Computing Technology and Science. IEEE, 2010.

[11] Department of Defense (2018). DOD Cloud Strategy. Accessed 18 Jul 2023. https://media.defense.gov/2019/Feb/04/2002085866/-1/-1/1/DOD-CLOUD-STRATEGY.PDF

[12] Dutta, Amab, Guo Chao Alex Peng, and Alok Choudhary. “Risks in enterprise cloud computing: the perspective of IT experts.” Journal of Computer Information Systems 53.4 (2013): 39-48.

[13] Bhat, Suhas, et al. “Top Threats to Cloud Computing: Egregious Eleven Deep Drive.” Cloud Security Alliance (2020).

[14] Glaser, B. G., & Strauss, A. L. (1967). The discovery of grounded theory: Strategies for qualitative research. Hawthorne, NY: Aldine Publishing Company.

[15] Holton, J. A. “The coding process and its challenges.” Grounded Theory Review 9.1 (2010): 21-40.

[16] Wiesche, Manuel, et al. “Grounded theory methodology in information systems research.” MIS quarterly 41.3 (2017): 685-A9.

About the Authors

Captain Joseph Moore, USAF, is a recent graduate of the Air Force Institute of Technology (AFIT) where he earned a Master’s of Science in Cost Analysis. He is currently assigned as a cost analyst for the Air Force Cost Analyst Agency

Captain Joseph Moore

Cost Analyst

Joint Base Andrews, MD

joseph.moore.53@us.af.mil

Dr. Brandon M. Lucas is an Assistant Professor of Systems Integration & Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BA in History from the University of Texas at Austin, a MA in International Relations and ME in Teacher Education from the University of Oklahoma, a MS in Cost Analysis from AFIT, and a PhD in Economics from George Mason University. Dr. Lucas' research interests include profit analysis, cost & economic analyses, and incentive structures.

Dr. Brandon M. Lucas

Assistant Professor

AFIT, Wright Patterson AFB

Brandon.Lucas@afit.edu

Dr. Jonathan D. Ritschel is an Associate Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, MS in Cost Analysis from AFIT, and PhD in Economics from George Mason University. His research interests include public choice, the effects of acquisition reforms on cost growth in DoD weapon systems, and economic institutional analysis.

Dr. Jonathan D. Ritschel

Associate Proffessor

AFIT, Wright Patterson AFB

Jonathan.Ritschel@afit.edu

Dr. Robert D. Fass is Assistant Professor of Systems Integration and Cost Analysis, Department of Systems Engineering and Management, AFIT. He has received his BA Economics; MBA; and PhD, Business Administration and Management, all from New Mexico State University. Dr. Fass's research interests include: cost analysis, decision analysis, risk analysis, operations research, behavioral economics, organizational behavior, organizational change, and government acquisition policy.

Dr. Robert D. Fass

Assistant Professor

AFIT, Wright Patterson AFB

robert.fass@afit.edu

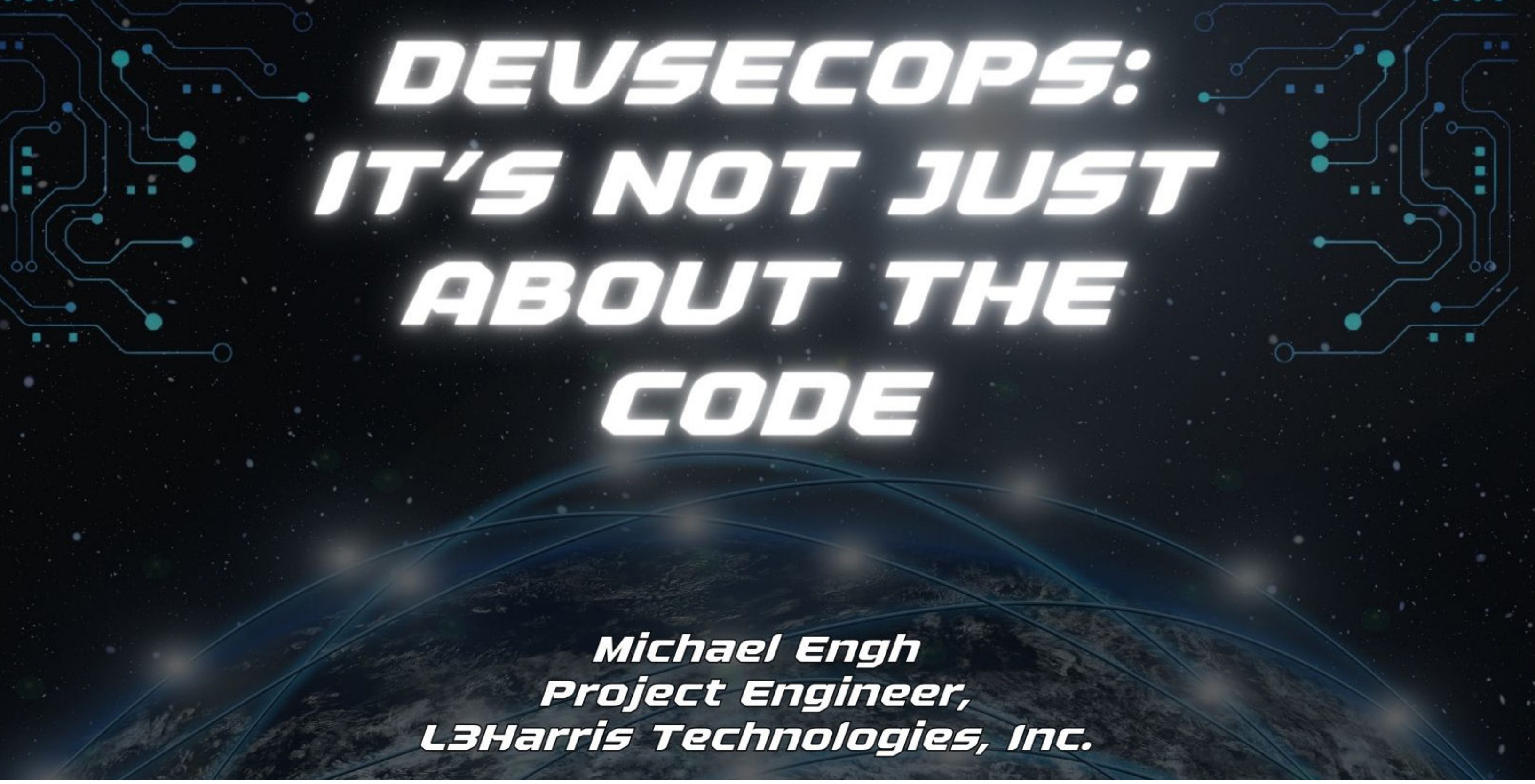
Vinny Papia is currently the Technical Director for Cost Analysis at the Command, Control, Communication, and Intelligence Networks Directorate (PEO C3I&N). There, he advises cost analysts and program managers on programmatic and technical details of enterprise IT programs. During his time there, he was the lead estimator for multiple Enterprise IT programs, including Air Force Cloud One, where he received an in-depth education on the idiosyncrasies of cloud computing and its role in the DoD. He has a Bachelor's degree in Mathematical Sciences and a Master's degree in Data Science, both from the Worcester Polytechnic Institute.

Mr. Vincent Papia

Technical Director

PEO C3I&N

vincent.papia.1@us.af.mil



DEVSECOPS: IT'S NOT JUST ABOUT THE CODE

Michael Engh
Project Engineer,
L3Harris Technologies, Inc.

Abstract

Development, Security, and Operations (DevSecOps) is meant to lead software programs to new levels of efficiency and productivity, but software teams are not the only ones who must adopt agile. While software teams use DevSecOps for their operating procedures, all other supporting organizations must come to understand how they play a part in the success of DevSecOps, even if they don't adopt DevSecOps for their own business practices. Without that understanding, supporting organizations may end up being a roadblock instead of a supporting agency.

Introduction

DevSecOps has been touted as the agile methodology that will revolutionize software development within the Department of Defense (DoD). In 2019, the Office of the Chief Information Officer, Mr. Thomas Lam, Acting Director, in collaboration with Mr. Nicolas Chaillan, Special Advisor for Cloud Security and DevSecOps from the Office of the Undersecretary of Acquisition and Sustainment (A&S), released a document entitled, “DoD Enterprise DevSecOps Reference Design” [1] that outlined the merits of DevSecOps and what it means to develop software in a DevSecOps environment. The document details what tools should be used, how they are accessed, deployment templates to the program applications, and defines what all the aspects of DevSecOps look like. The document is well outlined and defines the expectations for a software group looking to adopt DevSecOps for their operating procedures and how to define successful implementation. This document has been used over and over by groups throughout the DoD to establish DevSecOps software factories with varying degrees of success. The basic premise and direction dictate how software development teams are to implement DevSecOps agile methodologies. The premise for this paper is to address how all the teams that support software development, referred to as “enabling entities,” play an integral role in the successful implementation of a DevSecOps environment.

DevSecOps Defined

DevSecOps Lifecycle

Figure 1 below shows the DoD defined Enterprise DevSecOps Software Lifecycle. The development cycle for software is represented on the left. The operations cycle is represented on the right. The security part of the process envelops the entire cycle as security of the process, development, release, deployment, and operational use of software products must be accounted for to maintain the competitive edge and avoid tampering in any phase of the process.

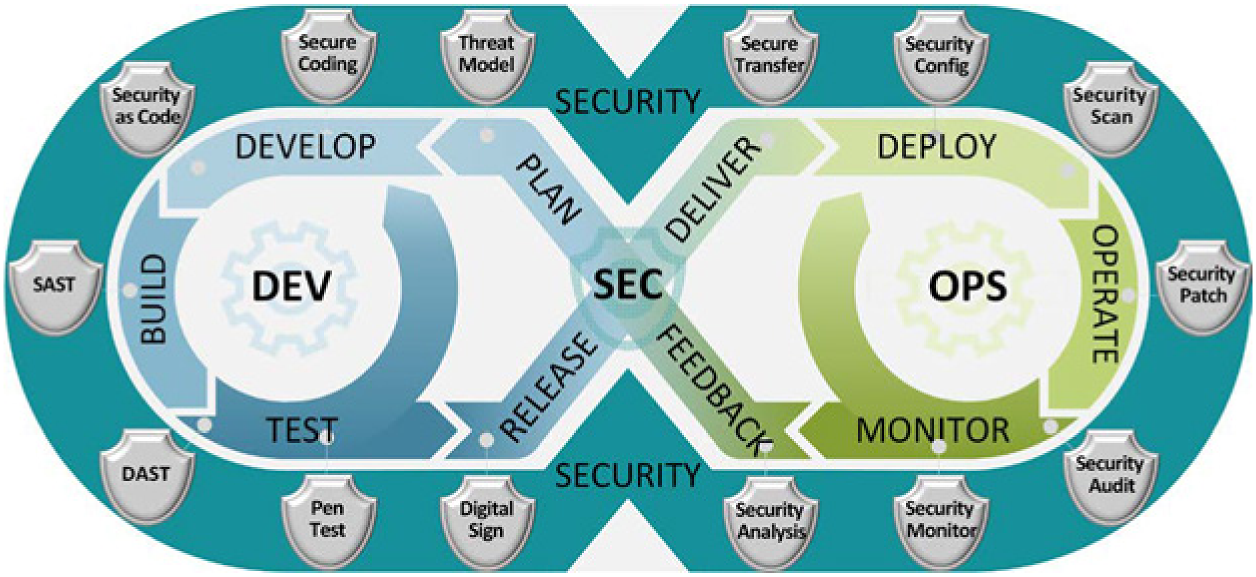


Figure 1. *DevSecOps Software Lifecycle.*

The tenets of DevSecOps are sound and, when implemented carefully and intentionally, a software development team can implement software releases more often, with higher quality, and with advanced levels of cybersecurity built in. This DevSecOps framework has been successfully implemented and demonstrated by numerous software organizations and teams throughout the DoD, including teams in the 309th Software Engineering Group (SWEG).

DevSecOps in 309 SWEG

Various teams within 309 SWEG have tested and implemented, with varying degrees of success, DevSecOps business practices. One of the most successful implementations of DevSecOps is demonstrated by the Personnel Recovery Command and Control (PRC2) team in the 309 SWEG's 517th Software Engineering Squadron (SWES). PRC2 was able to implement cloud-based development environments complete with a software factory running on pipelines using fully automated testing. PRC2 customers have visibility of the production processes and are embedded within the development team to provide immediate user feedback and requirements refinement.

Software development teams do not operate in a vacuum. **Figure 1** is assumed to be a software team working at peak performance with fully operational tools, resources, and end users at their fingertips. The DoD Enterprise DevSecOps Reference Design outlines the full process with several assumptions, but the underlying understanding is that the tools function properly, networks and servers never go down or become unresponsive, and all enabling entities are operating with the mission to support the DevSecOps lifecycle. In real life, meeting all these conditions continually is quite difficult.

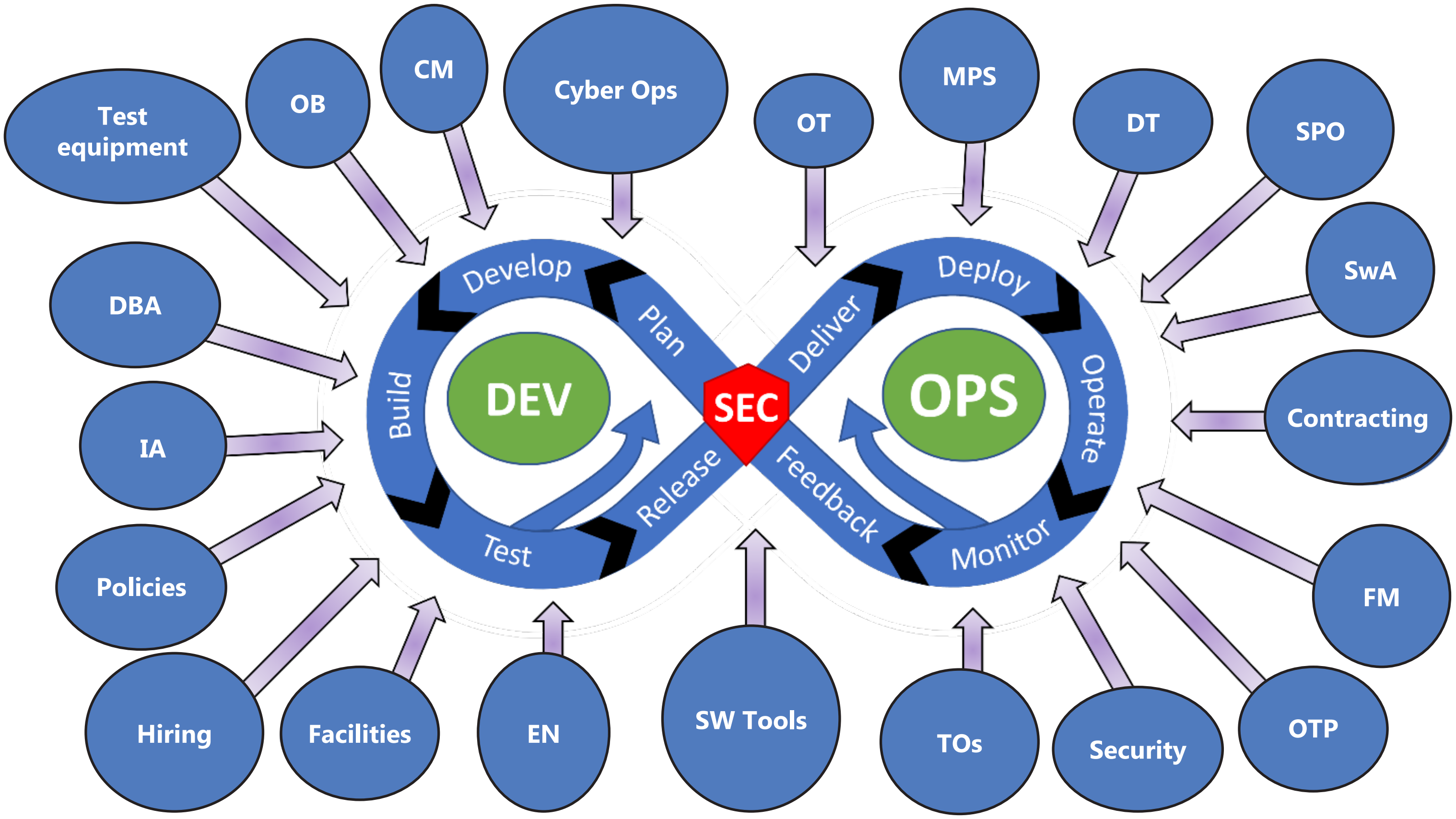


Figure 2. *A-10 OFP DevSecOps Enabling Entities Support System. For more information, See Table 1 below.*

Acronym	Definition
CM	Configuration Management
DBA	Database Administrator
DEV	Development
DT	Developmental Test
EN	Engineering Management
FM	Financial Management
IA	Information Assurance
MPS	Mission Planning System
OB	Organizational Business Management
OPS	Operations
OT	Operational Test
OTP	Operational Transfer Program
SEC	Security
SPO	System Program Office
SwA	Software Assurance
TOs	Technical Orders

Table 1. *Figures 2, 5, and 6 Acronym List.*

Enabling Entities

Figure 2 above shows an example of what enabling entities exist to support the A-10 Operational Flight Program (OFP) software development team in the 309 SWEG's 520 SWES. A-10 OFP is a large software development program of 95 personnel dedicated to ensuring the A-10 pilot community has the software updates and capabilities at the time they need it with the highest quality possible. This figure is not all-inclusive as there are other teams operating behind the scenes, such as Personnel, Base Security, Labor Relations, etc.

As shown in **Figure 2,** there are many enabling entities that directly support, develop in conjunction with, or otherwise impact the ability for A-10 OFP development to complete their mission. A-10 OFP development began the transition to an agile environment in spring 2020, utilizing the tenets of Scrum of Scrums and Scaled Agile Framework®, but designing much of the operations with DevSecOps principles. Over the course of the last three years, A-10 OFP has seen many process improvements, process throwaways, and - at times - complete process replacements to implement an efficient agile software work environment. The A-10 OFP agile development lifecycle has begun to stabilize into a process that proves that DevSecOps and agile development can and do work well for large programs and programs supporting decades-old infrastructure.

DevSecOps Implementation Challenges

Some of the challenges that have been identified have been the following. This list is not comprehensive but represents some of the bigger issues that teams can experience as they transition to a DevSecOps development organization.

- Team member buy-in
- Leadership buy-in
- Role definitions
  - Under Scrum, there are Scrum Masters, Product Owners, Coaches, etc. that need to be defined and those filling the roles need to be trained
  - After designating a Scrum Coach, role definitions and expectations should be established, and training must provided to team members
- System Program Office (SPO) buy-in
- Developmental and Operational Flight Test organizations being able to keep up with established release cadence
- Development network stability, which is critical to consistency in DevSecOps, has not been reliable
- Reliability and stability of network-based tools
- Test stand simulator/emulator development is critical and must be done quickly, reliably, and in sync with the OFP development
- The maintenance team implementing updates to the test equipment and lab is not always in the loop on what requirements need to be updated and/or added
- Software tool licenses have been difficult, at times, to procure from license control
- Cybersecurity and software assurance processes and tools must be defined and implemented
- The role and processes for configuration management (CM) has proved challenging
- Legacy development toolsets to pipeline enabled tools must be converted
- Test procedures for decades-old legacy code must be automated

The Silo Effect

One of the biggest hurdles to converting programs to DevSecOps business practices is the silo-managed business culture. A silo-based organization is run on the premise that each team is managed as an independent team with little to no input from other teams. Things such as “standard practices” are often defined differently by each team, even if the basis of the standard practice is the same. Ideas and resources are hoarded and kept internal to each team and collaboration is negligible or non-existent. The silo effect can be, and often is, detrimental to the overall success of an organization [2]. The issues of a silo-based organization are easily seen when DevSecOps principles are practiced by some teams and not by others.

When an engineering team encounters an issue with their development environment, their initial response is: “How do we fix/overcome the issue?” If the solution for the issue lies with an enabling entity that operates in a silo, it can be frustrating for the engineering team when the issue resolutions are not addressed in a timely manner. Instead, they find a way to “Band-Aid®” or “stopgap” the issue so they can continue working.

Whether intentionally done or just a result of decades of silo business practices, teams set up rules of engagement for support/communication that often become roadblocks. Individual team members on all teams are typically ready and willing to help each other, but for them to connect with one another, they must do it via backdoors or tunneling under the roadblocks. In some cases, team members can get around roadblocks, but in others, they may not be able to. **Figure 3** shows what this behavior looks like.

For DevSecOps to work, the expectation must be set that all team members need to be counted on to complete their responsibilities without much, or any, oversight. Enabling entities that support software development teams need to be counted on to provide the necessary support. Development teams need to set clear requirements for the enabling entities, so they have well-defined expectations of what successful support looks like. **Figure 4** shows the ideal relationship between teams with open communication channels that are fully supported.

Scores of books and training classes have been written about teamwork and the importance of working together. In *The DevOps Handbook* [3], there are many examples of businesses from many different industries who utilized the principles of DevOps to turn their failing business around. In many of the examples, different teams from different departments controlled the overall outcomes and success of their software development. For most, success began to take shape when the controlling entities were either combined and consolidated into single teams, or, in some cases, removed from the process altogether.

The PRC2 program experienced this firsthand as they began their conversion to DevSecOps. Their customer levied technical requirements on the program that the 309 SWEG enabling entities could not support at that time. For PRC2 to establish the development environment that they

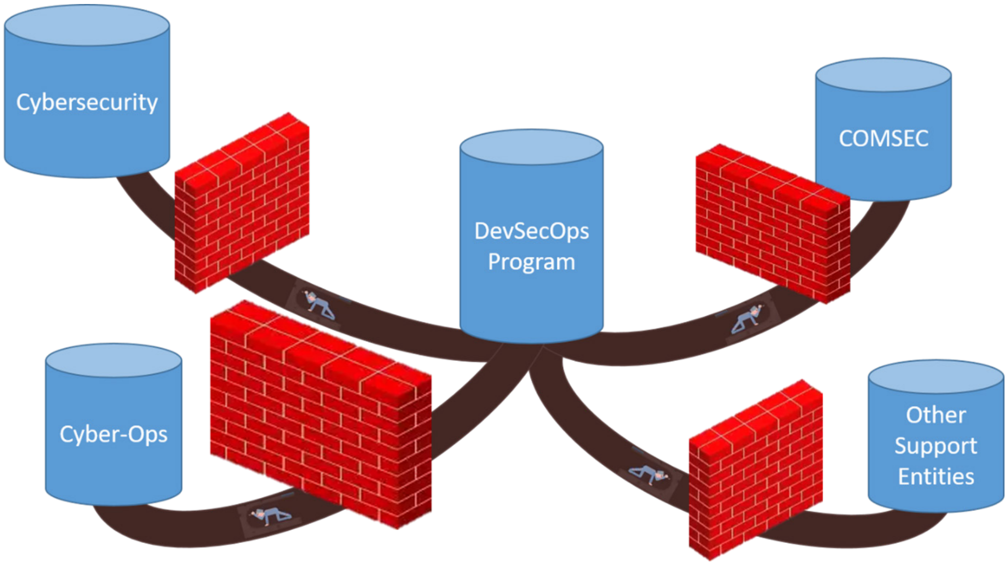


Figure 3. *DevSecOps Silo Team Interactions.*

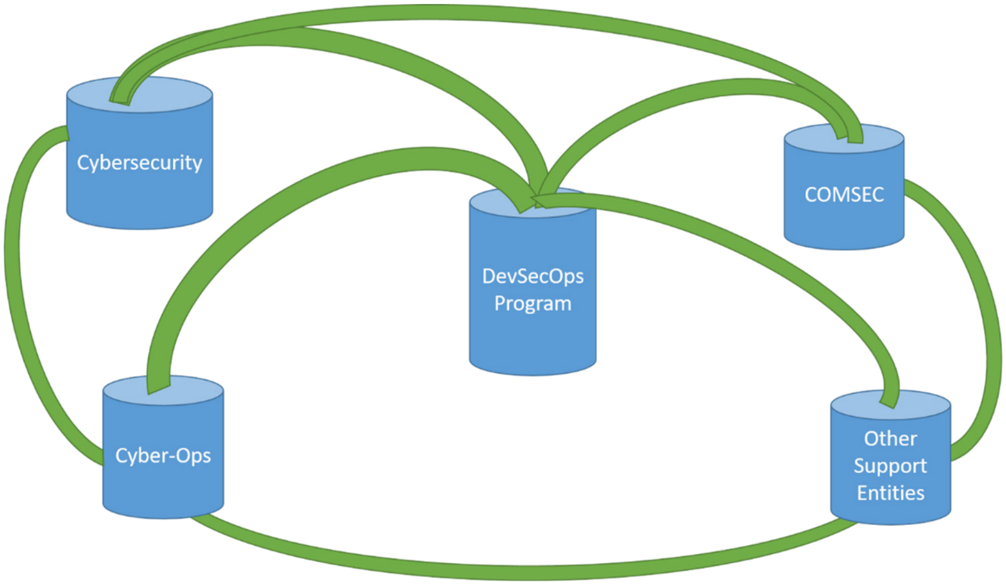


Figure 4. *DevSecOps Communication and Support Channels.*

needed with the tools required, they set up their own environment. They set up servers, development networks, repositories, pipelines, cyber security, worked with Authorizing Officials to secure ATOs, etc., hiring their own cyber security experts and IT specialists. After establishing their own cyber team, they were able to move forward and successfully transition to a true DevSecOps business model. Although it worked for PRC2, having every other program in 309 SWEG do the same is not sustainable or recommended. From a security standpoint, having a centralized network and tool control center helps keep expenses down (economies of scale) and standardizes security rules.

The overall mission of any organization is to succeed. When teams operate in silos, the definition of success for the organization is skewed to be the definitions of success interpreted by the different teams operating within the organization. The consolidation of successes achieved by each independent team do not always match the definition of success defined by the organizational leaders. When the silos and barriers between teams are broken down enough that the vision and mission of the organization becomes visible to everyone equally, success will inevitably follow. The definition of success needs to be shared and understood by all teams, rather than be defined by each team. Each team is a cog in the overall process machine. When one cog stops turning, the whole system comes to halt.

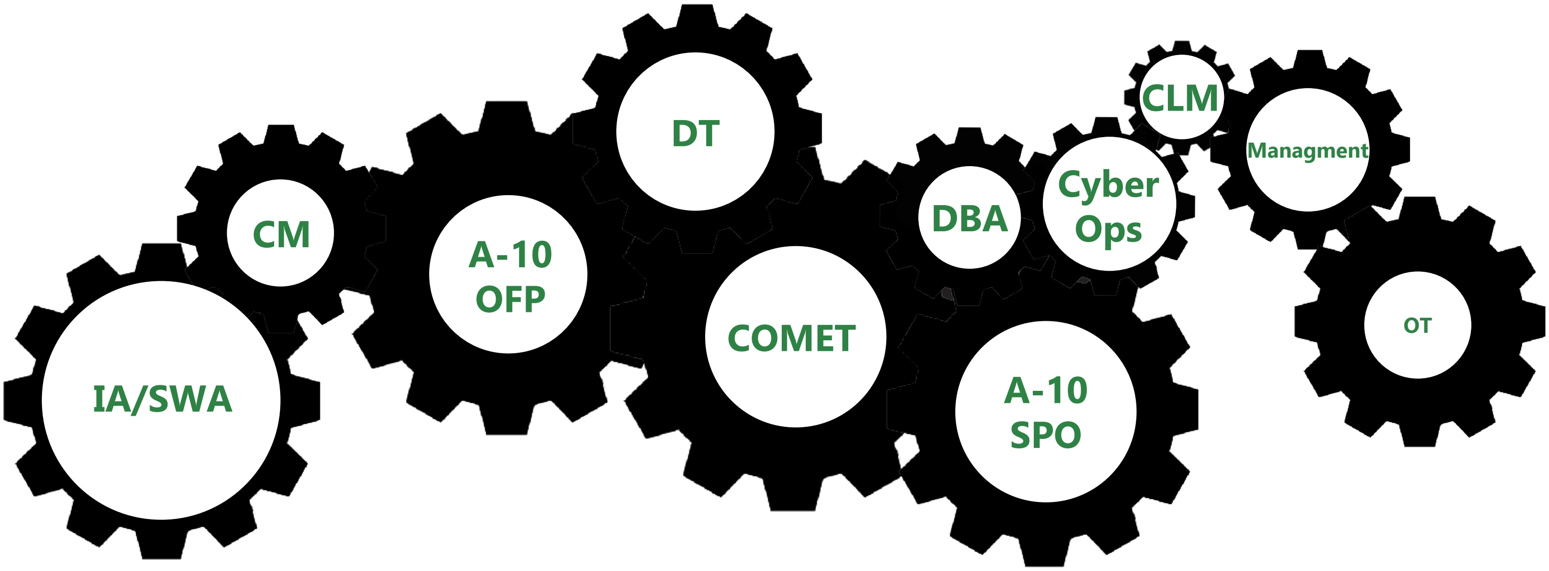


Figure 5. *Enabling Entities and Development Teams Work Together. For more information, See Table 1 on page 48.*

Implementation Focus

What has been discovered as the A-10 OFP team has gone through the process of transitioning to agile business practices is that when all the enabling entities are in the room discussing plans and procedures, everyone gets excited and ensures one another they are on board. What everyone is actually thinking is more along the lines of, “How in the world are we supposed to do that?” When they all leave the meetings, many often continue business as usual while the software team goes into DevSecOps mode, believing their enabling entities are ready to provide support.

In many of the discussions between the software teams and enabling entities, the focus of the discussion is how DevSecOps works for software development. All parties are generally expected to accept and adopt DevSecOps principles and practices for each of their teams. For an enabling entity, this doesn’t always make sense. For example: the cyber operations team has the duty to maintain the development network stability, enable required software development tools, and keep the software factory running smoothly. Their day-to-day business operations may not lend themselves to being run in an agile or DevSecOps business model. And that is okay. Yet, in these DevSecOps meetings, these entities are made to feel that if they do not adopt DevSecOps practices, they are somehow failing. This lack of proper understanding further enables the silo effect to perpetuate.

Next Steps

The path from silo-based to integrated teams is straight forward. Instead of software teams demanding that enabling entities adopt agile or DevSecOps practices, which often they cannot conform to, discussions should focus on how each enabling entity is expected to provide support. This can be a daunting endeavor in large organizations; the 309 SWEG, for example, has a half dozen or so squadrons doing development work, executing scores of work products, and utilizing hundreds of software tools. This means support entities not only have a wide range of responsibilities, but also a lot of people who need their help. Instead of teaching how an enabling entity should adopt and implement DevSecOps, software teams should have discussions that teach enabling entities how the software team is implementing DevSecOps and how the enabling entity affects the overall success of the software team.

By defining what support requirements the software team has from each entity and how failure to receive that support affects their success, enabling entities stop acting as if their failure to "go agile" is a failure on their part and start finding ways to help achieve success.

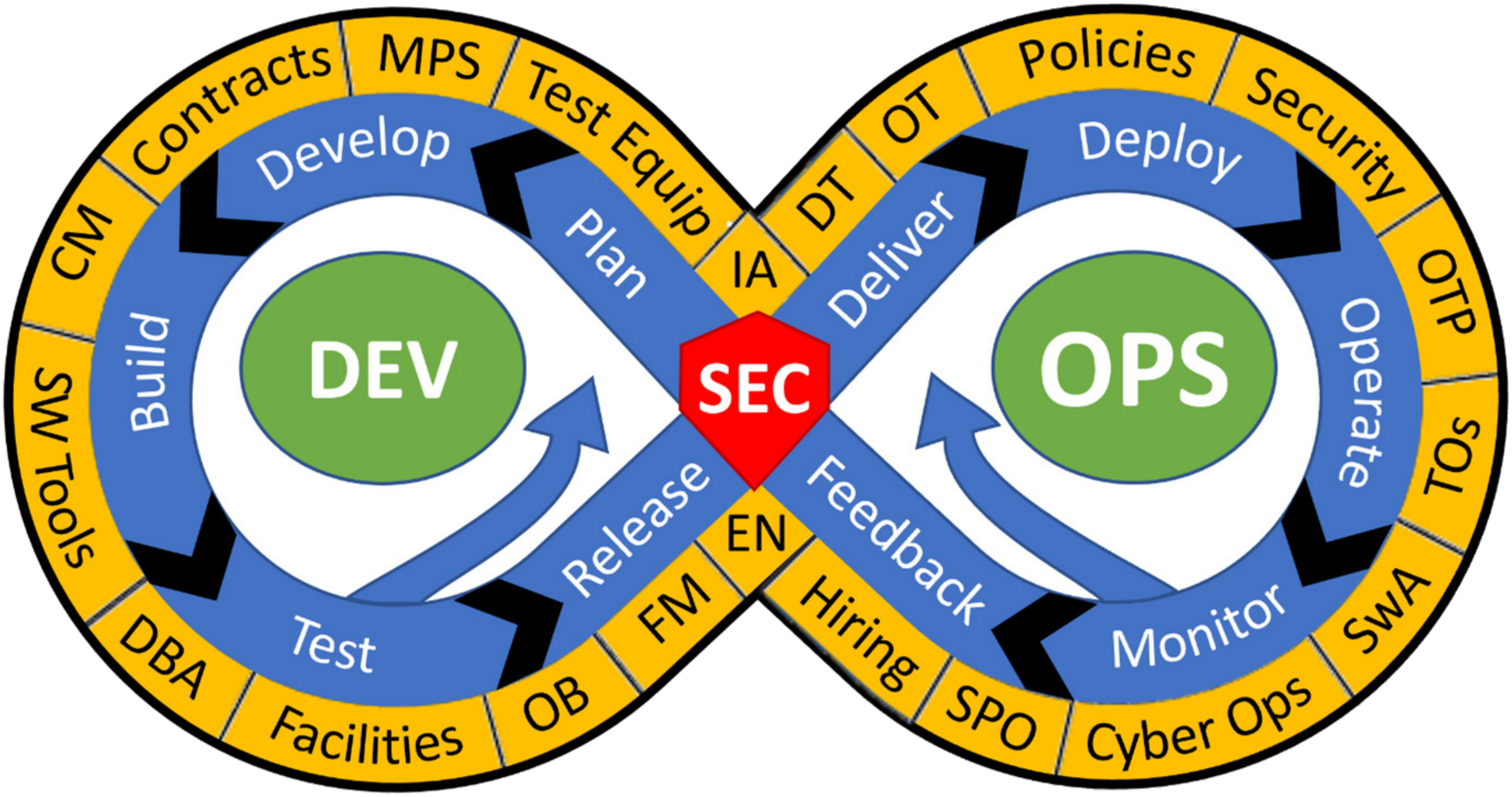


Figure 6. *All Groups Coming Together. For more-information, See Table 1 on page 48.*

Enabling entities need to be aware of the support needs and requirements of the programs they are supporting. Enabling entities have specific areas of responsibility and expectations for what support they should be providing, and development programs need to understand what each enabling entity is expected to support and be careful not to levy unrelated or unrealistic expectations on an enabling entity. When things get rough, enabling entities should not only understand what services they are expected to provide, but how the lack of services negatively impacts overall mission success. For expectations to be understood, all parties need to define what they need, what they will support, how they are expected to communicate with each other, and expectations of timelines for support requests.

Recommendations

As development programs evolve, they need to engage with enabling entities to communicate any changes in requirements and request changes to the support they need from those entities. Open communication is not just a good idea but is crucial to the success of all parties involved. Enabling entities must define their role in the agile process and what they can/will support.

At higher levels of Air Force and even the DoD, senior leadership that deals with software development has been pushing for software development to evolve into the agile mindset. There have been many changes to policies and DoD instructions that support the transition to agile for software programs, but additional support in waiving/rewriting outdated policies at lower, middle-management levels in favor of pushing the agile transition faster will be required for enabling entities to make the changes necessary to fully support agile software acquisition. It will take senior leaders disrupting “business-as-usual” mindsets to make the agile transition succeed.

Conclusion

The process of driving change, especially to an organization as large as the 309 SWEG, is difficult at best. Enabling entities play an integral role in the success of all software development programs. Development programs need to acknowledge this fact and enabling entities need to feel like part of the team. DevSecOps implementation will succeed once all teams are on board with how they fit into the big picture. The narrative needs to change from, “Adopt agile or else” to “Understand how agile DevSecOps business practices work so you can better understand how to support them.”

Not every enabling entity needs to adopt agile principles, but they must be able to support the teams that have. There are many ways to do this and each enabling entity should be empowered to define the solution that works for their team to meet the requirements from the organizations they support. They must commit to sustainably providing that level of support. The mission of the 309 SWEG – and of all DoD software organizations - is to “Produce – Support – Sustain for the Warfighter.” To do this, all teams need to unite behind this mission and leave their silos and competing interests behind.

References

[1] Department of Defense, “DoD Enterprise DevSecOps Reference Design,” August 2019

[2] Indeed Editorial Team, “What Are Silos in Business? Causes for Organizational Silos,” June 10, 2021, extracted from https://www.indeed.com/career-advice/career-development/silos-in-business, extracted on February 9, 2022

[3] Debios, Patrick; Humble, Jez; Kim, Gene; and Willis, John, “The DevOps Handbook: How to create world-class agility, reliability & security in technology organizations,” 2016, IT Revolution Press LLC

About the Author

Michael is a Project Engineer with L3Harris, applying the lessons learned with converting a legacy program to use agile business methods with his new program. He spent 11 years with the USAF in the 309th Software Engineering Group where he learned how to run successful programs with great teams who were ready to make changes. He earned a Bachelor degree in Electrical and Computer Engineering and a Master’s in Business Administration.

Michael Engh

Project Engineer

L3Harris Technologies, Inc

Michael.engh@l3harris.com

435-757-4134

BACKTALK

Failure to Communicate

Alan C. Jost
Senior Software Program Manager,
Raytheon Technologies (RTX)

The tagline from *Cool Hand Luke* (1967) has often been modified from its original. The Captain (actor Strother Martin) said to recalcitrant chain gang prisoner, Luke (actor Paul Newman):

“What we’ve got here is (pause) failure to communicate,”

NOT

“What we have here is a failure to communicate.”

We don’t even quote the quote correctly. This BackTalk is a look at situations where communication among team members was the critical factor in the potential success or failure of the program. When humans communicate in written form, verbally, and, yes, even with non-verbal communications, many times the receiver receives the message not quite the way the transmitter meant. The generation of Policy and Guidance (P&G) documentation generates its own communication issues where the author or responsible organization generating the P&G communicates from their perspective, which can be totally different from the readers or the folks who must implement the P&G. Even more dramatic is the gap between software and hardware, be it at the human engineering level or at the lower physical level of software code executing on the hardware. The “failure to communicate” is the root cause for many program failures, policy non-compliances, or implementation errors using the guidance instructions; this occurs many more times than we would like to admit or appreciate. Throughout my career, I experienced some “program failures,” and even the term “failure” is relative and subject to a wide range of meanings depending on the folks participating in the discussion. These program failures can be related to the focus of this BackTalk: “failure to communicate.” Everyone reading this article more than likely has similar experiences and could add to the few situations described below. But I do not want to just dwell on the failures, so good examples on how participants were able to “communicate” is presented.

NASA Mars probe

One of the most dramatic failures of a project caused by a failure to communicate was a NASA probe project. The probe, called the Mars Climate Orbiter, was to orbit Mars to gather climatic data. The Orbiter, at a cost of about $125M, traveled over 400,000,000 miles to get to the planet. Upon arrival, the Orbiter entered an orbit 60 miles too low and, since it was not built to withstand the Mars atmosphere, the Orbiter was destroyed. The software design calculations used to place the spacecraft into orbit were made in imperial measures in terms of “pounds force.” The software team, however, developed the “burn” control software using metric measurements and units in terms of newtons. While the error was less than 0.000015 percent, it was enough to be fatal to the hardware mission. The error was uncovered during the post-mortem of the failed mission. This major failure to communicate between teams of highly professional, super intelligent, and experienced professionals that did not check even the most obvious items in the design and limitations in the implementation . Perhaps a simple P&G to use standard imperial measures would have eliminated this basic error resulting in software communicating to the hardware probe to establish an orbit too low for survivability of the hardware mission. What we’ve got here is…failure to communicate!

Radar Red Time

In one situation I was personally involved with, three organizations [the contractor, customer, and operational user] were collaborating to build a large radar system. The new radar was located near the old radar it was replacing. The old radar would not be decommissioned until the new radar was successfully operationally tested. In order to do this, maintenance Red Time of the old radar had to be scheduled when the new radar would be tested, and this is where the three organizations ‘failed to communicate’. A simple P&G describing the specific definition of “Red Time” would have been an exceptional piece of written policy and guidance which did not exist. Through many planning meetings for Red Time, each group had a different interpretation of what exactly Red Time was. The meetings were productive and provided for a very detailed operational test schedule. However, each organization had a different interpretation of the Red Time and how that would be implemented with the old and new radars operating in close proximity of each other. The failure to communicate between the organizations was discovered at the first operational test event when the contractor requested that the old radar be turned off for the scheduled Red Time!

The contractor software engineers assumed that Red Time meant the old radar was turned off so they could test the new radar without interference from radiation being transmitted from the old radar. The customer hardware maintenance engineers assumed that Red Time meant that the old radar, while not turned off, would be placed in a maintenance state where the transmission radiation would be re-routed through the wave-guides, eliminating a large portion of the ambient radiation. The Space Command operational user’s version of Red Time meant that only the transmission lines for the radar data would be “disconnected,” essentially grounded, so a false target could not be transmitted. Well, the reaction from the operational user was “turn the old radar off!!?? The radar maintenance engineers had never turned the old radar off, they did not even know how to turn it off and, even worse, they didn’t know how to turn it back on!!” What we’ve got here is…failure to communicate!

At the heart of the situation was the klystron, the large tube that generated the radiation used to transmit the radar signal. Once turned on, it had not been turned off for years and there was no guidance on how to turn it off nor back on. In near real time, the three groups had to communicate with the klystron manufacturer to generate guidance and procedure to minimize the energy and redirect the lower energy down the wave-guides. The new guidance did work, and the power down/up sequence was successfully repeated numerous times to support the operational testing of the new radar. The “failure to communicate” the concept of Red Time among the participating organizations would have led directly to a major schedule impact on the program. It forced a real time communication between the participating software engineering contractor, operational radar maintenance engineers, the operational user organization organizations, and hardware manufacturer resulting in the power down/up procedure which, if it failed, would have resulted in a major impact to the program. Worse yet, the impact to the strategic mission of the radar. The procedure worked and the major schedule impact was avoided. What we’ve got here is… communication! We closed the software and hardware gap.

Apollo 13 Recovery

The original Apollo 13 problem was caused when the number two oxygen tank in the Service Module exploded by short circuit in the hardware oxygen tank that occurred during a routine software “stirring” procedure. This problem was not the result of a “failure to communicate.” What I’m using this dramatic mission failure for is to demonstrate the success achieved with the ability of the NASA Apollo ground team to communicate effectively, not only between themselves to develop solutions, but to communicate those solutions to the Apollo 13 crew. The initial explosion also caused the number one oxygen tank to fail and the fuel cells that supplied the Command Module with electricity also had problems.

In the initial 90 minutes, the mission control ground crew brainstormed a way to use the Lunar Lander as a lifeboat for the crew. However, the Lunar Lander was designed to be used for 45 hours only and the return mission around the moon would take 90 hours. There was plenty of oxygen with barely enough electrical power to make the rescue journey. The problem was the eventual buildup of CO2 (carbon dioxide) in the command module and lunar lander spacecrafts. There were enough lithium hydroxide canisters in the Command and Lunar Lander modules between them, but the Command module square canisters were not compatible with the round openings in the Lunar Lander module control system.

The Houston Mission Control gave the brainstorming team only the materials available to the Apollo 13 crew. The brainstorming team had to come up with the solution to the Apollo 13 “square peg in the round hole” problem. Once they came up with the solution, they had to communicate that solution to the crew to implement. Using plastic bags, tape, cardboard, and the square canisters themselves, the brainstorming team came up with the solution. In addition, software calculations and commands were needed to conserve the oxygen and battery power in conjunction with calculating the return trajectory and re-entry models to allow the damaged spacecraft to return the three astronauts safely home. They were able to communicate that solution to the crew in time for their implementation, and the rest is history… “What we’ve got here is… communication.”

Conclusion

Human to human communication is critical in managing programs. This is even recognized in the Capability Maturity Model Integration (CMMI) where stakeholder involvement, reviews with higher levels of management, and other process areas, specific and generic Policy and Guidance are based on “NOT failing to communicate.” One of our popular phrases is “I hear you,” which, generally translated, means that one person understood what the other person meant to say. While the words used truly mean that you physically heard the words spoken, a more appropriate response is “I understood you.” I leave you with a famous movie quote which is an excellent example of precise communication. Again, a bit contrived, but it makes the point. In the movie, “The Fugitive,” during the scene right after the train wreck where Dr Richard Kimball (Harrison Ford) escapes, US Marshall Sam Girard (Tommy Lee Jones) has to take over a just formed, very large search team of local police extremely reluctant to be led by a “Wyatt Earp” US Marshall. He communicates precisely what he needs done. In one short, memorable speech he communicates his requirements and what he needs done !

US Marshall Sam Girard: “Listen up ladies and gentleman. Our fugitive has been on the run for 90 minutes.

Average foot speed over uneven ground, barring injury, is four miles an hour. That gives us a radius of six miles.” - requirements – “What I want out of each and every one of you is a hard target search of every: Gas station, Residence, Warehouse, Farmhouse, Henhouse, Outhouse, and Dog House in that area. Check points go up in 15 minutes. (pause). Your fugitive’s name is Dr. Richard Kimball. (pause). Go get him! - [what he wants done].

Any questions on how clear his communication was? In real estate, the most important thing is: location, location, location; to close the Software/Hardware Gap is: communication, communication, communication.

About the Author

Alan C. Jost, retired USAF Lt Col, received a Bacheors degree in Mathematics, the University of Miami; AFROTC commissioned 1971; Masters in Psychology, University of Northern Colorado; MS in Computer Science, University of Arizona; Mechanical Engineering in Computer Engineering, Vanderbilt University. Retiring from Air Force in 1991, he is now a Senior Software Program Manager at Raytheon Technologies (RTX), managing international and domestic Air Traffic Control automation systems. He is currently managing RTX programs located in Norway, Australia, and the domestic EnRoute Automation System (ERAM) in the U.S.

Lt Col Allen C. Jost

Senior Software Program Manager

Raytheon Technologies (RTX)

Alan.C.Jost@rtx.com

CrossTalk Sponsor

NOW HIRING

309th Software Engineering Group, Hill AFB, Utah

OPEN POSITIONS:

- Software Engineer ✓
- Computer Scientist ✓
- Mechanical Engineer ✓
- IT Specialist ✓
- Cybersecurity Specialist ✓

For More Information:

https://afscsoftware.dso.mil/careers

Send Your Resume:

✉ 309SMXG.Recruiting@us.af.mil

SWEG Socials