VOLUME 6 NUMBER 3
OCTOBER 2018

CSIAC JOURNAL

A Quarterly Publication of the Cyber Security & Information Systems Information Analysis Center

INNOVATION BASED ECOSYSTEMS

INTEGRATED CYBER CAPABILITIES
CENTRALIZED DATA COLLECTION
DISTRIBUTION POINTS
DATA-BASED DECISIONS

CSIAC

Cyber Security & Information Systems Information Analysis Center

ABOUT THE CSIAC

As one of three DoD Information Analysis Centers (IACs), sponsored by the Defense Technical Information Center (DTIC), CSIAC is the Center of Excellence in Cyber Security and Information Systems. CSIAC fulfills the Scientific and Technical Information (STI) needs of the Research and Development (R&D) and acquisition communities. This is accomplished by providing access to the vast knowledge repositories of existing STI as well as conducting novel core analysis tasks (CATs) to address current, customer focused technological shortfalls.

OUR MISSION

CSIAC is chartered to leverage the best practices and expertise from government, industry, and academia in order to promote technology domain awareness and solve the most critically challenging scientific and technical (S&T) problems in the following areas:

- Cybersecurity and Information Assurance
- Software Engineering
- Modeling and Simulation
- Knowledge Management/Information Sharing

The primary activities focus on the collection, analysis, synthesis, processing, production and dissemination of Scientific and Technical Information (STI).

OUR VISION

The goal of CSIAC is to facilitate the advancement of technological innovations and developments. This is achieved by conducting gap analyses and proactively performing research efforts to fill the voids in the knowledge bases that are vital to our nation. CSIAC provides access to a wealth of STI along with expert guidance in order to improve our strategic capabilities.

WHAT WE OFFER

We provide expert technical advice and assistance to our user community. CSIAC is a competitively procured, single award contract. The CSIAC contract vehicle has Indefinite Delivery/Indefinite Quantity (ID/IQ) provisions that allow us to rapidly respond to our users' most important needs and requirements.

Custom solutions are delivered by executing user defined and funded CAT projects.

CORE SERVICES

- Technical Inquiries: up to 4 hours free
- Extended Inquiries: 5 - 24 hours
- Search and Summary Inquiries
- STI Searches of DTIC and other repositories
- Workshops and Training Classes
- Subject Matter Expert (SME) Registry and Referrals
- Risk Management Framework (RMF) Assessment & Authorization (A&A) Assistance and Training
- Community of Interest (COI) and Practice Support
- Document Hosting and Blog Spaces
- Agile & Responsive Solutions to emerging trends/threats

PRODUCTS

- State-of-the-Art Reports (SOARs)
- Technical Journals (Quarterly)
- Cybersecurity Digest (Semimonthly)
- RMF A&A Information
- Critical Reviews and Technology Assessments (CR/TAs)
- Analytical Tools and Techniques
- Webinars & Podcasts
- Handbooks and Data Books
- DoD Cybersecurity Policy Chart

CORE ANALYSIS TASKS (CATS)

- Customer tailored R&D efforts performed to solve specific user defined problems
- Funded Studies - $1M ceiling
- Duration - 12 month maximum
- Lead time - on contract within as few as 6-8 weeks

CONTACT INFORMATION

266 Genesee Street
Utica, NY 13502

1 (800) 214-7921

info@csiac.org

/DoD_CSIAC
/CSIAC
/CSIAC

CSIAC is operated by Quanterion Solutions Inc and sponsored by the Defense Technical Information Center (DTIC)
266 Genesee Street Utica, NY 13502 | 1 (800) 214-7921 | info@csiac.org | https://www.csiac.org

ABOUT THE JOURNAL OF CYBER SECURITY AND INFORMATION SYSTEMS

ABOUT THIS PUBLICATION

The Journal of Cyber Security and Information Systems is published quarterly by the Cyber Security and Information Systems Information Analysis Center (CSIAC). The CSIAC is a Department of Defense (DoD) Information Analysis Center (IAC) sponsored by the Defense Technical Information Center (DTIC) and operated by Quanterion Solutions Incorporated in Utica, NY.

Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the CSIAC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the CSIAC, and shall not be used for advertising or product endorsement purposes.

ARTICLE REPRODUCTION

Images and information presented in these articles may be reproduced as long as the following message is noted:

"This article was originally published in the CSIAC Journal of Cyber Security and Information Systems Vol.6, No 3"

In addition to this print message, we ask that you notify CSIAC regarding any document that references any article appearing in the CSIAC Journal.

Requests for copies of the referenced journal may be submitted to the following address:

Cyber Security and Information Systems
266 Genesee Street
Utica, NY 13502

Phone: 800-214-7921
Fax: 315-732-3261
E-mail: info@csiac.org

An archive of past newsletters is available at **https://www.csiac.org/journal/.**

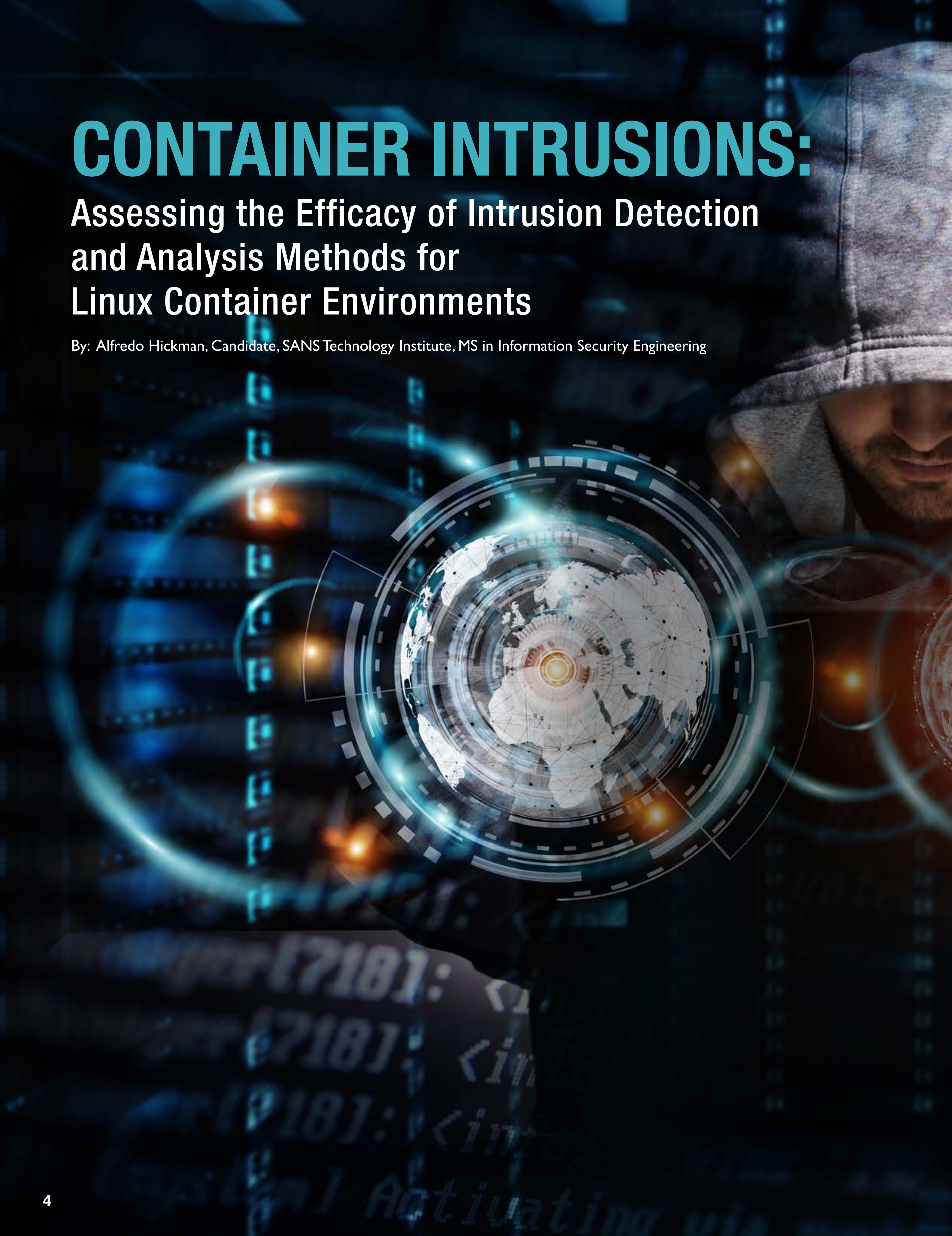
*To unsubscribe from CSIAC Journal Mailings please email us at **info@csiac.org** and request that your address be removed from our distribution mailing database.*

Journal of Cyber Security and Information Systems
INNOVATION BASED ECOSYSTEMS

CONTAINER INTRUSIONS:

Assessing the Efficacy of Intrusion Detection and Analysis Methods for Linux Container Environments

By: Alfredo Hickman, Candidate, SANS Technology Institute, MS in Information Security Engineering



The unique and intrinsic methods by which Linux application containers are created, deployed, networked, and operated do not lend themselves well to the conventional application of methods for conducting intrusion detection and analysis in traditional physical and virtual machine networks. While similarities exist in some of the methods used to perform intrusion detection and analysis in conventional networks as compared to container networks, the effectiveness between the two has not been thoroughly measured and assessed: this presents a gap in application container security knowledge. By researching the efficacy of these methods as implemented in container networks compared to traditional networks, this research will provide empirical evidence to identify the gap, and provide data useful for identifying and developing new and more effective methods to secure application container networks.

INTRODUCTION

This research systematically assesses the efficacy of intrusion detection and analysis methods as applied to Docker Linux application container environments compared to the effectiveness of similar methods applied in traditional networks. Linux application container technologies can provide many benefits, but can also introduce complexity and vulnerabilities. Furthermore, the means and methods for securing container environments are young and not evolving as rapidly as the container technologies themselves. With the rapid evolution and adoption of Linux application container technologies in the enterprise, not much scholarly research exists on how to balance the benefits that containers provide with the vulnerabilities that they introduce.

The unique and intrinsic methods by which Linux application containers are created, deployed, networked, and operated do not lend themselves well to the conventional application of methods for conducting intrusion detection and analysis in traditional physical and virtual machine networks. While similarities exist in some of the ways used to perform intrusion detection and analysis in conventional networks as compared to container networks, the effectiveness between the two has not been systematically measured and analyzed: this presents a gap in application container security knowledge. By researching the efficacy of intrusion detection and analysis methods as implemented in container networks compared to traditional networks, this research will provide empirical evidence to identify the gap, and provide data useful for conceiving and developing new and effective methods to secure container networks As such this research will attempt to answer the following question: How effective are methods for conducting intrusion detection and analysis in Docker Linux application container networks when compared with the efficacy of similar methods in traditional networks?

"empirical evidence to identify the gap, and provide data useful for conceiving and developing new and effective methods to secure container networks"

LINUX CONTAINERS AND DOCKER – A BRIEF HISTORY

Linux application containers as they are known today, are most directly the product of Cgroups, which was formally introduced in the Linux Operating System Kernel version 2.6.24, in 2007 (Bottomley & Emelyanov, 2014). At a high-level, Linux application containers are lightweight virtual machines that share the same underlying operating system kernel, consume the same or shared resources, and contain the code, tools, dependencies, and settings required to function. Due to the benefit of containerized applications sharing the same underlying host kernel, container hosts can reach a much higher deployment density than traditional dedicated application or virtual machine hosts. In addition to the application deployment density benefit, Linux container deployments also benefit from a shared kernel, with application dependencies residing within the individual containers. This benefit allows developers, I.T. operators, and system owners to reduce the equipment, software, and operational overhead required to service application workloads and their associated costs: Both reasons are why Linux application container technologies have soared in popularity over the last few years (Mohallel, Bass, & Dehghantaha, 2016).

In 1979, Bell Laboratories released Unix v7, which introduced chroot into the Unix ecosystem. Change root, or chroot for short, gives the operating system thecapability to change the logical root directory of a running process and its child processes to isolate the processes from awareness and access to neighboring processes and resources. The chroot feature enables efficient and more secure practices for application context isolation and testing, and it set the conceptual stage for Linux application containers almost 30 years later. After the development of chroot in 1979, it was not until the early 2000's that new applications of process isolation and control, more resembling the current Linux application containers we know today, began to emerge. Systems such as FreeBSD Jails, Linux VServer, Solaris Zones, and others set the stage for contemporary Linux application container and management technologies such as LXC, RKT, Kubernetes, and most famously, Docker (Petazonni, 2015).

While relatively new to the mainstream, Linux application containers have been around since 2006, when Google developers, Paul Menage and Rohit Seth developed Control Groups (cgroups). Control Groups are a Linux kernel feature that enables group process management and accounting. Another critical and foundational technology that allowed the creation of modern Linux application containers is Linux namespaces. Linux namespaces, introduced in the Linux Kernel version 2.4.19, while similar to cgroups that came after it, is different and complimentary to cgroups. Namespaces serve to isolate groups of processes into logical units that are restricted to the unit and limited in their interaction with and consumption of host system resources (Bottomley & Emelyanov, 2014). In essence, the foundational technologies that enable Linux application containers are cgroups for resource consumption management and accounting, and namespaces for logical partitioning and regulation of host resource access and consumption.

A lot has changed in the Linux application container world since the development of chroot and the adoption of namespaces and cgroups into the Linux kernel. Moreover, while still relatively new in the general enterprise, companies such as Google, AWS, and Facebook have been using containers for the better part of a decade (Winkel, 2016). So, since Linux application containers have been around for years, why are we only now seeing the general adoption of the technology into the enterprise? The likely answer to that is - Docker. Solomon Hykes launched the Docker project while working as an engineer at dotCloud in France. Hykes realized that while Linux application containers were readily available and decently mature for production implementation, the technologies were overly complicated and not yet palatable for general enterprise adoption (Hykes, 2013).

At Pycon 2013, with this realization in mind, Hykes released Docker for open source distribution. At a high-level, Docker is a Linux application container management system which abstracts away much of the complexities associated with containerized application development and host infrastructure operations (Mohallel, Bass, & Dehghantaha, 2016). However, since the public release of Docker and with the associated and significant increase in its development and adoption in the enterprise, many vulnerabilities in the underlying and related technologies have been discovered. Furthermore, the complexity associated with developing, delivering, deploying, and operating containerized applications and their host infrastructures have introduced new challenges and paradigms in the way that security professionals secure such environments.

Linux Container Security – a New World

The unique methods by which application containers are created, deployed, networked, and operated present unique challenges when designing, implementing, and operating security systems for these environments. Due to the frequent practice of binding containers to non-standard network ports, deploying application workloads dynamically over distributed hosts, integrating rapidly evolving application code on containers in production, and having specific container instances provisioned for brief periods of times, container technologies have become prime targets for adversary attack and exploitation. Just as the security industry evolved to secure the enterprise during the introduction of computer virtualization, the security industry will need to evolve again, and more rapidly, to secure application container infrastructures if the industry hopes to keep up with the rapid rate of change.

> *"OSSEC is one traditional HIDS that can leverage the Linux Audit logic to parse system calls and enable BoSC implementations"*

INTRUSION DETECTION SYSTEMS AND ANALYSIS IN A DOCKERIZED WORLD

The existing body of scholarly literature related to developing methods and systems for conducting intrusion detection and analysis in application container networks is limited. However, there does exist a body of foundational scholarly research and literature in application container security, intrusion detection methods, and analysis on which to build. Furthermore, there are many sources available related to traditional methods and systems for conducting intrusion detection and analysis to compare to new and evolving techniques used in container networks.

For instance, Abed, Clancy, and Levy (2015), found that due to the way container technologies enable a single host operating system kernel to provide resources to containers, attacks on the container based-applications could result in compromises of the container hosts, other containers, and even other networks. With this realization, Abed et al. proposed the adaptation of the Bag of System Calls (BoSCs) method, sometimes used in traditional host-based intrusion detection, to create a container-based host intrusion detection system. The technique that Abed and team developed requires the monitoring of system call frequency between individual container processes and the host operating system kernels for anomaly detection (2015). By recording the frequency of system calls between the container host kernel and container processes, the BoSC system could learn what normal system call behavior is and then identify statistical deviations from normal to identify anomalous and potentially malicious behavior.

Such adaptations of existing methods for conducting intrusion detection and analysis in traditional networks to container networks is an emerging and promising trend in container security. OSSEC is one traditional HIDS that can leverage the Linux Audit logic to parse system calls and enable BoSC implementations. Such adapted methods aim to port proven security methods to mitigate emerging threats and vulnerabilities that, while not unique to container deployments, are only exacerbated by the typically high volume and speed in which containers are deployed and operated.

Vulnerabilities such as the kind that Gao et al., 2017 discovered indicate how incomplete and ineffective methods used for partitioning and allocating host operating system kernel resources to application containers in multi-tenant cloud environments resulted in information leakages. In the Synergistic-Power Attack proof-of-concept, the authors demonstrated how attackers could use aggregated container, and container host leaked data to potentially orchestrate a distributed power spike attack in a multi-tenant container-cloud to cause power supply faults and electrical outages in a data center.

With the research that Gao (et al., 2017) conducted, intrusion detection and analysis methods could be created to detect the behavior associated with container and container hosts information leakage, and the techniques, tactics, and procedures (TTP), that an attacker would use to conduct the Synergistic-Power Attack. For example, a BoSC based system could monitor the system calls associated with information leakage between a Docker host and container to identify such vulnerabilities.

The adaptation, creation, and implementation of container-centric intrusion detection and analysis methods and systems becomes even more pressing due to research findings which indicate that more vulnerabilities are present in container application deployments than traditional physical or virtual system deployments. For instance, Mohallel, Bass, and Dehghantaha, 2016 conducted quantitative research into how attack surface area differs between applications deployed in traditional physical or virtual machine implementations as compared to container-based implementations. The authors discovered that the amount of vulnerabilities introduced into a container host equals the sum of the vulnerabilities found within the host operating system, the container's base image, and the software packages contained within the containers. The research indicates that deploying applications in containers can increase the number of vulnerabilities present on a container host.

"while the technologies involved in evaluating malicious activities and vulnerabilities have evolved significantly over the years"

Not only does research indicate an increase in the number of vulnerabilities introduced by application container implementations, but it also shows an increased scope and criticality of the vulnerabilities. For example, Winkel (2017), found that attackers could exploit vulnerabilities present in versions of the Linux kernel to escape the process, resource, and permissions security mechanisms provided by the operating system to the application container. Like what Abed (et al., 2015) discovered, this type of exploit could result in an attacker escaping the container and then exploiting the underlying host system and possibly other systems accessible through the network. While similar escape exploits exist and are detectable in traditional virtual machine environments, due to the unique nature of container networking, resource allocation, and deployment methods, the same is not the case in container environments. The complexity of container technologies and operations, the vulnerabilities associated with the technologies, and the immaturity of available security systems, warrants research into adapted and new means for securing such environments.

In contrast to host-based methods for intrusion detection and analysis, such as BoSC, Winkel proposes a network security monitoring (NSM) approach to collect telemetry and provide forensic visibility to human analysts conducting intrusion detection and analysis in Docker container networks. Colm Kennedy, 2016 also proposes a network-based approach that can adapt to container networks. Kennedy's method calls for using network decoys which mimic production systems to coax would-be attackers to exploit the systems. However, these honeypot decoys would be instrumented and monitored in a manner that would facilitate intrusion detection and analysis.

Complimenting the decoy method, Patrick Neise (2016), proposes the idea of using network flow and graphs to identify relationships between hosts and events in a network to aid in intrusion detection and analysis. While the networking and deployment methodologies that application container networks employ are significantly different from traditional TCP/IP network implementations, the methods that Neise describes are analogous to sFlow and relational graph (link) analysis based methods that have been employed to gain visibility into Docker container networks.

As such, network-based intrusion detection and analysis methods such as implementing decoys, flow analysis, and relational graph (link) analysis provide analogous examples to host-based methods such as BoSC and kernel system call tapping. Also, both host and network-based techniques lend themselves well to building containerbased intrusion detections systems and comparing the efficacy between their analogous implementations in traditional networks.

This literature review represents some of the latest research in methods for detecting data leakage, anomalous behavior, vulnerabilities, and exploitation methods in container based environments. Furthermore, the non-container related literature reviewed here represents practices that can and have been adapted to create application container security systems.

Intrusion Detection and Analysis in Traditional and Virtual Networks (Normal IDS & A)

Much literature exists about intrusion detection and analysis in traditional physical and virtual networks. At a high-level, the two standard, mature, and capable approaches to the practice are network-based and host-based intrusion detection and analysis. Tracing their conceptual origins to events in 1986, computer network intrusion detection and analysis gained prominence when

Cliff Stoll, a systems manager at the Lawrence Berkeley National Laboratory, a U.S. government research facility, noticed financial discrepancies in an accounting system. This incident resulted in a dramatic investigation which discovered that the accounting discrepancies were not due to flawed computer logic or an accident by a human accountant, but were due to coordinated intrusions by a foreign state-sponsored agent (Bejtlich, 2013). The event is relevant the practice of intrusion detection and analysis in that it served to raise awareness at the highest levels of the U.S. government to the importance of securing sensitive computer networks and developing national strategic capabilities for conducting computer network defense and offense. In many ways, the events at the Berkeley Lab in 1986 spawned the intrusion detection and analysis industry that we know today (Bejtlich, 2013). Moreover, while much has changed in intrusion detection and analysis since the 1980's, at its core, today's traditional approaches to the practice remain much the same.

At a high-level, modern intrusion detection and analysis systems monitor and assess networks and hosts for patterns and conditions that are indicative of potentially malicious activities and vulnerabilities. Moreover, while the technologies involved in evaluating malicious activities and vulnerabilities have evolved significantly over the years, it is still the predefined or near-real-time definition of malicious activities or vulnerabilities which underpin intrusion detection and analysis methods available today. Even with advances in artificial intelligence, machine learning, and threat information sharing, intrusion detection, and analysis systems rely on patterns of expected normal behavior, definitions of malicious behavior, and identification of deviations from "normal" conditions to identify potential malicious activities and vulnerabilities.

For example, many traditional applications of network intrusion detection and analysis systems are dependent on consistent and pre-defined bindings of an application's network port assignments for analysis. Also, these systems are often reliant on the predefined or near-real-time definition of normal or abnormal network or host activities. These systems will then match signatures against associated events or identify deviations from normal conditional thresholds to produce alerts or automated responses (Bejtlich, 2013). It is easy to see that in environments where what is "normal" for one instance of a provisioned application that may only exist for minutes and be configured with nonstandard network port bindings presents severe challenges to the traditional network and host intrusion detection and analysis paradigms. With the advent of Linux containerized application deployments, that is usually the case.

Intrusion Detection Systems and Analysis in Dockerized Networks

As is often the case in Linux application container deployments, application instances and the containers that host them exist for short periods of time and are regularly provisioned with non-standard network port assignments bound to the underlying host. Furthermore, with best practices for deploying containerized applications calling for microservice architectures, one application deployment could require the provisioning of tens of containers to service the overall application (Hayden, 2015). Microservice architectures in container deployments require that individual services be provisioned one per container and grouped in a logical manner that facilitates services to the whole application instance and its dependencies (Winkel, 2016).

The idea behind microservices architectures in Linux application container networks is to limit the interaction between adjacent services, to continuously deploy and improve the individual services, and to scale resources as required more efficiently. However, it is in many ways the adoption of microservice architectures and the complexity and variance that they introduce into the network which exacerbates the already challenging nature of monitoring and securing Linux application container networks. However, the value that application containers provide, coupled with the vulnerabilities and challenges that the technologies introduce have correspondingly stimulated the evolution of the security industry.

"idea behind microservices architectures in Linux application container networks is to limit the interaction between adjacent services"

RESEARCH METHODS - A TALE OF TWO IDSS

For this research, attack, analysis, and capability experiments were conducted in a lab to assess the efficacy of intrusion detection and analysis capabilities in Docker container networks compared to the effectiveness of similar methods in traditional networks. The lab consists of a single network with deployments of both traditional and container-centric intrusion detection systems. The tests were conducted on Ubuntu 16.04 LTS hosts. All hosts were up to date at the time of the experimentation and were instrumented with the OSSEC HIDS and a Splunk universal forwarder. The OSSEC HIDS configurations are identical across all the implementations and have log, malware, and file integrity monitoring enabled. The Splunk universal forwarders are configured with all default inputs enabled and to transmit syslog to a Splunk unified indexer and search head for collection and analysis. On Docker container deployments, the Monitoring Docker Splunk App, installed on the Splunk forwarder, facilitates intercontainer and host telemetry collection.

All test hosts serve the Damn Vulnerable Web App (DVWA), which will be the primary target for assessing the efficacy of the various intrusion detection and analysis systems. Furthermore, Security Onion 14.04 is deployed in the lab with the Snort NIDS, OSSEC HIDS, Bro for traffic monitoring, and ELSA, Squert, Wireshark, and associated tools for analysis. Security Onion enables the efficacy assessments of the intrusion detection and analysis experiments conducted in the traditional application host environment, as well as the application of traditional NIDS and HIDS in the Docker host and containerized application environment. In implementations covered by Security Onion, the Snort NIDS and OSSEC HIDS configurations are identical and have all rules enabled. Wazuh with the OSSEC HIDS and Sysdig Falco with the falco-probe host kernel module, for tapping and assessing Linux container host and intra-container activities, enable the efficacy assessments of the container intrusion detection and analysis use cases.

Once the lab infrastructure was deployed and configured, the attack experiments were conducted from a Kali Linux host. The attack experiments represent various phases of the Cyber Kill-Chain (Lockheed Martin), and they serve to assess the intrusion detection and analysis capabilities of the various systems. The attack types, test cases, and required capabilities are located in the appendix. Testing artifacts were collected from the various intrusion detection and analysis systems. The artifacts and testing results serve to measure the effectiveness and capabilities of the multiple systems to detect and enable analysis of the various attacks and intrusions.

Table 1. Damn Vulnerable Web App Hosted on Traditional Virtual Machine and Protected by Security Onion. ***(Source: Author)***

Attack Phase Detection, Ca	Test Cases	Outcome	Score
Scanning Detection	Sparta scan with nmap	Snort detected scan	3
Scanning Detection	Nikto Web App Scan	Snort detected scan	3
Scanning Detection	NMAP host scan intense plus UDP	Snort detected scan	3
Scanning Detection	NMAP host scan stealth (SYN scan)	Neither Snort nor OSSEC detected nmap stealth scan	1
Scanning Detection	Internal network scan intense	Neither Snort nor OSSEC detected	1
Scanning Detection	Host vulnerability scan Nessus basic	Snort detected scan	3
Scanning Detection	Host vulnerability scan Nessus WebApp Scan	Snort detected scan	3
App Attack Detection	Conduct SQL injection attack	Snort detected scan	3
App Attack Detection	Conduct authentication and session management attack	Snort detected scan	2
App Attack Detection	Conduct XSS attack reflected	Snort detected scan	3
Malware Detection	Deploy malicious payload to host	Neither Snort nor OSSEC detected	1
Malware Detection	Execute malicious payload on host	Neither Snort nor OSSEC detected	1
C2 Detection	Execute C2 activity on host	Neither Snort nor OSSEC detected	1
Privilege Escalation Detection	Execute privilege escalation on host	Snort detected	3
Data Exfiltration Detection	Conduct data exfiltration	Neither Snort nor OSSEC detected	1
File Integrity Detection	Alter sensitive files and check FIM for alerts (registry, conf files, password files, system files, user	OSSEC detected	2
System Information Leakage	Check for detection of leaked system data (resource usage, location services)	Snort detected	2
Auto Anomaly Detection	Check for automated alerting of suspected suspicious behavior - execute potentially malicious	Snort detected	2
Attacker - Victim Relation	Check for relationship mapping between attacker and victim	Attacker victim auto detected and	3
Forensic Artifact Retrieval	Check for capabilities to retrieve forensic artifacts (logs, pcaps, flows, files)	Capable	3
Total Points			**44**

Table 2. Damn Vulnerable Web App Hosted in a Docker Container and Protected by Security. Onion. ***(Source: Author)***

Attack Phase	Test Cases	Outcome	Score
Scanning Detection	Sparta scan with nmap	Snort detected scan	3
Scanning Detection	Nikto web app scan	Snort detected scan	3
Scanning Detection	NMAP host scan intense plus UDP	Snort detected	2
Scanning Detection	NMAP host scan stealth (SYN scan)	Neither Snort nor OSSEC detected	1
Scanning Detection	Internal network scan intense	Neither Snort nor OSSEC detected	1
Scanning Detection	Host vulnerability scan Nessus basic	Snort detected scan. OSSEC did not.	3
Scanning Detection	Host vulnerability scan Nessus WebApp Scan	Snort detected scan. OSSEC did not.	3
App Attack Detection	Conduct SQL injection attack	Snort detected scan. OSSEC did not.	2
App Attack Detection	Conduct authentication and session management attack	Snort detected scan. OSSEC did not.	2
App Attack Detection	Conduct XSS attack reflected	Snort detected scan. OSSEC did not.	3
Malware Detection	Deploy malicious payload to host	Neither Snort nor OSSEC detected	1
Malware Detection	Execute malicious payload on host	Neither Snort nor OSSEC detected	1
C2 Detection	Execute C2 activity on host	Neither Snort nor OSSEC detected	1
Privilege Escalation Detection	Execute privilege escalation on host	Neither Snort nor OSSEC detected	1
Data Exfiltration Detection	Conduct data exfiltration	Neither Snort nor OSSEC detected	1
File Integrity Detection	Alter sensitive files and check FIM for alerts (registry, conf files, password files, system files, user	OSSEC detected	2
System Information Leakage	Check for detection of leaked system data (resource usage, location services)	Snort detected	2
Auto Anomaly Detection	Check for automated alerting of suspected suspicious behavior - execute potentially malicious activity	Snort detected	2
Attacker - Victim Relation	Check for relationship mapping between attacker and victim	Attacker victim auto detected and	3
Forensic Artifact Retrieval	Check for capabilities to retrieve forensic artifacts (logs, pcaps, flows, files)	Capable	3
Total Points			**40**

Effectiveness Criteria

The effectiveness of the various intrusion detection and analysis systems are measured against the following criteria and associated test cases: Note: The associated test cases are located in the appendix.

1. Detection of scanning activity
2. Detection of application attacks
3. Detection of malware deployment
4. Detection of malware execution
5. Detection of malicious command and control
6. Detection of malicious privilege escalation
7. Detection of malicious data exfiltration
8. Detection of file integrity violations
9. Detection of leaked system data
10. Auto-detection of anomalous behavior
11. Auto-detection of attacker, victim, infrastructure relationship
12. Capability for forensic artifact retrieval (PCAP, Flow, Logs,)

Measurement Criteria

A scoring system is used to measure the effectiveness of the intrusion detection systems to detect and provide analysis capabilities of the associated test case experiments. Each test case experiment will have a maximum of three points awarded.

Points are weighted as follows:

- **One Point:** Not Effective (Method did not work).
- **Two Points:** Moderately Effective (Method worked, but did not allow for complete functionality, or equivalent to traditional network implementation).
- **Three Points:** Effective (Method worked as effectively as traditional network implementation).

The point-based measurements of effectiveness will describe the efficacy of intrusion detection and analysis methods as applied in container networks, compared to the effectiveness of similar methods employed in traditional networks. Also, the findings of this research and the scoring of the effectiveness criteria could aid in the identification and development of new methods for securing container networks.

RESEARCH FINDINGS – THE ANSWERS TO LIFE, THE UNIVERSE, AND EVERYTHING

The following are the effectiveness results and analysis of the various intrusion detection and analysis methods assessed. Note: Where applicable, the NIDS and HISD configurations are identical and vary only in implementation or capabilities provided by the analysis platforms, such as Security Onion, Splunk, or Wazuh.

In this use case, Security Onion was deployed with the Snort network-based intrusion detection system with the Emerging Threats ruleset completely enabled, and the OSSEC host-based intrusion detection system on the protected virtual machine application host.

For the scanning portion of the tests, Snort detected all but the Nmap stealth and network range scans. OSSEC did not detect the Nessus host and web application vulnerability scans, or Nmap scans during the software service and version enumeration portions of the scans.

For the attack portion of the tests, Snort detected all the attacks. However, Snort only detected the authentication and session management attack via the curl detection policy which triggered when curl was used to pull the session ID token from DVWA. For this, I subtracted one point. OSSEC did not detect any of the attacks.

Table 3. Damn Vulnerable Web App Hosted in a Docker container and Protected by Wazuh.

Attack Phase	Test Cases	Outcome	Score
Scanning Detection	Sparta scan with nmap	OSSEC detected	3
Scanning Detection	Nikto web app scan	OSSEC did not detect	1
Scanning Detection	NMAP host scan intense plus UDP	OSSEC detected	3
Scanning Detection	NMAP host scan stealth (SYN scan)	OSSEC did not detect	1
Scanning Detection	Internal network scan intense	OSSEC detected	3
Scanning Detection	Host vulnerability scan Nessus basic	OSSEC detected	3
Scanning Detection	Host vulnerability scan Nessus WebApp Scan	OSSEC did not detect	1
App Attack Detection	Conduct SQL injection attack	OSSEC did not detect	1
App Attack Detection	Conduct authentication and session management attack	OSSEC did not detect	1
App Attack Detection	Conduct XSS attack reflected	OSSEC did not detect	1
Malware Detection	Deploy malicious payload to host	OSSEC detected	2
Malware Detection	Execute malicious payload on host	OSSEC did not detect	1
C2 Detection	Execute C2 activity on host	OSSEC did not detect	1
Privilege Escalation Detection	Execute privilege escalation on host	OSSEC detected	3
Data Exfiltration Detection	Conduct data exfiltration	OSSEC did not detect	1
File Integrity Detection	Alter sensitive files and check FIM for alerts (registry, conf files, password files, system files, user data)	OSSEC detected	3
System Information Leakage	Check for detection of leaked system data (resource usage, location services)	OSSEC detected	1
Auto Anomaly Detection	Check for automated alerting of suspected suspicious behavior - execute potentially malicious activity	Capable	3
Attacker - Victim Relation	Check for relationship mapping between attacker and victim	Capable	3
Forensic Artifact Retrieval	Check for capabilities to retrieve forensic artifacts (logs, pcaps, flows, files)	Moderately Capable	2
Total Points			**38**

For the malware portion of the tests, neither Snort nor OSSEC detected the downloading of the EICAR test file nor the execution of the EICAR payload in a shell script.

Neither Snort nor OSSEC detected the command and control activities that were conducted on the victim host using both SSH and Netcat.

Snort detected privilege escalation. OSSEC did not detect privilege escalation attempts on the victim host.

Neither Snort nor OSSEC detected the exfiltration of the passwd and shadow files from the protected /etc/ directory.

OSSEC detected file integrity modifications in protected directories. However, one point was subtracted due to Security Onion not surfacing the alerts automatically or in real time via Sguil. Hunting was required to find the associated alerts in ELSA. Snort did not detect file integrity attacks.

Table 4. Damn Vulnerable Web App Hosted in a Docker container and Protected by Sysdig Falco.

Attack Phase	Test Cases	Outcome	Score
Scanning Detection	Sparta scan with nmap	Detected	3
Scanning Detection	Nikto web app scan	Detected	3
Scanning Detection	NMAP host scan intense plus UDP	Detected	3
Scanning Detection	NMAP host scan stealth (SYN scan)	Not Detected	1
Scanning Detection	Internal network scan intense	Detected	3
Scanning Detection	Host vulnerability scan Nessus basic	Detected	3
Scanning Detection	Host vulnerability scan Nessus WebApp Scan	Detected	3
App Attack Detection	Conduct SQL injection attack	Detected	2
App Attack Detection	Conduct authentication and session management attack	Detected	2
App Attack Detection	Conduct XSS attack reflected	Detected	2
Malware Detection	Deploy malicious payload to host	Not Detected	1
Malware Detection	Execute malicious payload on host	Not Detected	1
C2 Detection	Execute C2 activity on host	Not Detected	1
Privilege Escalation Detection	Execute privilege escalation on host	Detected	3
Data Exfiltration Detection	Conduct data exfiltration	Not Detected	1
File Integrity Detection	Alter sensitive files and check FIM for alerts (registry, conf files, password files, system files, user data)	Detected	3
System Information Leakage	Check for detection of leaked system data (resource usage, location services)	Not Detected	1
Auto Anomaly Detection	Check for automated alerting of suspected suspicious behavior - execute potentially malicious activity	Detected	3
Attacker - Victim Relation	Check for relationship mapping between attacker and victim	Detected	2
Forensic Artifact Retrieval	Check for capabilities to retrieve forensic artifacts (logs, pcaps, flows, files)	Capable	2
Total Points			**43**

Snort detected the leakage of certain system information such as software names and version numbers. However, one point was subtracted due to Security Onion not surfacing the alerts automatically or in real-time via Sguil. Hunting was required to find the associated alerts in ELSA. OSSEC did not detect system information leakage.

Sguil automatically surfaced Snort detections of potentially anomalous behavior. However, one point was subtracted due to Security Onion not surfacing the associated OSSEC alerts automatically or in real time via Sguil. Hunting was required to find the associated alerts in ELSA.

Security Onion was able to efficiently and dynamically depict attacker to victim relationships via collected telemetry.

Security Onion was able to produce logs, pcaps, flow data, and associated files. Of the intrusion detection and analysis platforms evaluated, Security Onion with the Snort NIDS and OSSEC HIDS deployed to protect a traditional virtual machine application host was the most effective platform and received a score of 44 points.

In this use case, Security Onion was deployed with the Snort network-based intrusion detection system with the Emerging Threats ruleset completely enabled, and the OSSEC host-based intrusion detection system on the protected Docker application container host.

For the scanning portion of the tests, Snort detected all but the Nmap stealth and network range scans. OSSEC did not detect the Nessus host and web application vulnerability scans, or Nmap scans during the software service and version enumeration portions of the scans.

For the attack portion of the tests, Snort detected all the attacks. However, Snort only detected the authentication and session management attack via the curl detection policy which triggered when curl was used to pull the session ID token from DVWA.

Furthermore, Security Onion did not surface the associated SQL injection attack alert automatically or in real time via Sguil. Hunting was required to find the associated alerts in ELSA. For these two deficiencies, one point per attack was deducted. OSSEC did not detect any of the attacks.

For the malware portion of the tests, neither Snort nor OSSEC detected the downloading of the EICAR test file nor the execution of the EICAR payload in a shell script.

Neither Snort nor OSSEC detected the command and control activities that were conducted on the victim host using both SSH and Netcat.

Neither Snort nor OSSEC detected the privilege escalation attempts on the victim host.

Neither Snort nor OSSEC detected the exfiltration of the passwd and shadow files from the protected /etc/ directory.

OSSEC detected file integrity modifications in protected directories. However, one point was subtracted due to Security Onion not surfacing the alerts automatically or in real-time via Sguil. Hunting was required to find the associated alerts in ELSA. Snort did not detect file integrity attacks.

Snort detected the leakage of certain system information such as software names and version numbers. However, one point was subtracted due to Security Onion not surfacing the alerts automatically or in real-time via Sguil. Hunting was required to find the associated alerts in ELSA. OSSEC did not detect system information leakage.

Sguil automatically surfaced Snort detections of potentially anomalous behavior.

However, one point was subtracted due to Security Onion not surfacing the associated OSSEC alerts automatically or in real time via Sguil. Hunting was required to find the associated alerts in ELSA.

Security Onion was able to efficiently and dynamically depict attacker to victim relationships via collected telemetry.

Security Onion was able to efficiently produce logs, pcaps, flow data, and associated files.

Of the intrusion detection and analysis platforms evaluated, Security Onion with the Snort NIDS and OSSEC HIDS deployed to protect a Docker application container host and workloads was the second most effective platform and received a score of 40 points.

In this use case, Wazuh was deployed with the OSSEC host-based intrusion detection system on the protected Docker application container host, and the Wazuh PCI DSS extension enabled.

For the scanning portion of the tests, OSSEC detected all but the Nikto and Nessus web application scans and the Nmap stealth scan.

"Wazuh was able to efficiently and dynamically depict attacker to victim relationships via collected telemetry"

For the attack portion of the tests, OSSEC did not detect any of the attacks.

For the malware portion of the tests, OSSEC detected the placement of the EICAR payload shell script in the protected /etc/ directory. However, it is unlikely that OSSEC would have detected, in real time, the test malware file if it was deposited and executed from a non-protected directory. OSSEC, as configured on all the test hosts, conducts daily malware checks.

- OSSEC did not detect the execution of the EICAR payload shell script.
- OSSEC did not detect the command and control activities that were conducted on the victim host using both SSH and Netcat.
- OSSEC detected the privilege escalation attempts on the victim host via the Wazuh PCI DSS extension.
- OSSEC did not detect the exfiltration of the passwd and shadow files from the protected /etc/ directory.
- OSSEC detected file integrity modifications in protected directories.
- OSSEC did not detect the leakage of certain system information such as software names and version numbers.
- OSSEC automatically surfaced potentially anomalous behavior.

Wazuh was able to efficiently and dynamically depict attacker to victim relationships via collected telemetry.

Wazuh was only capable of producing limited alert and log reports. Wazuh was unable to produce specific logs, pcaps, flow data, and associated files.

Of the intrusion detection and analysis platforms evaluated, Wazuh with the OSSEC HIDS deployed to protect a Docker application container host and workloads was the least effective platform and received a score of 38 points.

In this use case, Sysdig Falco was deployed with the falco-probe Linux kernel module on the Docker host. The falco-probe kernel module facilitates the tapping of bidirectional container host to container and container to container system call communications. Furthermore, Falco is a headless application that can surface alerts to numerous output destinations such as standard output, syslog, flat files, and local programs. In this use case, Falco alerts, and telemetry was sent to a central Splunk instance via a Splunk universal forwarder and the Monitoring Docker Splunk app installed on the test host. All intrusion analysis was done via Splunk.

For the scanning portion of the tests, Falco detected all but the Nmap stealth scan.

Falco did not detect any of the attacks. One point was subtracted per test case due to the alerts surfacing through log management capabilities in the Monitoring Docker Splunk app used in the falco implementation.

- Falco did not detect any of the malware test cases.
- Falco did not detect the command and control activities that were conducted on the victim host using both SSH and Netcat.
- Falco detected the privilege escalation attempts on the victim host.
- Falco did not detect the exfiltration of the passwd and shadow files

from the protected /etc/ directory.

- Falco detected file integrity modifications in protected directories.
- Falco did not detect the leakage of certain system information such as software names and version numbers.
- Falco automatically surfaced potentially anomalous behavior.

Falco was not able to efficiently and dynamically depict attacker to victim relationships. One point was subtracted due to associated correlations surfacing through the log management capabilities in the Monitoring Docker Splunk app used in the falco implementation.

Falco was only capable of producing limited alert and log reports. One point was subtracted due to Falco's inability to produce specific logs, pcaps, flow data, and associated files.

Of the intrusion detection and analysis platforms evaluated, Sysdig Falco with the falco-probe kernel module and Monitoring Docker for Splunk app deployed to protect a Docker application container host and workloads was the most effective platform and received a score of 43 points.

WHAT NOW – RECOMMENDATIONS AND IMPLICATIONS FOR SECURITY AND A BETTER TOMORROW

The research presented in this article indicates that while technology can do much to enable security, it can also do much to hinder security and introduce vulnerabilities. As such, experienced security professionals skilled in their tools, tactics, and procedures are paramount to security. Defense in depth is still critical to security. This research indicates that no one security technology, nor single security platform can detect all the attacks, vulnerabilities, and threats to an environment.

Capability, capacity, configuration, and implementation architecture define security coverage. If the security tooling deployed and implemented is incapable, misconfigured, or deployed in a position of incomplete coverage, it will not be effective.

Furthermore, exclusive reliance on the fidelity and capability of security tooling to prevent, detect, and surface all attacks, vulnerabilities, and threats present in an environment, even if correctly configured and implemented, is unrealistic and unwise. Proactive threat hunting and centralized log management are required to mitigate the tool capability gap. The capability gap was demonstrated in the research in instances where attack experiments resulted in telemetry that was not surfaced as an alert in the security tooling user interfaces but instead was detected in the SIEM or NSM.

Vulnerability assessments of application containers and their associated images are essential to overall container environment security. By integrating purpose-built container and image vulnerability scanning into the continuous integration and continuous deployment (CI/CD) pipeline, security professionals can dynamically detect when vulnerabilities are introduced into the images used to create containers and into the software packages, application logic, and dependencies used when presenting the applications. With this capability, security professionals can then remediate or mitigate the discovered vulnerabilities.

> *"while technology can do much to enable security, it can also do much to hinder security and introduce vulnerabilities"*

Recommendations for More Effective IDS solutions in Application Container Environments

Hardening, instrumenting, monitoring, and segmenting application container hosts and management platforms are critical to container environment security. The Center for Internet Security publishes security configuration benchmarks for the most common Linux operating systems and web servers used in container implementations. Furthermore, CIS also published benchmarks for both the community and enterprise versions of Docker. The CIS benchmarks are located here: https://www.cisecurity.org/cis-benchmarks/

This research indicates that instrumenting application container hosts with security tooling is critical. As such, host-based systems such as Sysdig Falco with its Linux kernel module that can monitor system calls between the host and containers to detect malicious activities is key to container environment security. The research also indicates that monitoring application container hosts with non-kernel module HIDS, such as those relying on Linux Audit, is also useful. However, in-depth analysis of container host and intra-container communications are only possible with kernel level tapping modules.

Hand-in-hand with proper instrumentation is active monitoring of container environments by experienced and skilled security professionals. Application container deployments introduce even more complexity and telemetry into environments than traditional network implementations. Furthermore, as described in the research findings, even when telemetry is generated in container networks and ingested into security platforms, alerts are not guaranteed to be produced or surfaced. In these instances, hunting conducted by security professionals is crucial to the prevention, detection, alerting, response, and remediation of associated vulnerabilities, threats, attacks, and intrusions.

Appropriate segmentation of application container networks can also assist in intrusion detection and analysis. Due to the typically high deployment densities of containerized applications on hosts, and the complex orchestration of containerized workloads, non-standard network port assignments are common in container environments. This complexity makes traditional network firewall and intrusion detection impractical for securing individual containerized workloads. However, segmenting application container hosts within secured networks and then deploying traditional network firewalls and intrusion detection systems can aid in securing the overall container environment by restricting access to the network and alerting when unusual activity occurs. Furthermore, implementing container-aware web application firewalls that can dynamically associate container instances with application traffic and network port assignments can help overall security.

Implications for Future Research

The practice of application container security is ripe for research. For instance, one of the most recent and compelling technologies developed to secure web applications is RASP, or runtime application self-protection. RASP is built into the application and is executed at runtime allowing for the detection and response of malicious activities at the application layer. At this time, RASP technologies are restricted to web application deployments based a limited set of webservers and custom application runtime environments. However, RASP technology is promising and developing rapidly. Furthermore, RASP applied to containerized applications is nascent and prime for development. RASP presents exciting and potentially valuable opportunities for future research.

Container network-based intrusion detection is also prime for future research. By solving for dynamic application container behavior profiling and network application port mapping, advances in container firewalls have set the stage for the development of container NIDS. Especially compelling is the potential value in combining data and information gained from container HIDS, with container network security telemetry generated by application and network aware container firewalls, to facilitate the development of container NIDS.

"By solving for dynamic application container behavior profiling and network application port mapping, advances in container firewalls have set the stage for the development of container NIDS"

Another point of future research is the development of machine-learning applications to facilitate the development of active container intrusion detection and analysis systems. The dynamic nature of containerized application development and operations makes securing these environments difficult, especially when operating under traditional security paradigms. As such, automation provided by machine learning can augment security operations. Methods, such as Bag of System Calls, briefly covered in this research, can provide such assistance. Using machine learning systems such as BoSC, security tooling and procedures can be developed to automatically detect, alert, and respond to unusual and potentially malicious activities and conditions.

CONCLUSION

Application container technologies are evolving rapidly, their adoption into the enterprise is soaring, and the implementation use cases are growing in proportion, criticality, and complexity. Furthermore, the vulnerabilities introduced by application container implementations and the attacks being developed to exploit the vulnerabilities are also evolving rapidly. Combine this landscape with the rapid digital transformation of business processes and the widespread adoption of public cloud technologies, commonly used to host containerized applications, and the necessity to develop effective container intrusion detection and analysis systems become evident. As the research suggests, no one security platform was able to secure the whole container environment. It appears that securing application container environments both at the network and at the host-level is key to effective security. Furthermore, centralized collection and analysis of container network and host telemetry were beneficial to the security of the environments tested.

The research presented here is limited to assessing the effectiveness of methods for conducting intrusion detection and analysis in Docker Linux application container networks when compared with the efficacy of similar methods in traditional networks. For this purpose, Security Onion with the familiar Snort NIDS and OSSEC HIDS, Wazuh with the OSSEC HIDS, and Sydig Falco, with its kernel tapping module were selected. This research attempted to remove biases by scoring against absolute effectiveness, absolute ineffectiveness, and moderate effectiveness. However, moderate effectiveness can be judged subjectively due to the assessor's definition of the term. While not exhaustive, this research resented experiments which were representative of typical attack types depicted in the Cyber Kill-Chain.

Furthermore, the techniques and tools utilized during the experiments are representative of those commonly used by security professionals when plying their trade. In sum, this research aimed to identify gaps in current knowledge and capabilities available to secure application container networks and to spur the development of new research, techniques, and technologies to secure such environments.

REFERENCES

[13] Abed, A. S., Clancy, C., & Levy, D. S. (2015). Intrusion Detection System for Applications Using Linux Containers. *Security and Trust Management Lecture Notes in Computer Science*, 123-135. doi:10.1007/978-3-319-24858-5_8

[14] Alonso, A. A. (n.d.). Intrusion Detection Through Relationship Analysis. Retrieved August 22, 2016, from https://www.sans.org/reading-room/whitepapers/detection/intrusiondetection-relationship-analysis-37352 Accessed from the SANS Reading Room

[15] Bejtlich, R. (2013). *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. San Francisco: No Starch Press.

[16] Bosco, P. (2016, January 20). Intrusion Detection and Prevention Systems Cheat Sheet: Choosing the Best Solution, Common Misconfigurations, Evasion Techniques, and Recommendations. Retrieved February 22, 2017, from https://www.sans.org/readingroom/whitepapers/detection/intrusion-detection-prevention-systems-cheat-sheetchoosing-solution-common-misconfigurations-evasion-techniques-recommendations-36677 Accessed from the SANS Reading Room

[17] Bottomley, J., & Emelyanov, P. (2014). Operating Containers. USENIX, 39(5). Retrieved December 11, 2017, from https://www.usenix.org/system/files/login/articles/login_1410_02-bottomley.pdf

[18] Goyal, P. (2017, July 6). CIS Docker Community Edition Benchmark [PDF]. East Greenbush: Center for Internet Security.

[19] Davidoff, S., & Ham, J. (2012). Network Forensics Tracking Hackers Through Cyberspace. Upper Saddle River: Prentice Hall.

[20] Gao, X., Gu, Z., Kayaalp, M., Pendarakis, D., & Wang, H. (2017, June). ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds [PDF]. Williamsburg: College of William and Mary. Presented at the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks

[21] Hayden, M. (2015, July 26). Securing Linux Containers. Retrieved February 23, 2017, from https://www.sans.org/reading-room/whitepapers/linux/securing-linux-containers-36142 Accessed from the SANS Reading Room

[22] Hosburgh, M. (n.d.). Offensive Intrusion Analysis: Uncovering Insiders with Threat Hunting and Active Defense. Retrieved July 6, 2017, from https://www.sans.org/readingroom/whitepapers/detection/offensive-intrusion-analysis-uncovering-insiders-threathunting-active-defense-37885 Accessed from the SANS Reading Room

[23] HOW TO SECURELY CONFIGURE A LINUX HOST TO RUN CONTAINERS [PDF]. (2017). San Francisco: Twistlock. Accessed from https://www.twistlock.com/resources/securely-configure-linux-host-runcontainers/

[24] Hykes, S. (Writer). (2017, November 20). The future of Linux Containers. Live performance in Pycon U.S. 2013: Santa Clara Convention Center, Santa Clara.

[25] Kennedy, C. (2016, June 29). Deception Techniques as Part of Intrusion Detection Strategy. Retrieved February 22, 2017, from https://www.sans.org/readingroom/whitepapers/detection/deception-techniques-intrusion-detection-strategy-37140 Accessed from the SANS Reading Room

[26] Lockheed Martin Corporation. (n.d.). T*he Cyber Kill Chain* [Brochure]. Author. Retrieved October 23, 2018, from https://www.lockheedmartin.com/us/what-wedo/aerospace-defense/cyber/cyber-kill-chain.html

[27] Mohallel, A. A., Bass, J. M., & Dehghantaha, A. (2016). Experimenting with docker: Linux container and base OS attack surfaces. 2016 International Conference on Information Society (i-Society). doi:10.1109/i-society.2016.7854163

[28] Petazonni, J. (2015, August 15). Anatomy of a Container: Namespaces, cgroups & Some Filesystem Magic. Retrieved July 30, 2017, from https://www.slideshare.net/jpetazzo/anatomy-of-a-container-namespaces-cgroups-somefilesystem-magic-linuxcon Presentation at LinuxCon 2015

[29] Robinson, A. (2016, November 18). A Checklist for Audit of Docker Containers. Retrieved February 23, 2017, from https://www.sans.org/readingroom/whitepapers/auditing/checklist-audit-docker-containers-37437 Accessed from the SANS Reading Room

[30] Souppaya, M., Morello, J., & Scarfone, K. (n.d.). Draft (2nd) NIST Special Publication 800-190 Application Container Security Guide (USA, NIST). Retrieved July 13, 2017, from http://csrc.nist.gov/publications/drafts/800-190/sp800-190-draft2.pdf

[31] Winkel, S. (2016, November 18). Security Assurance of Docker Containers. Retrieved February 23, 2017, from https://www.sans.org/reading-room/whitepapers/assurance/securityassurance-docker-containers-37432 Accessed from the SANS Reading Room

[32] Winkel, S. (2017, July 9). Forensicating Docker with ELK. Retrieved July 30, 2017, from https://www.sans.org/reading-room/whitepapers/forensics/forensicating-docker-elk-37870 Accessed from the SANS Reading Room

ABOUT THE AUTHOR

ALFREDO HICKMAN leads cybersecurity product engineering at Rackspace's MSSP, better known as Rackspace Managed Security. Before working at Rackspace, Alfredo held various technical and leadership roles at various U.S. Department of Defense contracting companies. Alfredo is also a veteran of the U.S. Marine Corps, where he was an infantry sergeant, personal security detachment team-leader, and operations and training NCO. Alfredo holds a Bachelor of Arts degree in Political Science with highest honors from the University of Texas at San Antonio, and he is a candidate for the Master of Science degree in Information Security Engineering from the SANS Technology Institute.

A NEW REALITY: MODELLING & SIMULATION AS A SERVICE

By: Dr. Robert Siegfried, Chris McGroarty, and Tom van den Berg,

__NATO and nations use simulation environments for various purposes__, such as training, capability development, mission rehearsal and decision support in acquisition processes. Consequently, Modelling and Simulation (M&S) has become a critical capability for the alliance and its nations. M&S products are highly valuable resources and it is essential that M&S products, data and processes are conveniently accessible to a large number of users as often as possible. However, achieving interoperability between simulation systems and ensuring credibility of results currently requires large efforts with regards to time, personnel and budget.

Recent developments in cloud computing technology and service-oriented architectures offer opportunities to better utilize M&S capabilities in order to satisfy NATO critical needs. M&S as a Service (MSaaS) is a new concept that combines service orientation and the provision of M&S applications via the as-a-service model of cloud computing to enable more composable simulation environments that can be deployed and executed on-demand. The MSaaS paradigm supports stand-alone use as well as integration of multiple simulated and real systems into a unified cloud-based simulation environment whenever the need arises.

NATO Modelling & Simulation Group MSG-136 ("Modelling and Simulation as a Service – Rapid deployment of interoperable and credible simulation environments") investigated MSaaS with the aim of providing the technical and organizational foundations to establish the *Allied Framework for M&S as a Service* within NATO and partner nations. The *Allied Framework for M&S as a Service* is the common approach of NATO and nations towards implementing MSaaS and is defined by the following documents:

- Operational Concept Document
- Technical Reference Architecture
- Governance Policies

MSG-136 evaluated the MSaaS concept through various experiments and extensive gathering of stakeholder requirements and their assessment. The experimentation results and initial operational applications demonstrate that MSaaS is capable of realizing the vision that M&S products, data and processes are conveniently accessible to a large number of users whenever and wherever needed. MSG-136 strongly recommends NATO and nations to advance and to promote the operational readiness of M&S as a Service, and to conduct required Science & Technology efforts to close current gaps.

This article provides an overview of the MSG-136 results, the consolidated point of view of the global stakeholders, and outlines the way forward. From 2018-2021 the initial concepts are extended by MSG-164 (i.e., specification of issues and challenges not yet addressed) and validated through regular exercise participation and dedicated evaluation events. For this purpose, MSG-164 will focus on two main work streams:

1. To advance and to promote the operational readiness of M&S as a Service.
2. To investigate critical research and development topics to further enhance the benefits of M&S as a Service.

INTRODUCTION

Background

NATO and the nations use distributed simulation environments for various purposes, such as training, mission rehearsal, and decision support in acquisition processes. Consequently, modeling and simulation (M&S) has become a critical technology for the coalition and its nations. Achieving interoperability between participating simulation systems and ensuring credibility of results currently requires often enormous efforts with regards to time, personnel, and budget.

The NATO Modelling and Simulation Group (NMSG) is part of the NATO Science and Technology Organization (STO). The mission of the NMSG is to promote cooperation among Alliance bodies, NATO, and partner nations to maximize the effective utilization of M&S. Primary mission areas include: M&S standardization, education, and associated science and technology. The NMSG mission is guided by the NATO Modelling and Simulation Masterplan (NMSMP) [1]. The NMSMP vision is to "Exploit M&S to its full potential across NATO and the Nations to enhance both operational and cost effectiveness". This vision will be achieved through a cooperative effort guided by the following principles:

- Synergy: leverage and share the existing NATO and national M&S capabilities.
- Interoperability: direct the development of common M&S standards and services for simulation interoperability and foster interoperability between Command & Control (C2) and simulation.
- Reuse: Increase the visibility, accessibility, and awareness of M&S assets to foster sharing across all NATO M&S application areas.

The NMSG is the Delegated Tasking Authority for NATO

M&S interoperability standards. This is the rationale for the close relationship between NMSG and the Simulation Interoperability Standards Organization (SISO), which was formalized in a Technical Cooperation Agreement signed in July 2007.

Recent technical developments in the area of cloud computing technology and service oriented architecture (SOA) may offer opportunities to better utilize M&S capabilities in order to satisfy NATO critical needs. A new concept that includes service orientation and the provision of M&S applications via the as-a-service model of cloud computing may enable composable simulation environments that can be deployed rapidly and on-demand. This new concept is known as M&S as a Service (MSaaS).

NATO MSG-136 ("Modelling and Simulation as a Service – Rapid deployment of interoperable and credible simulation environments") [2] is one of the technical working groups under the NMSG. This group investigated the new concept of MSaaS with the aim of providing the technical and organizational foundations for a future permanent service-based Allied Framework for MSaaS within NATO and partner nations. NATO MSG-136 started its three-year term of work in November 2014 and finished in November 2017. MSaaS is looking to provide a strategic approach to deliver simulation coherently against the NMSMP vision and guiding principles.

This paper provides an overview of the activities performed by MSG-136 and presents the results achieved, from the following perspectives:

- *Operational concept* of MSaaS: how it works from the user point of view;
- *Technical concept* of MSaaS: reference architecture, services metadata, and engineering process;
- *Governance concept* and roadmap for MSaaS within NATO.

Terminology

M&S products are highly valuable to NATO and military organizations and it is essential that M&S products, data and processes are conveniently accessible to a large number of users as often as possible. Therefore a new M&S ecosystem is required where M&S products can be accessed simultaneously and spontaneously by a large number of users for their individual purposes. This "as a Service" paradigm has to support stand-alone use as well as integration of multiple simulated and real systems into a unified simulation environment whenever the need arises.

This article uses the term service always in the sense of M&S service, unless stated otherwise, using the following definition:

> An **M&S service** is a specific M&S-related capability delivered by a provider to one or more consumers according to well defined contracts including service level agreements (SLA) and interfaces. The provided capability is implemented in a (distributed) system and/or organization.

> **M&S as a Service (MSaaS)** is an enterprise-level approach for discovery, composition, execution and management of M&S services.

Allied Framework for MSaaS

The Allied Framework for MSaaS is the common approach of NATO and Nations towards implementing MSaaS and is defined by the following documents:

- *Operational Concept Document*: The Operational Concept Document (OCD) describes the intended use, key capabilities and desired effects of the Allied Framework for MSaaS from a user's perspective.
- *Technical Reference Architecture*: The Technical Reference Architecture describes the architectural building blocks and patterns for realizing MSaaS capabilities.
- *Governance Policies*: The Governance Policies identify

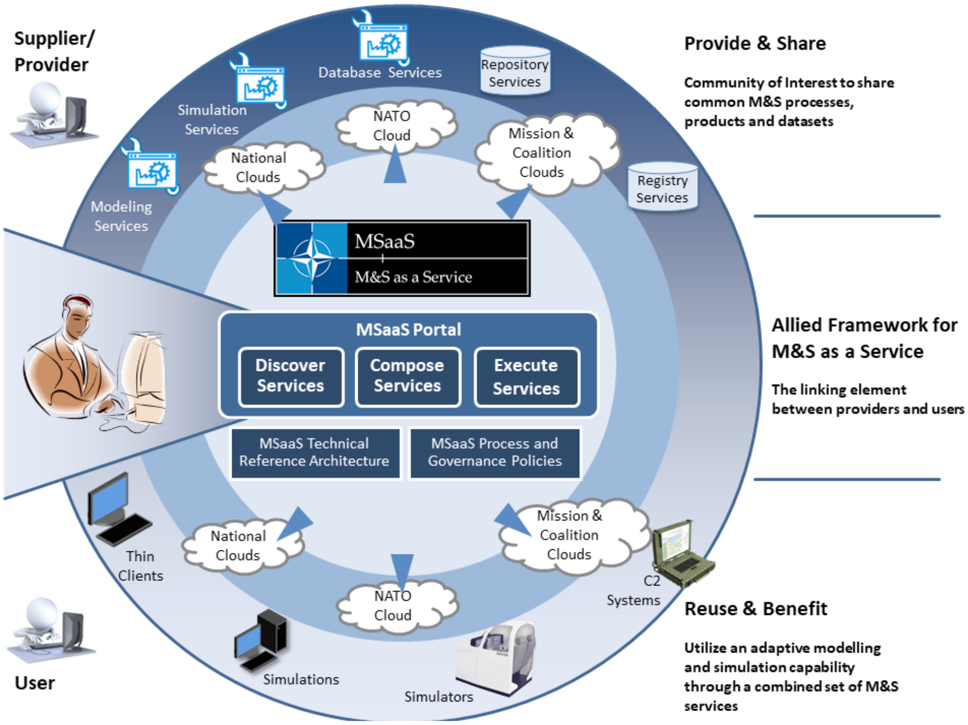

Supplier/
Provider
Database Services
Repository Services
Simulation Services
NATO Cloud
Mission & Coalition Clouds
National Clouds
Registry Services
Modeling Services
MSaaS
M&S as a Service
MSaaS Portal
Discover Services
Compose Services
Execute Services
MSaaS Technical Reference Architecture
MSaaS Process and Governance Policies
National Clouds
Mission & Coalition Clouds
Thin Clients
NATO Cloud
C2 Systems
Simulations
Simulators
User
Provide & Share
Community of Interest to share common M&S processes, products and datasets
Allied Framework for M&S as a Service
The linking element between providers and users
Reuse & Benefit
Utilize an adaptive modelling and simulation capability through a combined set of M&S services

Figure 1: MSaaS concept. ***(Source Author)***

MSaaS stakeholders, relationships and provide guidance for implementing and maintaining the Allied Framework for MSaaS.

The above mentioned documents define the blueprint for individual organizations to implement MSaaS. However, specific implementations – i.e. solutions – may be different for each organization.

Document Overview

This article is structured as follows:

- **Section 2** discusses the Operational Concept for the Allied Framework for MSaaS. The purpose of the operational concept is to inform relevant stakeholders how the framework will function in practice. The capabilities and key characteristics of the proposed framework are discussed as well as the interactions of the users.
- **Section 3** presents the technical concept of the Allied Framework for MSaaS. The technical concept is described in three volumes: Reference Architecture, Services Discovery, and Engineering Process.
- **Section 4** discusses the governance concept. This covers roles, policies, processes, and standards for the management of the Allied Framework for MSaaS within NATO.
- **Section 5** provides an overview of the experimentation performed. This includes experimentation to explore and test enabling technology for architecture building blocks from the reference architecture, and experimentation to test solutions for certain types of simulation services.
- **Section 6** provides an overview of the evaluation activities performed.
- **Section 7** discusses the next steps and the incremental development and implementation strategy for the Allied Framework for MSaaS.
- And finally, **section 8** provides a summary and conclusions.

OPERATIONAL CONCEPT

MSaaS from the User Perspective

MSaaS enables users to discover new opportunities for training and working together and enables users to enhance their operational effectiveness, saving costs and efforts in the process. By pooling individual user's requirements and bundling individual requests in larger procurement efforts, the position of buying authorities against industrial providers is strengthened.

> *"MSaaS aims to provide the user with discoverable M&S services that are readily available on-demand and deliver a choice of applications in a flexible and adaptive manner."*

MSaaS aims to provide the user with discoverable M&S services that are readily available on-demand and deliver a choice of applications in a flexible and adaptive manner. It offers advantages over the existing stove-piped M&S paradigm in which the users are highly dependent on a limited amount of industry partners and subject matter experts.

The MSaaS concept is illustrated in Figure 1. MSaaS is an enterprise-level approach for discovery, composition, execution and management of M&S services. MSaaS provides the linking element between M&S services that are provided by a community of stakeholders to be shared and the users that are actually utilizing these capabilities for their individual and organizational needs.

The Allied Framework for MSaaS defines user-facing capabilities (front-end) and underlying technical infrastructure (back-end). The front-end is called the MSaaS Portal. The front-end provides access to a large variety of M&S capabilities from which the users are able to select the services that best suit their requirements, and track the experiences and lessons learned of other users. The users are able to discover, compose and execute M&S services through the front-end, which is the central access point that guides them through the process:

- **Discover:** The Allied Framework for MSaaS provides a mechanism for users to search and discover M&S services and assets (e.g., Data, Services, Models, Federations, and Scenarios). A registry is used to catalogue available content from NATO, National, Industry and Academic organizations. This registry provides useful information on available services and assets in a manner that the user is able to assess their suitability to meet a particular requirement (i.e., user rating, requirements, simulation specific information, and verification and validation information). The registry also points to a repository (or owner) where that simulation service or asset is stored and can be obtained, including business model information (i.e., license fees, pay per use costs).
- **Compose:** The Framework provides the ability to compose discovered services to perform a given simulation use case. Initially it is envisaged that simulation services will be composed through existing simulation architectures and protocols (e.g., using DIS, HLA, DDS) and can be readily executed on-demand (i.e., with no set up time). In the longer term, distributed simulation technology will evolve, enabling further automation of discovery, composition and execution than is possible today.
- **Execute:** The Framework provides the ability to deploy the composed services automatically on a cloud-based or local computing infrastructure. The automated deployment and execution allows to exploit the benefits of cloud computing (e.g., scalability, resilience).

Once deployed and executed the M&S services can be accessed on-demand by a range of users (Live, Virtual, Constructive) directly through a simulator (e.g., a flight simulator consuming a weapon effects service), through a C2 system (e.g., embedded route planning functionality that utilizes a route planning service) or may be provided by a thin client or by a dedicated application (e.g., a decision support system utilizing various services like terrain data service, intelligence information service etc.). The execution services support a range of business models and are able to provide data relevant to those models (i.e., capture usage data for a pay-per-use business model).

The Allied Framework for MSaaS is the linking element between service providers and users by providing a coherent and integrated capability with a Technical Reference Architecture, recommendations and specifications for discovery, composition and execution of services, and necessary processes and governance policies.

Operational Concept Document

The purpose of the Operational Concept Document (OCD) for the Allied Framework for MSaaS is to inform relevant stakeholders how the framework will function in practice. The capabilities and key characteristics of the proposed framework are included in the OCD as well as how stakeholders will interact with the system.

Specifically, the main goals of the OCD are to inform the operational stakeholders how to evolve from their current operational stove-piped systems to the Allied Framework for MSaaS. It also serves as a platform for stakeholders to collaboratively adapt their understanding of the systems operation as new developments, requirements or challenges arise. Therefore, the OCD is written in the common language of all interested parties.

Vision and Goals

The *MSaaS Vision Statement* is defined as:

M&S products, data and processes are conveniently accessible and available on-demand to all users in order to enhance operational effectiveness.

To achieve the MSaaS Vision Statement the following MSaaS goals are defined:

1. To provide a framework that enables credible and effective M&S services by providing a common, consistent, seamless and fit for purpose M&S capability that is reusable and scalable in a distributed environment.

2. To make M&S services available on-demand to a large number of users through scheduling and computing management. Users can dynamically provision computing resources, such as server time and network storage, as needed, without requiring human interaction. Quick deployment of the customer solution is possible since the desired services are already installed, configured and on-line.

3. To make M&S services available in an efficient and cost-effective way, convenient short set-up time and low maintenance costs for the community of users will be available and to increase efficiency by automating efforts.

4. To provide the required level of agility to enable convenient and rapid integration of capabilities, MSaaS offers the ability to evolve systems by rapid provisioning of resources, configuration management, deployment and migration of legacy systems. It is also tied to business dynamics of M&S that allow for the discovery and use of new services beyond the users' current configuration.

TECHNICAL CONCEPT

The technical concept comprises several volumes:

- Volume 1: MSaaS Technical Reference Architecture: discusses layers, architecture building blocks and architectural patterns [15].
- Volume 2: MSaaS Discovery Service and Metadata: discusses services metadata and metadata for services discovery [16].
- Volume 3: MSaaS Engineering Process: discusses a services oriented overlay for the DSEEP [17].

This section will focus primarily on the MSaaS Reference Architecture (RA) and briefly explain the other volumes.

MSaaS Reference Architecture

Principles

The MSaaS RA is defined with a number of principles in mind. These principles are similar to the Open Group SOA Reference Architecture (SOA RA) [3] key principles and are the starting point for the architecture work by MSG-136. The principles are:

The MSaaS RA:

1. Should be a generic solution that is vendor-neutral.
2. Should be modular, consisting of building blocks which may be separated and recombined.
3. Should be extendable, allowing the addition of more specific capabilities, building blocks, and other attributes.
4. Must be compliant with NATO policies and standards (such as AMSP-01 [4] and STANAG 4603 [5]).
5. Must facilitate integration with existing M&S systems.
6. Should be capable of being instantiated to produce:
   a. Intermediary architectures
   b. Solution architectures
7. Should address multiple stakeholder perspectives.

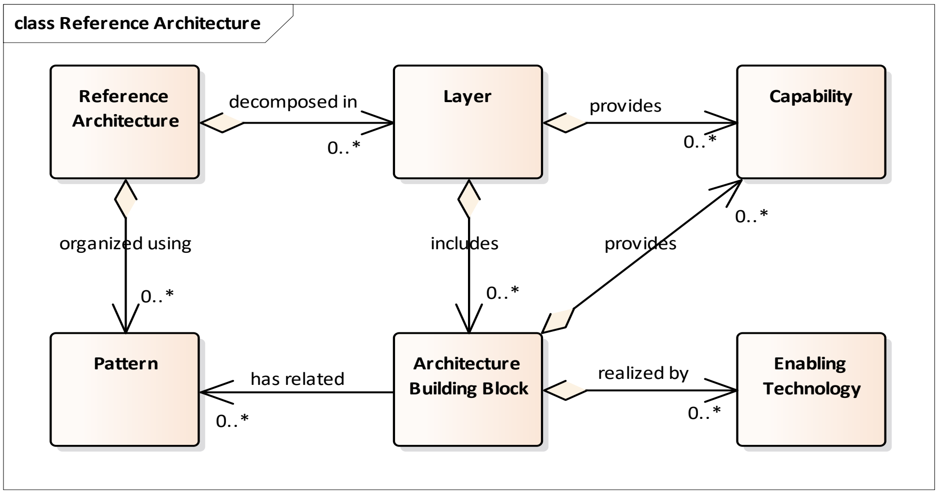


Figure 2: Reference architecture concepts. ***(Source Author)***

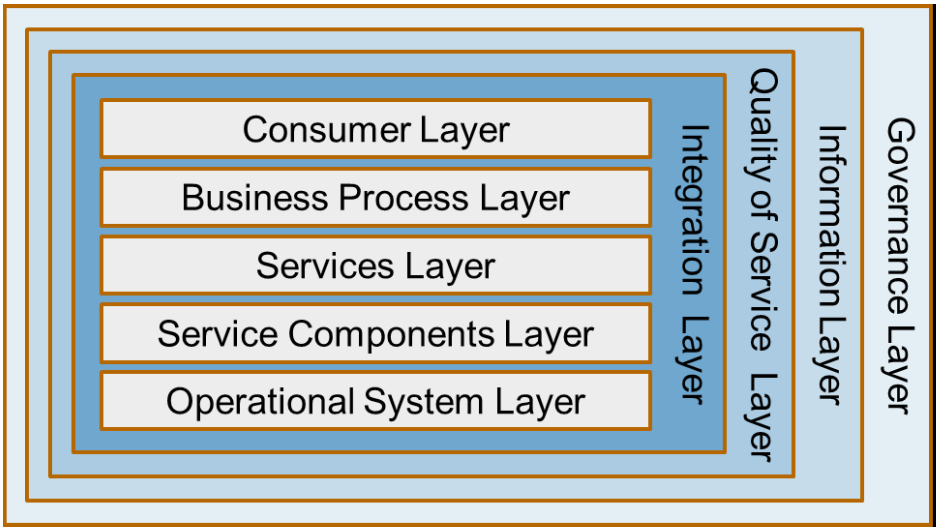


Figure 3: Reference architecture layers. ***(Source Author)***

Architecture Concepts

An architecture can generally be described at different levels of abstraction and the term *reference architecture* is typically used for a more abstract form of architecture. The purpose of the MSaaS RA is to provide a template for the development of an MSaaS *intermediate architecture* or of one or more specific MSaaS *solution architectures*. The MSaaS RA provides guidelines, options, and constraints for making design decisions with regards to an MSaaS solution architecture and solution implementation.

The MSaaS RA uses several concepts for describing the architecture. These concepts and their relationships are illustrated in Figure 2.

The MSaaS RA defines a number of capabilities in the form of architecture building blocks and organizes these capabilities in so-called layers. An architecture building block captures, amongst others, requirements, applicable standards, relationships with other building blocks, related architectural patterns, and references to (examples of) enabling technology. The particular connection between architecture building blocks that recur consistently in order to solve certain classes of problems is called a pattern. A pattern describes how architecture building blocks can be put together for creating proven solution architectures. The enabling technology provides means for the technical realization of an architecture building block.

The MSaaS RA layers are modelled after the SOA RA layers [3], while the content of each layer in terms of architecture building blocks is supplied by the NATO C3 Taxonomy [6].

Layers and Architecture Building Blocks

The MSaaS RA is decomposed in layers, similar to the SOA RA layering structure, and each layer includes a set of architecture building blocks that provide some capability. The 9 layers are illustrated in Figure 3. Some of the layers are cross-cutting layers. For example the architecture building blocks in the Quality of Service Layer affect the building blocks in the Operational System Layer up to the Integration Layer.

Note that the SOA RA layers are presented from technical infrastructure layers to consumer-facing layers in that order. Also, some naming may cause confusion between C3 Taxonomy users and the SOA RA users. For example, the Operational Systems Layer does not refer to the defence operations that the C3 Taxonomy's Operational Capabilities layer does, but rather to the operational run-time capabilities in a SOA.

The architecture building blocks per layer are shown in Table 1.

The architecture building blocks are aligned with the NATO C3 Taxonomy and necessary changes will be recommended.

As an example, the *Business Process Layer* provides the capabilities to compose and execute a simulation, and contains the following architecture building blocks:

- **M&S Composition Services:** compose a simulation environment from individual services that together meet the objectives of the simulation environment.
- **M&S Simulation Control Services:** provide input to, control, and collect output from a simulation execution.
- **M&S Scenario Services:** manage the simulation of scenarios.

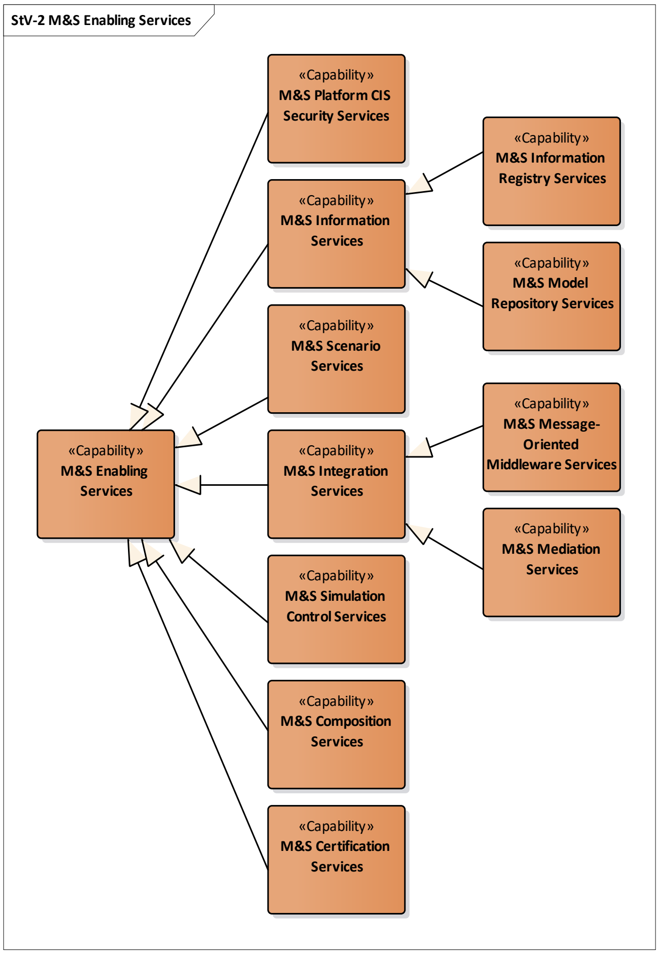

StV-2 M&S Enabling Services
«Capability» M&S Platform CIS Security Services
«Capability» M&S Information Registry Services
«Capability» M&S Information Services
«Capability» M&S Model Repository Services
«Capability» M&S Scenario Services
«Capability» M&S Message-Oriented Middleware Services
«Capability» M&S Enabling Services
«Capability» M&S Integration Services
«Capability» M&S Mediation Services
«Capability» M&S Simulation Control Services
«Capability» M&S Composition Services
«Capability» M&S Certification Services

Figure 4: Taxonomy of architecture building blocks. *(Source Author)*

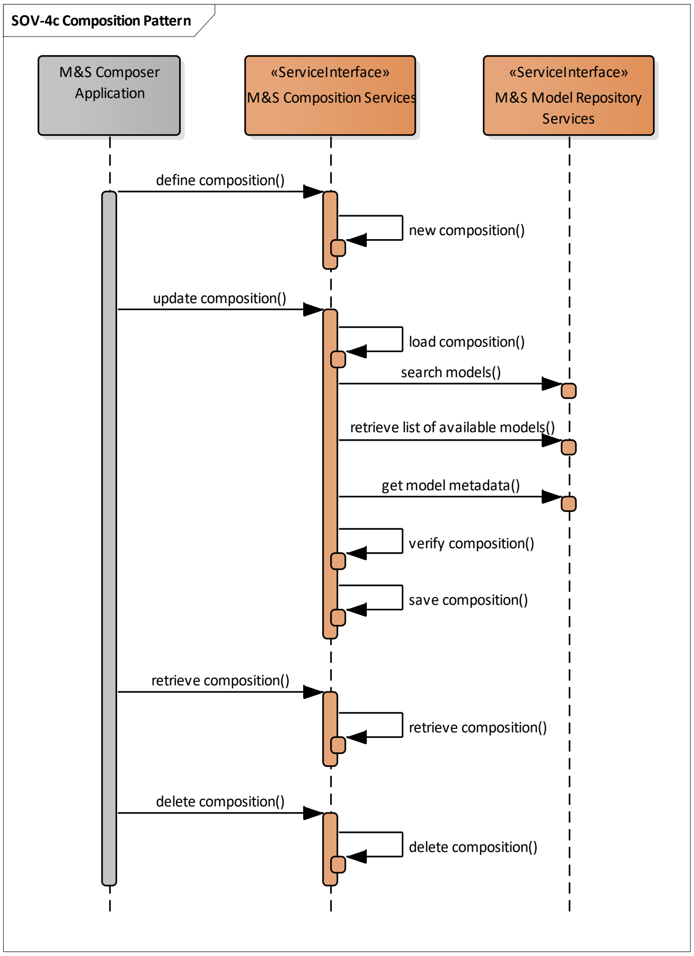


Figure 5: Example of a pattern. *(Source Author)*

Each of these architecture building blocks has associated requirements and other attributes. As an example, some requirements for the M&S Composition Services are listed in Table 2.

The architecture building blocks of the MSaaS RA are organized in a taxonomy, in line with the NATO C3 Taxonomy (see Figure 4). Most of the architecture building blocks in Table 1 fall under the M&S Enabling Services, providing capabilities to create a simulation environment in which M&S Specific Services are brought together to fulfil the purpose of that simulation environment. M&S Specific Services are mostly Simulation Services and Composed Simulation Services, such as Synthetic Environment Services, Route Planning Services, or Report Generation Services.

Architectural Patterns

The architectural patterns show how architecture building blocks in the MSaaS RA are related, can be combined, how they interact, and what information is generally exchanged. The architectural patterns serve as reference for solution architectures and design patterns for solution architectures. An initial set of architectural patterns is documented, but the idea is that the architecture building blocks as well as the architectural patterns are governed as a "living document" and will evolve further as knowledge is gained and as technology evolves.

Figure 5 illustrates one example of an architectural pattern, in relation to the M&S Composition Services mentioned earlier.

In this example, a user composes a simulation environment using an M&S Composer Application. This application, in turn, employs the capabilities of M&S Composition Services and the M&S Model Repository Services. This pattern provides support for the definition, update, retrieval, and deletion of compositions. The M&S Composer Application is user-facing while the other architecture building blocks operate "behind the scene". The interactions in the figure also imply requirements on each architecture building block.

MSaaS Discovery Service and Metadata

Technical volume 2 [16] discusses information and standards related to the description of services and exchange of metadata. More specifically:

- provides an overview of standards related to services discovery and services interface description, and
- presents national initiatives related to the exchange of services metadata, and to information models that support the (automated) composition, deployment and execution of simulation environments.

Table 1: Layers and architecture building blocks. ***(Source Author)***

Layer	Architecture Building Blocks
Operational Systems Layer	» Infrastructure Services » Communication Services
Service Components Layer	» SOA Platform Services
Services Layer	» M&S Specific Services
Business Process Layer	» M&S Composition Services » M&S Simulation Control Services » M&S Scenario Services
Consumer Layer	» M&S User Applications » NATO User Applications
Integration Layer	» M&S Message-Oriented Middleware Services » M&S Mediation Services
Quality of Service Layer	» SOA Platform SMC Services » M&S Security Services » M&S Certification Services
Information Layer	» M&S Information Registry Services
Governance Layer	» M&S Repository Services » Metadata Repository Services

Table 2: M&S Composition Services requirements. ***(Source Author)***

Function	Requirements
Manage Lifecycle	1. The M&S Composition Services shall provide the means to define a parameterized simulation composition. 2. The M&S Composition Services shall provide the means to update, delete and retrieve a defined simulation composition.
Execute composition	3. The M&S Composition Services shall provide the means to start the execution of a simulation composition, and to provide composition parameter values. 4. The M&S Composition Services shall provide the means to orchestrate, restart and stop the execution of a simulation composition.
Programmatic Interfaces	5. The M&S Composition Services shall provide APIs to the Manage Lifecycle and Execute Composition functionality.

Table 3: Service provider lifecycle stages. ***(Source Author)***

Lifecycle Stage	Description
Proposed	The proposed service's needs are identified and assessed as to whether needs can be met through the use of services.
Definition	The service's requirements are gathered and the design is produced based on these requirements.
Development	The service specifications are developed and the service is built.
Verification	The service is inspected and/or tested to confirm it is of sufficient quality, complies with the prescribed set of standards and regulations, and is approved for use.
Production	The service is available for use by its intended consumers.
Deprecated	The service can no longer be used by new consumers.
Retired	The service is removed from the Allied Framework and is no longer used.

This volume relates to several architecture building blocks in the MSaaS RA, such as the M&S Composition Services for automated composition, deployment and execution; and the M&S Model Repository Services for metadata standards.

MSaaS Engineering Process

Technical volume 3 [17] discusses a service-oriented overlay for the Distributed Simulation Engineering and Execution Process (DSEEP) [7], by adding an overlay for a service-oriented implementation strategy (besides HLA, DIS, and TENA). This volume discusses the activities or tasks related to this implementation strategy.

GOVERNANCE CONCEPT

Governance and Roles

A challenging aspect of establishing a persistent capability like the Allied Framework for MSaaS is to develop an effective governance model. Governance ensures that all of the independent service-based efforts (i.e. design, development, deployment, or operation of a service) combined will meet customer requirements.

MSG-136 developed policies, processes, and standards for managing the lifecycle of services, service acquisitions, service components and registries, service providers, and consumers. These are defined in the *Allied Framework for Modelling and Simulation as a Service (MSaaS) Governance Policies* [1 3], and are intended to be published as Allied Modelling and Simulation Publication AMSP-02.

The NMSG is the delegated NATO authority for M&S standards and procedures. Nations are encouraged to use the standards nationally or in other multi-national collaborations. After completion of the MSG-136 task group, the NMSG M&S Military Operational Requirements Subgroup (MORS) will become custodian of the governance policies. MORS is the custodian of best practices with regards to the use of M&S in the training domain and in other domains. The governance policies will be submitted to MORS for future maintenance, updates and dissemination with respect to operational needs of NATO agencies and national stakeholders.

The NMSG M&S Standards Subgroup (MS3) will become custodian of the MSaaS Technical Reference Architecture [15], and is responsible for the maintenance of the MSaaS technical aspects and standards documents.

General Policies

The general policies for instituting governance mechanisms of MSaaS-based solutions are:

- An MSaaS implementation shall conform to the governance policies as identified and established by the governance document.
- An MSaaS solution architecture shall comply with the MSaaS Technical Reference Architecture (see section 3, Technical Concept).
- Any M&S service shall conform to the practices and recommendations

for Integration, Verification and Compliance Testing as defined by NATO MSG-134 [8].

The ability to effectively manage all stages of the service lifecycle is fundamental to the success of governing M&S services. The Service Lifecycle Management Process as defined in [9] contains a set of controlled and well-defined activities performed at each stage for all versions of a given service. Table 3 lists the sequential service provider lifecycle stages.

All service providers shall define levels for each service (e.g., regarding availability, etc.). Service Providers and users shall agree on a Service Level Agreement (SLA) prior to usage. Obviously, service providers are required to indicate the forecasted retirement date of a specific version of a service.

Security Policies

The approach to ensuring security is intrinsically related to the cloud computing service model (SaaS, PaaS, or IaaS) and to the deployment model (Public, Private, Hybrid, or Community) that best fits the Consumer's missions and security requirements. The Consumer has to evaluate the particular security requirements in the specific architectural context, and map them to proper security controls and practices in technical, operational, and management classes. Even though the Cloud Security Reference Architecture [19] inherits a rich body of knowledge of general network security and information security, both in theory and in practice, it also addresses the cloud-specific security requirements triggered by characteristics unique to the cloud, such as decreased visibility and control by consumers. Cloud security frameworks including information management within an infrastructure shall support the cloud implementers, providers and consumers [10]. However, MSG-136 recognizes that a more tailored approach may be needed to exploit MSaaS specific capabilities and proposes to develop additional guidelines as part of follow-on work.

Compliancy Policies

Compliancy testing of individual components of a NATO or multi-national simulation environment is the ultimate responsibility of the participating organizations. Currently, NMSG and its support office (MSCO) do not provide compliancy testing services or facilities. Some existing HLA certification tools and services cover only basic testing (i.e., HLA Rules, Interface Specification and Object Model Template (OMT) compliance) and do not provide in-depth functional testing that is needed to support federation integration and validation. The available tools are also outdated. The current NMSG activity MSG-134 is addressing the next generation of compliancy testing and certification needs for HLA [8].

EXPERIMENTATION

MSG-136 performed several experiments to test enabling technology for MSaaS. Two strands of experimentation were performed: (1) experimentation to explore and test enabling technology for architecture building blocks from the reference architecture, and (2) experimentation to test solutions for certain types of Simulation Services. Test cases were defined, tests performed, and test results recorded in an experimentation report [18]. A brief overview of the experimentation and test cases follows below.

Explore and Test Enabling Technology

Most test cases in this strand of experimentation evolve around container technology as the enabling technology for a number of architecture building blocks. This technology enables M&S Enabling Services and M&S Specific Services to run on a local host as well as in a cloud environment.

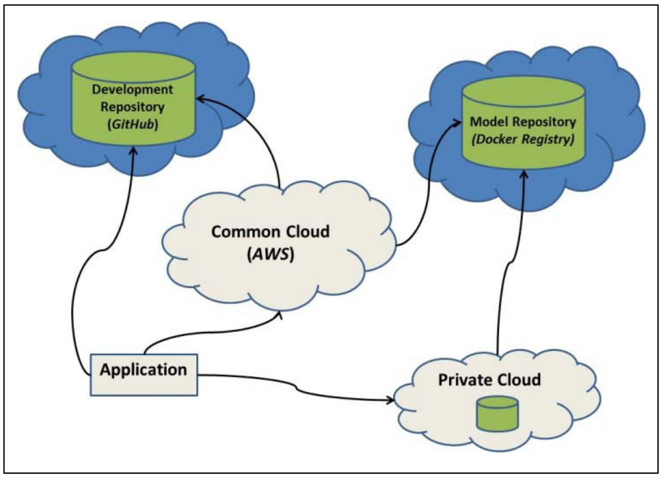


Figure 6: Illustration of experiment environment. ***(Source Author)***

The experiment environment that was used for the test cases is illustrated in the following figure. The experiment environment is a collection of private clouds and a common cloud. The common cloud is Amazon Web Service (AWS), sponsored by NATO CSO.

Common components are:

- A private Docker Registry and a web-based front-end for the exchange of Docker container images (provided by NLD);
- A private GitHub repository for the description of container images in the Docker Registry, and for the exchange of software, configuration files and other developmental data (provided by USA).

The Docker Registry contains several container images for containerized HLA federates, from which various compositions can be created for the different test cases. Many of these images have been created following the design patterns in [11].

Test cases include:

- Container networking: explore different container networking models for connecting containerized HLA federate applications.
- Containerization of HLA federates: evaluate approaches in containerizing HLA federate applications (see also [11]).

- Metadata Repositories and Discovery: Demonstrate interoperation of repositories across nations.
- Simulation Composition: explore automated composition and execution of services.
- Container Orchestration Environments: evaluate two popular container orchestration environments for M&S (see also [12]).

Test Solutions for Simulation Services

Tests cases in this strand of experimentation concern the interoperation of applications with certain types of Simulation Services. Test cases include:

- Computer Generated Forces (CGF) – Synthetic Environment Service: connect a CGF simulator to a Synthetic Environment Service to request environment data in various formats.
- C2 Application – Route Planning Service: connect a C2 Application to a Route Planning Service to request route planning information.

EVALUATION

The evaluation activities focus on whether MSaaS will reduce costs and integration time for creating a new instance of a simulation environment, compared to what it costs today. What is the main advantage of having an MSaaS-based solution? The premise of the evaluation activities is to answer this objectively based on the measurements performed and data collected. The evaluation activities of MSG-136 are currently ongoing and will be included in the MSG-136 Final Report.

IMPLEMENTATION STRATEGY AND NEXT STEPS

Implementation Strategy

Service-based approaches rely on a high degree of standardization and automation in order to achieve their goals. Therefore the development and implementation of a recommended set of supporting standards is a key output of the reference architecture. MSG-136 research has identified the importance for the following capabilities:

- **M&S Composition Services:** create and execute a simulation composition. A composition can be created from individual simulation services or from smaller compositions.
- **M&S Repository Services:** store, retrieve and manage simulation service components and associated metadata that implement and provide simulation services, in particular metadata for automated composition.
- **M&S Security Services:** implement and enforce security policies for M&S services.

MSG-136 proposes an incremental development and implementation strategy for the Allied Framework for M&S as a Service. The incremental approach facilitates a smooth transition in the adoption of an Allied Framework for M&S as a Service and describes a route that will incrementally build an Allied Framework for M&S as a Service.

The proposed strategy also provides a method to control the rate of expansion of the new framework permitting the iterative development and training of processes and procedures. Finally, it permits those nations that have been early adopters of an Allied Framework for M&S as a Service and have national capabilities to accrue additional benefits from their investments and highlight the benefits as well as providing lessons learned and advice to those nations considering similar investments.

As illustrated in Figure 7, the implementation strategy is broken down into three phases:

1. Phase 1: "Initial Concept Development” The Initial Concept Development (2015 until end of 2017) is executed by NMSG-136 and consists of concept development and initial experimentation. For this period an MSaaS Portal and individual M&S services were provided by individual members of MSG-136 for trial use.
2. Phase 2: “Specification & Validation” From 2018-2021 MSG-164 will mature MSaaS in an operationally

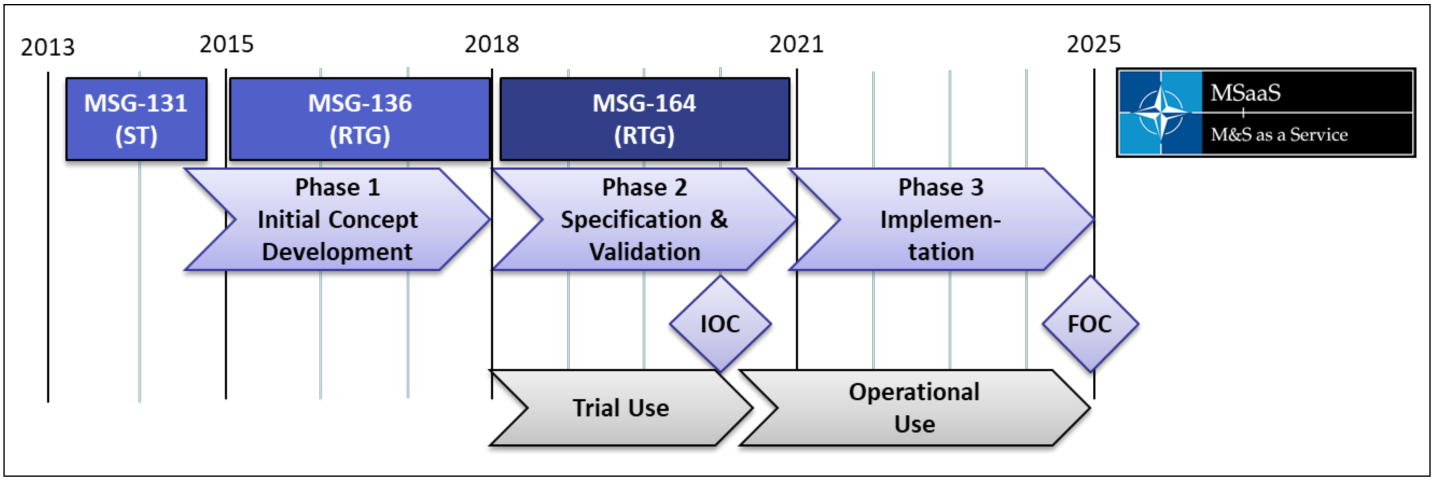


Figure 7: MSaaS implementation strategy. *(Source Author)*

relevant environment and conduct necessary research and development efforts to evolve and extend the initial concepts as developed by MSG-136. This phase includes development of suitable STANAGs or STANRECs, and moving from prototype implementation to operationally usable and mature systems.

3. Phase 3: "Implementation" By 2025 Full Operational Capability (FOC) is achieved which includes adaptation of many existing simulation related services to the MSaaS Reference Architecture. This is achieved primarily by adding services to the Allied Framework for M&S as a Service.

> *"MSaaS is a promising innovation towards more accessible and more cost effective M&S capabilities"*

Next Steps

The next steps in defining and evolving the Allied Framework for MSaaS are executed by MSG-164 (see previous section). MSG-164 kicked off in February 2018 and will finish in 2021. Building upon the Allied Framework for M&S as a Service developed by MSG-136 this activity addresses three main objectives:

1. To advance and to promote the operational readiness of M&S as a Service.
2. To align national efforts and to share national experiences in establishing MSaaS capabilities.
3. To investigate critical research and development topics to further enhance MSaaS benefits.

MSG-164 will specify and test an MSaaS infrastructure that is suitable for use in an operationally relevant environment and will support continued MSaaS experimentation and evaluation efforts. This activity will also deliver a Technical Report and recommendations with regards to the organizational perspective of introducing MSaaS in NATO and in the Nations.

To address the objectives, MSG-164 will cover the following topics:

1. Demonstrate MSaaS application in an operationally relevant environment through operational experimentation as part of exercises and integration into simulation applications (like simulation-based capability development). Annual participation in CWIX to develop MSaaS to maturity through a phased approach.
2. Maintain and enlarge the MSaaS Community of Interest.
3. Establish interim governance structure and collect experiences w.r.t. MSaaS governance.
4. Collect and share experiences in establishing MSaaS capabilities and providing M&S services.
5. Conduct research on M&S-specific service discovery and service composition.
6. Conduct research and development activities on M&S-specific federated cloud environments, federated identity management and cyber secure communications.
7. Conduct research on enabling services like scenario specification services, etc.

Additionally, MSG-164 will

1. Act as governance body for the Allied Framework for M&S as a Service, maintaining and updating (if needed) the therein included documents, i.e. AMSP-02 (MSaaS Governance Policies), the MSaaS Operational Concept Description, and the MSaaS Technical Reference Architecture) with associated technical documents.
2. Collaborate with international standards bodies (like SISO, IEEE, etc.).
3. Inform and engage stakeholders in NATO, Academia, and Industry about MSaaS.

SUMMARY AND CONCLUSIONS

SG-136 investigated the concept of M&S as a Service (MSaaS), a new concept for the discovery, composition, execution, and management of M&S services. The concept is described from different perspectives:

- **Operational concept** of MSaaS: how it works from the user point of view;
- **Technical concept** of MSaaS: technical reference architecture, services discovery metadata, and engineering process;
- **Governance concept** and roadmap for MSaaS within NATO.

Technical implementations of MSaaS have been developed and evaluated in several experiments and demonstrations (TRL 4). MSG-136 also proposed an MSaaS governance approach. The conclusion is that MSaaS is a promising innovation towards more accessible and more cost effective M&S capabilities.

The participating nations and NATO organizations are currently implementing MSaaS using cloud technology, based on the MSG-136 research and experimentation and to inform the user community. MSG-136 plans to further investigate a number of areas including discovery and composability of M&S services; and to address security aspects of cloud based solutions in more detail. A new technical activity is being prepared and will be submitted to NMSG for approval in the fall of this year.

The NMSG will continue to participate in the SISO Cloud-based M&S Study Group and share its approach and experiences. The goal is that our work will contribute to a set of open standards and recommendations for MSaaS.

ACKNOWLEDGEMENT

The authors would like to acknowledge the contributions of the MSG-136 team members for the ideas and results that are included in this paper.

REFERENCES

[1] NATO Modelling and Simulation Master Plan NMSMP v2.0 (AC/323/NMSG(2012)-015). https://www.sto.nato.int/NATODocs/NATO Documents/Public/NATO_MS_Master_Plan_Web.pdf

[2] https://www.sto.nato.int/Pages/activitieslisting.aspx?FilterField1=ACTIVITY_NUMBER&FilterValue1=MSG-136

[3] SOA Reference Architecture, C119, Open Group Standard, 2011

[4] NATO AMSP-01 (NATO Modeling and Simulation Standards Profile), Edition C version 1, NATO Standardization Office, March 2015

[5] STANAG 4603 Edition 2, Modeling and Simulation Architecture Standards for Technical Interoperability: HLA, NATO Standardization Office, 17 February 2015

[6] C3 Classification Taxonomy, Baseline 1.0, NATO Allied Command Transformation (ACT) C4ISR Technology and Human Factors (THF) Branch, 15 June 2012

[7] IEEE 1730-2010: Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)

[8] NATO Distributed Simulation Architecture & Design, Compliance Testing and Certification. STO Technical Report STO-TR-MSG-134. To be published.

[9] Federal Aviation Administration (FAA): "System Wide Information Management (SWIM) Governance Policies", Version 2.0, 12 March 2014.

[10] NATO Consultation, Command and Control Board (C3B): "NATO Cloud Computing Policy", AC/322-D(2016)0001, 7 January 2016.

[11] Guidelines and best practices for using Docker in support of HLA federations; 2016-SIW-031; SISO SIW Fall 2016; T.W. van den Berg, A. Cramp, B. Siegel.

[12] Container orchestration environments for M&S; 2017-SIW-006; SISO SIW Fall 2017; T.W. van den Berg, A. Cramp.

[13] NATO STO: Allied Framework for Modelling and Simulation as a Service (MSaaS) - Governance Policies. 16 January 2018.

[14] NATO STO: Operational Concept Document (OCD) for the Allied Framework for M&S as a Service. AC/323(MSG-136) TP/830. STO-TR-MSG-136-Part-III. 16 January 2018.

[15] NATO STO: Modelling and Simulation as a Service, Volume 1: MSaaS Technical Reference Architecture. AC/323(MSG-136)TP/831. STO-TR-MSG-136-Part-IV. 16 January 2018.

[16] NATO STO: Modelling and Simulation as a Service, Volume 2: MSaaS Discovery Service and Metadata. AC/323(MSG-136)TP/832. STO-TR-MSG-136-Part-V. 16 January 2018.

[17] NATO STO: Modelling and Simulation as a Service, Volume 3: MSaaS Engineering Process. AC/323(MSG-136)TP/833. STO-TR-MSG-136-Part-VI. 16 January 2018.

[18] NATO STO: MSaaS Concept and Reference Architecture Evaluation Report. AC/323(MSG-136)TP/829. STO-TR-MSG-136-Part-II. 16 January 2018.

[19] National Institute of Standards and Technology: NIST Cloud Computing Security Reference Architecture, NIST Special Publication 500-299, Draft, 15 May 2013.

ABOUT THE AUTHORS

ROBERT SIEGFRIED is Senior Consultant for IT/M&S projects and Managing Director of Aditerna GmbH, a company providing specialized services and consulting in this area. He earned his doctorate in modeling and simulation at the Universität der Bundeswehr München. Within several projects for the German Armed Forces, he has worked on documentation guidelines, model management systems, distributed simulation test beds and process models. He is active member of NATO MSG-128 ("Mission Training through Distributed Simulation") and is co-chair of NATO MSG-136 ("Modelling and Simulation as a Service (MSaaS) – Rapid deployment of interoperable and credible simulation environments"). He is actively involved in multiple SISO working groups and is member of the SISO Executive Committee.

JON LLOYD is a Principal Engineer working in Modelling and Simulation in the UK Ministry of Defence (MOD) Defence Science and Technology Laboratory (Dstl) for the past 16 years. During this time Jon has previously been a technical specialist in the development of CBRN Synthetic Environments and interoperability standards. Jon currently leads the Live, Virtual and Constructive Simulation research project in Dstl, overseeing research undertaken by Dstl, Industry and Academia on the development of simulation architectures, interoperability and management; representation of the future operating environment; and evaluation of emerging commercial simulation technology related to the use of simulation for Defence purposes. Jon has chaired and participated in various International Research Collaboration activities as part of his role and is the current UK national lead for NATO MSG-136.

TOM VAN DEN BERG is a senior scientist in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He holds an M.Sc. degree in Mathematics and Computing Science from Delft Technical University and has over 25 years of experience in distributed operating systems, database systems, and simulation systems. His research area includes simulation systems engineering, distributed simulation architectures, systems of systems, and concept development & experimentation. Tom is a member of several SISO Product Development / Support Groups, participates in several NATO MSG activities, and is co-chair of NATO MSG-136 ("Modelling and Simulation as a Service (MSaaS) – Rapid deployment of interoperable and credible simulation environments").

SOFTWARE DEFINED NETWORKING FOR ARMY'S TACTICAL NETWORK:

Promises, Challenges, and an Architectural Approach

By: Dr. Bharat Doshi, Derya Cansever, US Army CERDEC

Software Defined Networking (SDN) and Network Function Virtualization (NFV) together promise to revolutionize the networking with unprecedented improvements in the speed of introduction of new services, programmability and reconfigurability, resource efficiency, and security posture. While early successes of these concepts have been in database applications, emerging beneficiaries are the enterprise and carrier networks, both wired and wireless. Our qualitative analysis of the SDN/NFV in the context of Army's Tactical Networks (ATNs) uncovers significant potential but also identifies serious deficiencies in the standard SDN/NFV for this application. We identify a family of architectures, involving hierarchically distributed SDN Controllers, which have the potential of providing the promised benefits while mitigating the disadvantages. We then discuss the required research to add detail to this conceptual framework and to relate the architectural specifics and design parameters to the characteristics of the network. We briefly discuss a prototype novel SDN architecture, consistent with the Framework, for one SDN Controller controlling one network domain.

INTRODUCTION AND SUMMARY

In this article, we discuss potential benefits and challenges for SDN/NFV in the context of Army's tactical networks. We discuss a family of architectures for SDN Controllers that could meet these challenges, while providing the promised gains in programmability, efficiency, and security. We also discuss the research needed to select specifics of this architecture as functions of the characteristics of the network.

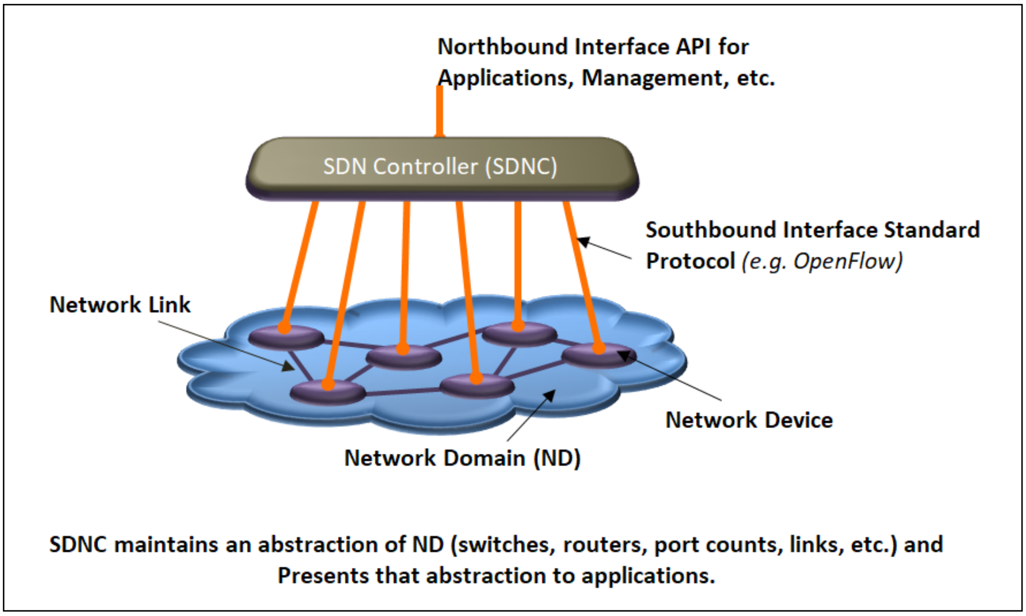

Northbound Interface API for Applications, Management, etc.
SDN Controller (SDNC)
Southbound Interface Standard Protocol (e.g. OpenFlow)
Network Link
Network Device
Network Domain (ND)
SDNC maintains an abstraction of ND (switches, routers, port counts, links, etc.) and Presents that abstraction to applications.

Figure 1: Basic SDN Architecture. ***(Source Author)***

The separation of control and data (forwarding) planes and the centralization of control functions in a controller are defining characteristics of SDN. The key characteristic of NFV is running multiple networking functions in one general purpose hardware platform as Virtual Machines (VMs) or Containers. In this paper, we will use 'SDN' to denote SDN and NFV together.

The separation of control and data planes allows a representation of all of the lower level networking infrastructure as an abstraction to the higher level control and management functions residing in the SDN controller, thus allowing introduction of new services without knowing the specific technologies of the underlying infrastructure. The result is a very high degree of programmability, which allows rapid service development and deployment, easier reconfigurability, and interoperability among diverse networks. In fact, many consider this programmability (and not the separation of forwarding and control planes) the defining characteristic of SDN [1]. With NFV, VMs and resources available to each VM are defined dynamically. In addition, a VM or one of its functions can be moved rapidly from one HW platform to another. Collectively, they permit rapid reconfigurability and adaptability, improved resource efficiency, scalability, redundancy, and security.

Control decisions are made by the SDN Controller based on the status of the network infrastructure and then communicated to the infrastructure nodes for actions. Thus, effective functioning of the standard SDN requires significant amount of information transfer between the SDN Controller and the network infrastructure and this information transfer needs to happen securely with minimal incremental latency. High availability, low latency, and secure communication are thus very critical.

Army Tactical Networks (ATNs) exhibit some unique characteristics. There is a strong hierarchy, mimicking the hierarchy in the force structure There are multi-hop peer-to-peer wireless communication networks (MANETs) within some layers of hierarchy. The data rates in the lower levels of hierarchy are typically low and connectivity is unpredictable. Many devices have limited energy supply. Devices may not have physical protection and may operate in enemy territory. The device could be captured by the adversary, thus providing authenticated access to malicious actors. Unit Task Reorganization (UTRs) are frequent and there is a need for supporting multiple classification levels. Virtual Private Networks (VPNs) are routinely used but are manually configured. A natural question is the value of SDN in such an environment.

In addition to the general benefits mentioned above, SDN can help automate VPN establishments, UTR execution, and distributions of security certificates and patches. It can also improve Network and Cyber Situation Awareness and deployment of policies. On the other hand, unpredictable connectivity and low data rates are perhaps the biggest challenges for the deployment of standard SDN in ATNs since the standard SDN relies on very frequent communication and information transfer between the SDN Controller and network devices. Other big challenges stem from a highly centralized architecture making the SDN Controller a single point of failure as well as the target of cyber-attacks.

As discussed below our proposed family of architectures addresses the above challenges and minimizes the reliability and security risks associated with a highly centralized SDN architecture.

We discuss the work that needs to be done to add specifics to this architecture, relate design parameters to network parameters, prototype key components, and provide early validation.

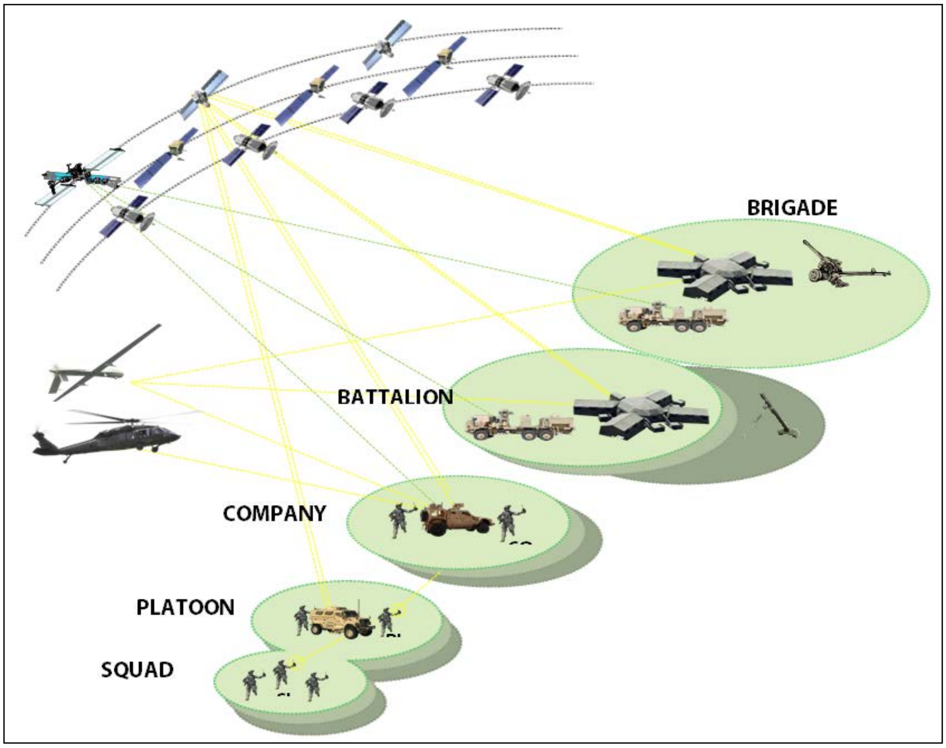


Figure 2: Army Tactical Hierarchy and Corresponding Networks. *(Source Author)*

SOFTWARE DEFINED NETWORKING (SDN) AND NETWORK FUNCTIONS VIRTUALIZATION (NFV)

Figure 1 shows the basic architecture of SDN.

SDN enables abstraction of the network infrastructure and data plane in the SDN Controller (SDNC) so a multi-technology, multi-vendor infrastructure is presented to higher layer applications and management functions as a unified network in a standard format. NFV allows creation of multiple virtual entities (VEs) on a single general purpose hardware platform. VEs can be implemented using Virtual Machines (VM) technology [2] or the lighter weight Containers (CONT) technology [3].

As mentioned earlier, we will use 'SDN' to refer to SDN and NFV together.

The above properties of SDN enable the network to be 'softwarized' and 'virtualized' at the SDNC. The following are key benefits:

- Rapid development and deployment of applications
- Separation of networking HW and SW businesses
- High degree of network programmability and reconfigurability
- Potential to enhance the cyber security posture
- Scalability of control and management
- Efficient use of network resources
- Broader situation awareness and faster responses to events
- Easier deployment of cloud computing and other centralization initiatives
- More efficient use of IT experts, very precious resources in tactical environment

Overall, SDN could do for the network infrastructure what the host virtualization has done for the computing infrastructure, namely make the complex network topologies and architectures vendor-independent, easier to monitor and manage, and efficient to operate. While the Southbound interface (between SDNC and network infrastructure) has been standardized (OpenFlow and other solutions), work is ongoing to define the Northbound interface (between SDNC and applications) standards. Applications development and deployment, network management, and configurations could then be much simpler, faster, and cheaper. The combined North and Southbound interfaces will also facilitate development of a common situation awareness, specification and deployment of network management policies, implementation of distributed firewalls, and defense in depth.

Of course, SDN raises concerns about the centralization of controls, resulting in a single point of failure, an attractive target for cyber-attacks, and a traffic bottleneck. In particular, the effectiveness of the standard SDN architecture relies on secure, reliable, and high speed connectivity between the network infrastructure devices and the SDNC.

When multiple network domains are involved, with possibly multiple ownerships, we may need multiple SDN Controllers. IETF is working on Interfacing SDN Domain Controller protocol (SDNi). Multiple SDN Controllers communicating via SDNi will also provide some scalability, incremental deployment of SDN, and support of diverse policies.

CHARACTERISTICS OF ATNS

Figure 2 shows a schematic of ATN architecture. ATNs reflect the hierarchy in Army's force structure, starting from individual soldier to squad, platoon, company, battalion, brigade, division, and corps. Higher levels of ATNs provide connectivity to the enterprise network. ATN is divided into Lower Tactical Internet (LTI), Mid-Tier Tactical Internet (MTTI), and Upper Tactical Internet (UTI). Within each level of hierarchy at lower levels, Line Of Sight (LOS) connectivity is typically provided by distributed wireless mesh (MANETs) with thin connectivity outside the mesh. Multiple

MANETs exist at a given level. Inter-MANET connectivity Beyond Line Of Sight (BLOS) could be provided by SATCOM or other communications relays. At higher levels of hierarchy, there is a mix of terrestrial wireless, SATCOM, and wireline connectivity. In addition to the networks designed for people-people communications, situation awareness, and C2, there are special purpose networks for FIRE support, Logistics, medical, etc.

Virtual Private Networks (VPNs) are used extensively in ATN. Manual setups are time and resource consuming. As the missions and tasks within a mission change, people may get reassigned to different units. These Unit Task Reorganizations (UTRs) are quite frequent in Army tactical environment. They require significant time-consuming reconfigurations. ATNs also support communications at multiple classification levels.

Data rates available within a level of hierarchy are typically low, especially in LTI and MTTI, where they range between 100 kbps and 2 - 5 Mbps. Tactical SATCOM providing BLOS connectivity also have low data rates (100 kbps – 1 Mbps). Also, individual radios are mobile and the entire MANET could be on-the-move. This mobility results in changing RF environment, unpredictable connectivity, and unpredictable data rates. Adversarial jamming adds to the unpredictability of RF links. Given that the user devices may also be network switches and routers, the network topology is governed by the user mission and not by the placement required for optimal RF links, thus increasing the unpredictability. A unit may get physically separated from the rest and may have very limited connectivity with the peer units or with the higher level.

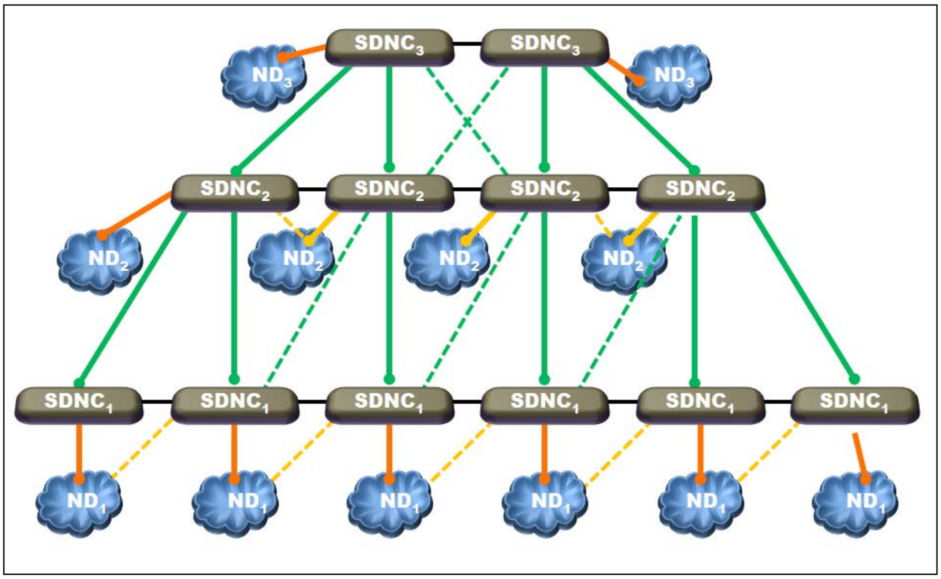


Figure 3: Hierarchically Distributed Family of SDN Controllers. ***(Source Author)***

ATNs support communications at different classification levels. Some parts of the network infrastructure may be shared among different classification levels while the others are segregated. Some segregation happens at physical layer, the other happens logically using Virtual Private Networks (VPNs), Multi-Protocol Label Switching (MPLS) and other logical isolation mechanisms.

POSSIBLE SDN ARCHITECTURES FOR ATN

Given the mission criticality of ATNs, a single SDNC controlling all networking functions in an entire tactical network (say, for an entire Brigade) would not be a desirable solution in any case. It would be a single point of failure for the entire network and a very attractive target for kinetic and/or cyber-attacks. Unreliable and low data rate communication between network devices and SDNC make it even more difficult to give the responsibility of very time critical control functions for a large number of users to a Centralized SDNC. Multiple classification levels add to the difficulty.

On the other hand, SDN concept may offer key advantages to ATNs, in addition to the benefits mentioned above for all SDNs:

- Enabling automation of several very important functions involving tedious manual processes today:
  - Setting up networking in a new location
  - UTR
  - VPN set up and maintenance
  - Management of PKI certificates and patch deployment
- Efficient use of network resources and spectrum, detection of cyber-attacks, and appropriate responses to such attacks via better situation awareness (SA)
- Enabling many breakthrough capabilities such as dynamic spectrum allocation, cognitive networking, and moving target cyber defense.
- Facilitating policy deployment (for network management, quality of service, priorities, cyber defense, etc.).
- Minimizing user interactions with the control plane could reduce the ability of an adversary to inflict major damage using a cyber-attack.

These potential benefits motivate us to investigate SDN architectures that can take advantage of the key concepts but mitigate the concerns expressed above.

Figure 3 is a simplified depiction of the family of architectures we propose for SDN to support ATN. We have a

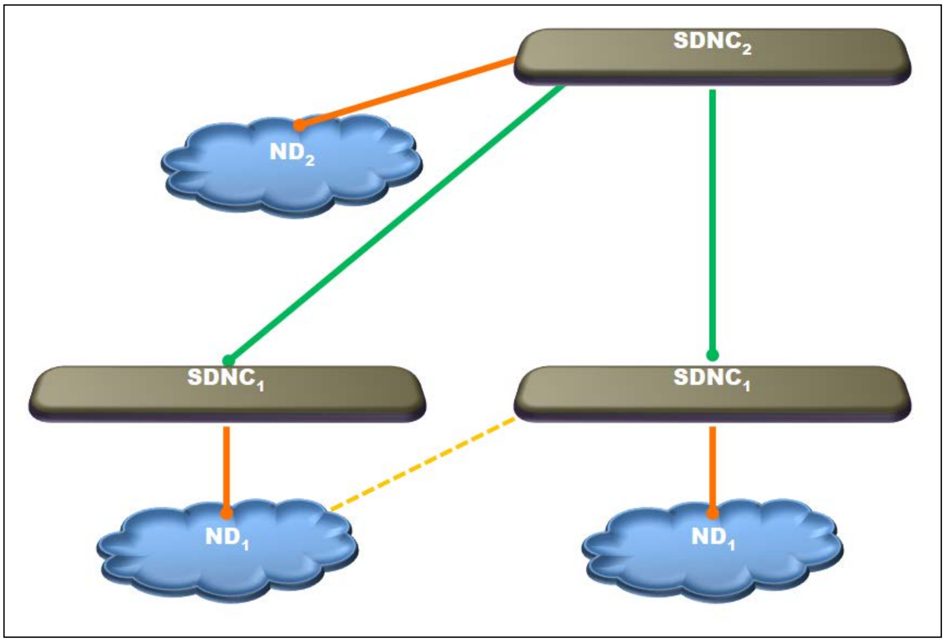


Figure 4: $SDNC_2$ Controls ND_2 Directly and ND_1s Indirectly. *(Source Author)*

pyramid of SDNCs. Each SDNC is assigned the primary responsibility for at least one Network Domain (ND) and uses a standard southbound protocol (such as OpenFlow) for communication with network devices in that ND. This relationship between an SDNC and ND is as shown in Figure 1. An ND may span one or more levels of the ATN hierarchy. Peer SDNCs are responsible for peer NDs. These peer SDNCs communicate among themselves using the emerging SDNi protocols. SDNCs at a higher level of the pyramid control higher levels of the ATN hierarchy. SNDCs between levels could also communicate using SDNi but the content may reflect the hierarchy. In addition to the primary responsibility for an ND, an SDNC also acts as an indirect higher level Controller for the tree of SDNCs subtending to that SDNC. Figure 4 illustrates direct and indirect control domains of an SDNC.

We now highlight several important features of this family of SDN architectures.

1. Each SDNC is physically close to the ND it controls and SDNCs are physically separated from one another when NDs are.
2. The pyramid structure limits the size of an ND controlled directly by an SDNC and reduces the fraction of time devices in an ND cannot communicate with the SDNC controlling that ND.
3. Unlike in the standard SDN, we do not require that all control plane functions for an ND are handled by its SDNC. An important architectural decision is the set of control plane functions that we keep tightly coupled with the data plane and managed by distributed controls as is done currently. Based on the discussion above, the functions which require very time critical communication between data and control plane are candidates for not moving to the SDNC. The decisions may be different for different NDs, especially for NDs at different levels of hierarchy.
4. Like the standard SDNC, each of our SDNCs would create an abstraction of the NDs it directly controls and present this software abstraction to the applications and management. However, as discussed in Figure 4, a higher level SDNC provides indirect controls for the NDs controlled by the SDNCs on the tree. This higher level SDNC should have an abstraction of the entire tree of NDs it controls indirectly, in addition to the abstraction of the ND it controls directly. During operations, it should also have SA of the entire tree of NDs and can use this broader SA to make decisions with broader impact than decisions made by a lower level SDNC can have. An interesting set of questions surfaces.
   a. The first set of questions are the granularity of the network infrastructure and SA an SDNC should carry for indirectly controlled NDs and how these abstractions are generated. It makes sense for the higher level SDNC to get a summarized view of indirectly controlled NDs with a greater focus on inter-ND connectivity and edge-to-edge behavior of the traffic through the ND. The SDNC controlling an ND directly could prepare a summarized version of the current status of that ND and send that summary to the next higher level SDNC controlling that ND indirectly. The higher level SDNC can prepare a summary of the summaries about these multiple NDs and send it up to the next higher level SDNC. This approach creates increasingly broader and less granular abstraction of the network as we move up the hierarchy of ATN. The same will apply to the Situation Awareness if we follow a similar approach in propagating SA up the hierarchy.
   b. Similar questions arise about the control flows from higher level SDNCs to lower level SDNCs and then to the NDs. A higher level SDNC could use its broader SA to select effects that optimize the network behavior across all NDs it controls directly or indirectly.

For directly controlled NDs, these effects are mapped into actions/controls that are communicated to the ND devices. For indirectly controlled NDs, the selected effects will get communicated to the next lower level SDNCs. Each SDNC at that level adds its more detailed knowledge of the NDs to arrive at actions for the NDs controlled directly by that SDNC or more detailed effects for the next lower level, and so on. The following are some examples of effects selected at higher level: Need for connectivity or additional capacity between two units that are separated without adequate communication capacity (this may get mapped into an action of deploying a communication relay at a lower level); Additions and deletions needed to carry out an UTR across many units; Response to a cyber-attack identified by correlating information from several lower level SDNCs.

c. 4a and 4b above reflect the application of Mission Command Philosophy [4] to the control and management of ATNs. Namely, the SA is broader but less granular at higher levels of hierarchy. On the other hand, the intent and desired effects selected at higher level are mapped into more detailed actions and controls closer to where the effects are to be realized. This is critical for scalability of network and human resources. It is also very important when the connectivity between levels of hierarchy is unpredictable and an SNDC needs the ability to take local actions based on the status of its ND and guidance and intent from higher levels.

5. The pyramid structure of the SDNC/ND network requires greater processing capacities and data rates at higher levels of the hierarchy, requirements that are easily met because, as we move up the hierarchy, the mobility reduces, enabling increases in the network data rates and device processing capacities. This architecture also supports hierarchically distributed firewalls with increasing complexity as we move up the ATN and SDNC hierarchy. Such an arrangement would facilitate defense in depth.
6. Pyramid architecture for SDNCs allows peer level connectivity among SDNCs at the same level of hierarchy and cross level connectivity between SDNCs at adjacent levels of hierarchy. This richer connectivity can be used to mitigate concerns about the centralization of controls, processing overload, single point of failure, and cyber-attacks.

   As shown in Figure 5, we can use more than one peer SDNCs to control one ND with one SDNC as the primary and the others as backups. The backups are primaries for other NDs.
7. The pyramid structure of the SDNCs in Figure 3 is used for the cyphertext (black) network carrying encrypted traffic between various plaintext (red) enclaves [4]. Plaintext (red) enclaves may exist at all levels of ATN hierarchy and may have different classification levels. Traffic originating in one red enclave does not typically go to a red enclave at a different level of hierarchy entirely in the red network. Black network connects the pairs of red enclaves. One or more red enclaves can together form a red ND managed by a red SDNC. If traffic is not encrypted, the hierarchy of SDNCs applies to the entire tactical network.

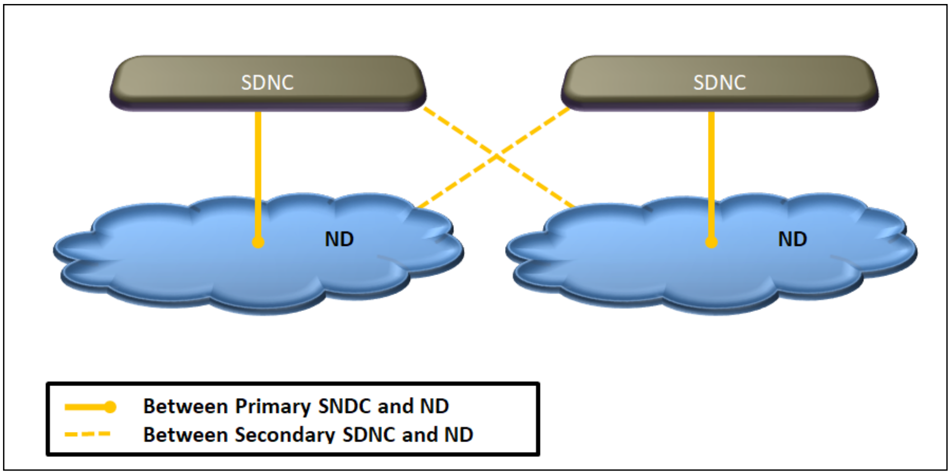


Figure 5: Primary and Backup SDNCs. ***(Source Author)***

AN EXPERIMENTAL SDN ARCHITECTURE AND PROTOTYPE IMPLEMENTATION

As discussed above, NDs at tactical edge (e.g. LTI) may keep some control functions in the forwarding plane. As an example, we consider a hybrid architecture where the mobile nodes in such an ND have local autonomy in making forwarding decisions and SDNCs perform less time critical functions like the following:

- Policy dissemination and enforcement: As an example, nodes could act as distributed firewalls sending suspicious packets to the SDNC for further action. Rules for identifying suspicious packets are disseminated by the SDNC.
- Monitoring and re-configuration of nodes: For example, SDNC can re-configure the nodes per UTR policy.
- Gateway functions in networks with multi-level security: The SDNC could use virtualization technologies to provide cross domain solutions.
- Interfacing with other networks and acting as default router: When a packet needs to be forwarded to a location outside of the ND, the

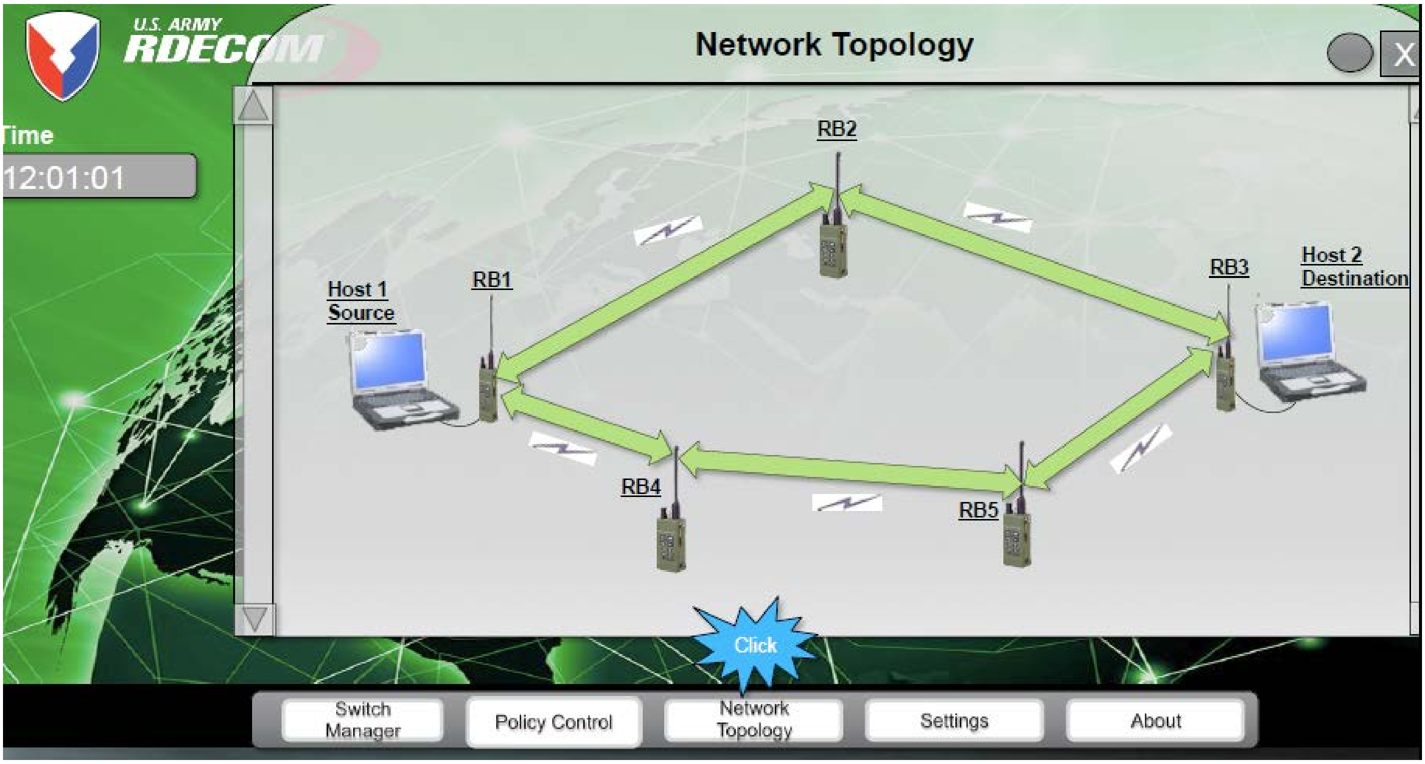


Figure 6: Display of SDN Controller Interface. *(Source Author)*

SDNC can be the default router for this operation. Direct connectivity to a node belonging to another ND is under consideration.

- Overriding of locally computed routes: An example of route override is when the SDNC decides to blacklist a node and instructs all the impacted nodes in the ND to modify their routing tables accordingly.
- Modification of routing parameters: SDNC may indirectly affect routing by modifying the cost of the links used in the computation of routes in its ND.

Our ND uses Transparent Interconnection of Lots of Links (TRILL) [6] for local routing. TRILL enables mobile nodes to function as Routing Bridges (RBridges) using IS-IS routing protocol [7] at link layer, thus combining the advantages of bridges and routers. TRILL offers the following advantages:

- Optimum (minimum cost) multicast & unicast forwarding
- Fast Convergence times
- Minimal configuration due to link layer operations
- Robust loop mitigation and/or preventions with Time to Live (TTL) marking.
- Scaling to large numbers of MAC addresses
- Equal Cost Multi Pathing (ECMP): Load-splitting among multiple paths
- Multi-pathing support for multicast traffic
- Native support for V-LANs.

The control plane consists of a combination of the locally run IS-IS protocol and policy based instructions delivered by the SDNC. Flow Table entries are updated by both the IS-IS protocol and by the SDNC policies. SDNC updates to Flow Tables occur at a slower time scale than IS-IS updates. When there is conflict between IS-IS updates and SDNC updates, the SDNC updates generally override the former, subject to stability constraints.

This hybrid capability is currently being tailored to support scenarios where wireless nodes are automatically reconfigured (IP addresses, frequency spectrum, keys, etc.) based on the requirements of a newly imposed task on the network.

BROAD SET OF QUESTIONS TO BE ADDRESSED

Section 5 describes a single SDNC and single ND architecture with basic routing function in the data plane, which could be advantageous in some NDs in ATN, especially in LTI. The long term goal is to architect the multi-controller SDN for the entire ATN. The family of architectures and functions defined at a high level in Section 4 (Figure 3) above provide a good starting point for discussing the broad set of questions to be addressed:

- Basic cyber security features to build the minimally required trust.
- Choice of control functions that should remain coupled with the data/forwarding plane of each ND. Other control functions will then be moved to the separate control plane in the directly controlling SDNC. The choice may vary among NDs and will be determined by the characteristics of the network connections in that ND and between ND and SDNC, time criticality of the function being considered for move to SDNC and security implications of the move.
- Comparisons of the alternatives

among hierarchically distributed SDNC Networks defined above, and development of a quantitative approach to select the best alternative for a given set of network parameters. In particular,

- Defining NDs and corresponding directly controlling SDNCs. More but smaller NDs create more SDNCs and allow more redundancy, but also increase communication requirements and could result in the use of less powerful SDNCs
- Selecting the number of peer SDNCs serving as backups for an SDNC. Larger number will lead to an increased availability of control functions residing in SDNCs and less time in purely autonomous mode. Even greater agility and cyber resilience could then be provided by moving individual functions among SDNC platforms. However, larger number will also lead to ND abstractions maintained in more SDNCs. The resulting increase in the memory and synchronization burden need to be factored in the decision.
- Number of layers of hierarchy in SDNC network.
- Choice of a higher level SDNC or a peer SDNC as a backup for an SDNC, depending on the expected connectivity and data rates available between SDNC and ND, additional burden on SDNC, and security profiles of the two choices.

- Approach to developing network abstraction at directly controlling SDNC and at indirectly controlling SDNCs. The latter will require aggregation logic and algorithms. A related question is the network Situational Awareness (SA) for the management of the network. What does the network SA mean to a tactical radio user? To the commander of a tactical unit? To the staff of a brigade commander? How do we build this SA hierarchically?
- Approach to building Cyber SA raises questions similar to the ones for the network SA.
- Appropriate use of Moving Target Defense and other agility and deception mechanisms that are facilitated by the hierarchically distributed SDNCs. This may involve SDNC centralization to coordinate changes in the network designed to offer a moving attack surface. It may also involve moving functions away from the compromised platforms, VMs, or Containers.
- Use of VMs vs Containers to implement VEs. Containers are much lighter weights and so more of them can fit in a platform. However, Containers in one platform share the Operating System software so the security implications need to be factored in.
- Use of SDNC to help set up and manage VPNs.
- Use of SDNCs to facilitate UTRs
- Cross Domain Solution between SDNCs.
- Overall security architecture
- Architecture for hybrid SDN, non-SDN operation

REFERENCES

[1] Bloch, K, “Software defined networks (SDN) - Enabling virtualised, programmable infrastructure”, *IEEE Conference on Local Computer Networks Workshop*, October 2013.

[2] Smith, James; Nair, Ravi, “The Architecture of Virtual Machines”. *Computer*, 38 (5), 2005, pp: 32–38.

[3] T. Andersen, “Past, Present and Future of Containers”, *First International Workshop on Container Technologies and Container Clouds*, March 2015

[4] Col. C. J. Ancker, III, U.S. Army, “The Evolution of Mission Command in U.S. Army Doctrine, 1905 to the Present”, *Military Review*, March-April 2013.

[5] Bharat Doshi, “A Prefix Space Partitioning Approach to Scalable Peer Gateway Discovery in Secure Virtual Private Networks”, *IEEE MILCOM’05*, 2005, pp. 2735-2741

[6] R. Perlman, et al, “Routing Bridges (RBridges): Base Protocol Specification”, *IETF RFC 6325* July 2011.

[7] R. Callon, Use of OSI IS-IS for Routing in TCP/IP and Dual Environments, *IETF RFC 1195* December 1990.

ABOUT THE AUTHORS

DR. BHARAT DOSHI retired from the US Army Communications-Electronics Research, Development, and Engineering Center (CERDEC), where he was Senior Research Scientist (ST) for Cyber Security. In that role, he developed strategies and techniques for securing current and new generation networks and information systems against all forms of attacks in the cyber domain. He also developed strategy for the convergence of cyber security, cyber operations, and intelligence. He also led a Department of Defense (DoD)-wide cyber community of interest (Cyber COI), developing priorities and roadmaps for the cyber science and technology across all services and agencies in the DoD. Prior to joining the U.S. Army, Dr. Doshi worked at Johns Hopkins University Applied Physics Laboratory (JHU/APL) for 10 years, Bell Labs for 24 years, and academia for six years. He managed research organizations of up to 200. His personal research and R&D management span the whole gamut of commercial communications and networking technologies as well as major communications, networking, and cyber security programs in the DoD and the Department of Homeland Security. He is a fellow of Bell Labs and a fellow of IEEE. He has published over 140 papers and holds 46 US patents. Dr. Doshi received his Ph.D. from Cornell University. Currently, Dr. Doshi supports Code 31 in the Office of Naval Research as an independent consultant.

DR. CANSEVER is a Program Manager in the U.S. Army Research Office (ARO), where he guides, evaluates and funds fundamental research on Communications and Hybrid Networks and on Multi-User Network Control. He has a Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign. He is conducting research in the areas of networks, cybersecurity, control and game theory. He previously taught at the University of Massachusetts and Boston University and he worked at U.S Army Research Laboratory (ARL), U.S. Army Communications and Electronics Research and Development Center (CERDEC), Johns Hopkins University Applied Physics Laboratory (APL), GTE Laboratories and Bell Laboratories and several start-ups as researcher, chief engineer and CTO.

Discover the value of sharing your DoD-funded research...

Increase peer citations and worldwide dissemination

Advance industry innovation

Maximize R&D dollars, foster collaboration

Leverage results of defense-funded research

Ensure long-term availability and preservation of documents

SUBMIT your research today!

R&E Gateway *Powered by DTIC*

https://go.usa.gov/xUFVJ

Defense Technical Information Center (DTIC) | Fort Belvoir, VA | https://discover.dtic.mil

GAME THEORETIC MODELING OF ADVANCED PERSISTENT THREAT IN INTERNET OF THINGS

By Charles A. Kamhoua, Nandi O. Leslie, and Michael J. Weisman
US Army Research Laboratory, Network Security Branch, Adelphi MD

Cyber-Physical Systems (CPS) and Internet of Things (IoT) devices such as sensors, wearable devices, robots, drones, and autonomous vehicles facilitate the Intelligence, Surveillance and Reconnaissance to Command and Control and battlefield services. However, the extensive use of information and communication technologies in such systems makes them vulnerable to cyber-attacks in the battlefield [1]. These IoT devices are most often designed without considering security [2]. Unprotected IoT devices can be used as "stepping stones" by attackers to launch more sophisticated attacks [3] such as advanced persistent threats (APTs). An APT is a cyber-attack in

which a malicious adversary gains access to a network and remains undetected for a long period of time. A later stage of APT is the "lateral movement" stage, where attackers use benign computer features to move step-by-step deeper into the network in a stealthy manner [4-6]. For instance, it has been reported that Samsung's smart fridge could be used to steal a user Gmail login [7]. One can imagine several additional steps such as sending fishing emails to friend or coworker followed by privilege escalation. The above challenges and the high risk and consequence of IoT attacks in the battlefield drive the need to accelerate basic research on IoT security. We are investigating proactive defense of IoT networks including cyber deception, cyber resilience, cyber agility —this process is also called Moving Target Defense (MTD). We consider the most intelligent adversaries that are able to launch sophisticated attacks (e.g. APTs). We also look into the scientific foundation of cyber security. Theoretical constructs and mathematical abstractions provide a rigorous scientific basis for cyber security because they allow for reasoning quantitatively about cyber-attacks. In particular, game theory provides a rich mathematical tool to analyze conflicts within strategic interactions and thereby gain a deeper understanding of cyber security issues. By definition, game theory is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers" [8]. The level of sophistication of recent cyber-attacks justifies our assumption of attackers' rationality and thus the need for an intelligent defense mechanism based on game theory.

ADVANCED PERSISTENT THREAT

The cyber kill chain in Figure 1 shows the stages of an APT (red) as well as the defender's best response at each stage (blue). At the *reconnaissance* phase, the attacker scans the system to identify potential vulnerabilities, understand the network topology, and find critical targets. This is followed by an *exploit* of a vulnerability to *command & control* a node. From that node, the attacker proceeds to a *privilege escalation* to gain elevated access that will enable *lateral movement* to reach a critical *target*. A proactive defense mechanism includes all scheme the defender can implement to *protect* the network before a cyber-attack is launched or early in the reconnaissance phase. Intrusion prevention systems (IPS), including firewalls and anti-virus, are designed to protect networks against cyber-attack attempts. However, these cyber systems tend to have inadequate prediction performance and misidentify malicious network traffic (e.g., malware, botnet) as benign—these packets are called false negatives. In addition to IPS, research shows that statistical learning techniques can accurately forecast or predict the timing and frequency [31] of cyber-attacks, based on network and organizational observations (e.g., domain name system traffic, network security policy). When proactive defense fail, the defender tries to *detect* the intruder, *deny* or *disrupt* malicious action or at least *contain* the attack. In the worst case of a successful attack, the defender should be prepared to quickly *recover*. The use of IoT devices in the battlefield increase the attack surface that our adversary can exploit. A game theoretic approach is suitable for all stage of an APT, from proactive cyber defense, to fighting through an attack in progress [9], or survive and recover from a successful attack [10]. Our prior work [9] uses stochastic game approach to contain a CPS attack in the lateral movement phase.

PROACTIVE CYBER DEFENSE

There are several challenges associated with IoT security compared to securing traditional information technology (IT) systems. First, IoT devices are rapidly mass produced to be low-cost commodity items without security protection in their original design. Even if a device initially has some security features, many IoT manufacturers do not provide any security updates and thus IoT devices can become unsecure as hackers discover new vulnerabilities. Second, IoT devices are highly dynamic, mobile, heterogeneous and lack common standards. Additionally, they have a limited battery capacity, memory, and processing power and cannot integrate standard encryption algorithms and security protocols. Third, it is imperative to understand the natural world, the physical process(es) under IoT control, and how these real-world processes can be compromised before to recommend any relevant security counter measure. When faced with these challenges to IoT security, a proactive approach is better suited to the defense of IoT assets. A proactive IoT defense allows us to plan in advance, analyze all cyber threats and gain a precise understanding of potential vulnerabilities before a cyber-attack is launched. Cyber deception, cyber agility [also referred to as Moving Target Defense (MTD) in the literature], and cyber resilience are the main components of a proactive cyber defense. Those components can be used separately or in conjunction to protect IoT.

Cyber deception is any attempt to disguise a network and impair the attacker's decision with false information to protect critical nodes. Deception can delay a cyber-attack by increasing uncertainty. Deception also forces the attacker to perform more trial and error in the reconnaissance phase which increases the probability of intruder detection. The use of honeypots is a basic form of cyber deception used to create the appearance of important targets to the attacker. Honeypots also help to identify attackers and provide a means to learn about their behaviors in a safe environment. The attacker's strategies learned via the use of honeypots aid in securing critical components [11]. A honeynet is a decoy network that contains one or more honeypots. Valuable deception techniques must confuse the attacker while being transparent to the defender and legitimate users [12]. Advanced deception techniques can dynamically hide or create fake vulnerabilities, data, protocols, communication links, software and applications. However, given enough time, an attacker may be able to discover the defender's deception strategy. Therefore, a sophisticated cyber deception technique is most often combined with cyber agility.

Cyber agility is the dynamic reconfiguration of network parameters, components, topology, and protocols to oppose an attacker's ability to collect information about the system. A static configuration gives enough time to attackers to learn about the system and identify potential vulnerabilities or exploits in the reconnaissance phase. An agility strategy randomly changes the network pattern faster than an attacker can learn. The Army Research Laboratory's Cyber Security Collaborative Research Alliance is currently investigating game theoretic approaches to cyber agility [13].

Cyber resilience refers to the network capability to continuously maintain

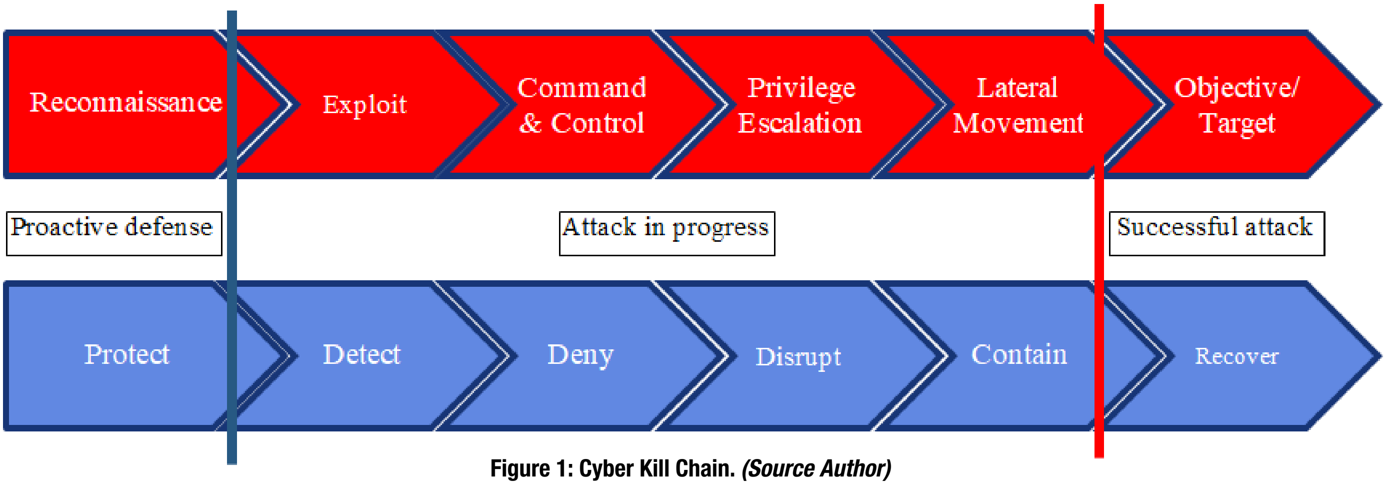


Figure 1: Cyber Kill Chain. ***(Source Author)***

mission essential functions after a cyber-attack. Resiliency must be an important consideration in IoT design for a number of reasons. First, the military uses commercial off-the-shelf (COTS) IoT devices available to the general public. Second, IoT devices interconnect with the commercial network not owned by the military. Third, most IoT devices are designed without concern for security and thus contain many vulnerabilities that can be exploited as weak links to gain access to more important targets. Fourth, it is beyond the capability of a developer or a network administrator to predict all natural failure and malicious attacks because of the increased interconnection, interdependency, and complexity of IoT networks. Those facts dictate our pessimistic view that some attacks may be successful regardless of efforts to maintain best practices in the areas of deception and agility. We should proactively design IoT networks while considering remediation against the worst case scenario, that of a successful attack. Resilient mechanisms sometimes involve system replication, to add redundancy and avoid a single point of failure [14]. Furthermore, the replica can be diversified to counter the attacker's ability to exploit the same vulnerability in all replicas.

GAME THEORY FOR ADVANCED PERSISTENT THREAT

A game in normal form is given by a set of players, the set of strategies available to each player, and a payoff function that allocates an award to each player given any combination of strategies representing the choice made by each player. Game theory is suited for proactive cyber defense because of its predictive power. The solution to a cyber security game is its Nash equilibrium (or its derivative). At a Nash equilibrium profile, no player can increase his payoff after a unilateral deviation. Also, each player is playing his best response to other players' strategies. Therefore, the defender can use the Nash equilibrium profile to predict the attacker's best action. The prediction power of game theory, combined with cyber deception, cyber agility, and cyber resilience can form the basis of a robust framework for proactive cyber defense.

Each player in a game attempts to maximize his payoff based on his information and his belief about others players' information. If the set of players, strategies and payoff function is common knowledge, we have a game of complete information. Otherwise, we have an incomplete information game. Therefore, cyber deception and agility which interfere with the attacker's ability to gain accurate information produces a game of incomplete information with the potential to diminish the attacker's payoff. However, one must also carefully consider skillful attackers able to deceive the defender. A skillful attacker can behave as if the defender's deception is effective to misguide the defender to reveal his mode of operation. A useful game model must consider several other possibilities that relate to incomplete information. A skillful attacker may develop unknown exploits (e.g., zero-day vulnerabilities), hide his true intent (i.e., target, payoff), or operate undetected for a long time —this is the intent of an APT).

Recently, there has been increased interest in the literature to apply game theory to cyber deception [15], [16] agility [17] resilience [10], [14] intrusion detection [18] lateral movement [9] and APT [19]. Cuong et al. [20] provide a detailed survey of these game-theoretic applications to cybersecurity. However, those works are restricted to a single stage of the kill chain and do not consider

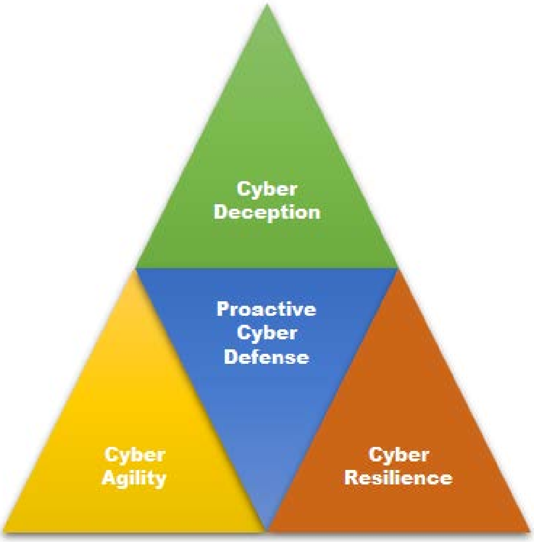


Figure 2: Proactive cyber defense. ***(Source Author)***

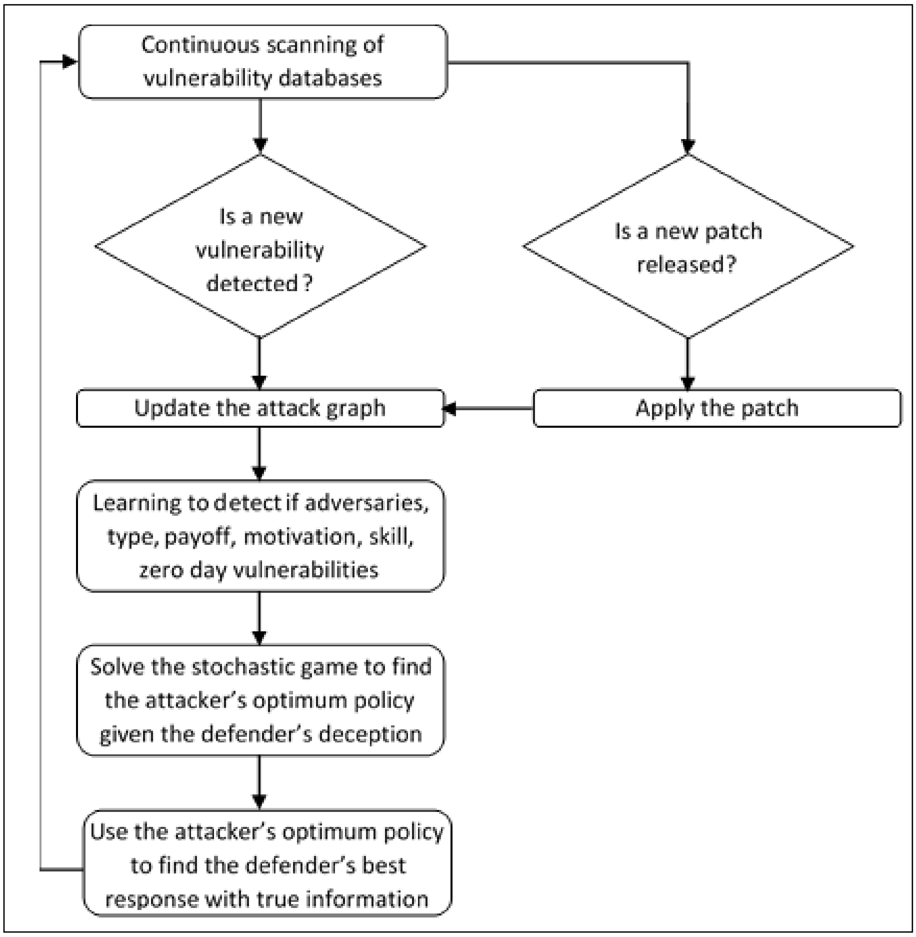

Continuous scanning of vulnerability databases
Is a new vulnerability detected ?
Is a new patch released?
Update the attack graph
Apply the patch
Learning to detect if adversaries, type, payoff, motivation, skill, zero day vulnerabilities
Solve the stochastic game to find the attacker's optimum policy given the defender's deception
Use the attacker's optimum policy to find the defender's best response with true information

Figure 3: Best response engine for self-secured systems. ***(Source Author)***

specific constraints of military IoT. We are currently investigating end-to-end defense mechanisms that can deal with a cyber-attack at multiple stages. The goal is to design mission-aware IoT with an autonomous cyber response capability.

We present a high-level description of our current approach to build an autonomous response [21] to IoT security with deception capability, learning for detection, and containing lateral movement. Figure 3 shows the diagram of the engine. From the configuration files of hosts (e.g., computers, operating systems, application, firewalls, servers, routers), the engine can compute the topology of the IoT network and generate the attack-graph. Two nodes V1 and V2 are connected in the attack-graph if there is a port, a protocol, and a vulnerable application on V2 that can be exploited to compromise V2 from V1. The engine incorporates a scanning tool capable of discovering new vulnerabilities from public vulnerabilities databases such as National Vulnerability Database (NVD) [22]. Once a new vulnerability is detected, the attack-graph is updated by adding new edges to the graph. We use the Common Vulnerability Scoring System (CVSS) [23] to compute a relevant assessment on how the attacker can access a vulnerability, how complex it is to exploit the vulnerability, and the number of times one must to authenticate (if any) in order to exploit the vulnerability. If a new patch is released from NVD, then the system will automatically apply the patch and updates the attack-graph by removing all the edges corresponding to the patched vulnerability.

Before an attack is detected, a dynamic cyber deception mechanism is implemented to mislead the attacker and minimize the attacker's impact on the IoT network. An adversarial machine learning approach robust to intelligent manipulation is implemented to detect these characteristics about the attacker: payoff, motivation, skill, and potential zero-day vulnerabilities not in the NVD database. The learning algorithm has to quickly converge to be compatible with fast changes in network topologies.

When an attacker is detected, a two-player stochastic game representing the interactions between the attacker and the defender is initiated. In this game, the states represent the nodes of the attack-graph and transitions correspond to the edge-vulnerabilities that the attacker can exploit to move laterally. The solution of the game will give the attacker's optimal policy with deception.

Given the attacker's optimum policy, the defender's best response is calculated with accurate information. The best response at any state of the game will allow the system to quickly recover to a secure state. The system uses the optimum policy to automatically disconnect or self-reconfigure vulnerable services and thus slow down the progression of the attack at any node of the system. Finally, continuous learning and scanning of vulnerabilities allows the system to adapt to new threats.

CHALLENGES AND FUTURE WORK

Modeling IoT presents several challenges that will be addressed in our future work. First, IoT devices may be autonomous and may not have a global knowledge of the network [21]. Also, directives sent from a central command to IoT devices may be delayed or lost. Therefore, a distributed security mechanism is more appropriate in IoT compared to the traditional attacker vs defender model.

Moreover, an IoT network may be subject to several simultaneous attacks from different point of the network, and at different stages of the kill chain. Those attackers may be acting independently or in collusion. The case of colluding attackers [24] is particularly challenging.

Monitoring is another key consideration. There are scenarios, where players cannot observe the other player's actions directly but can only observe an imperfect noisy signal correlated to those actions. For instance, the defender may not know exactly the last edge-vulnerability exploited by the attacker or can only infer the new position of the attacker in the attack graph.

Furthermore, IoT devices may have a short time to process a large amount of information in a complex environment with finite memory and limited computational power. This results in the limited rationality of IoT nodes which result in incorrect decisions that deviates from rational equilibrium behavior. Prior work has used evolutionary game theory [25]-[26] and prospect theory [27] to account for limited rationality.

Machine learning entails improvement of a computer's performance on a given task with experience. Machine learning algorithms and approaches are also important to our proposed framework for proactive cyber defense. Specifically, using 60 different classifiers (or supervised learning algorithms), Lee et al. [28] deploy honeypots and accurately identify social spammers on Twitter and MySpace. Furthermore, it is known that evolution-based algorithms that combine machine learning and genetic algorithms can advance cyber agility by periodically changing the system's configuration and attack surface [29]-[30]. In addition, a key aspect of proactive cyber defense and cyber resilience is cyber-risk quantification—this process involves predicting the number of successful cyber-attacks [31]. Moreover, each of these components of proactive cyber defense require robust intrusion detection systems (IDS) that are behavior or anomaly-based to detect the zero-day cyber-attacks instead of the classical signature-based detection models that are found exclusively in many IDS. For example, Alazab et al. [32] demonstrated that using support vector machines, a type of supervised learning algorithm, obfuscated malware can be effectively detected. However, there is the need to fully understand the limitation and vulnerabilities of machine learning algorithm [33]. The potential manipulation of those algorithms by an intelligent adversary introduces new threats that need to be investigated. In fact, all IoT devices rely on algorithms based on artificial intelligence and machine learning to operate. Future battlefields will have IoT devices (e.g., robots, drones) from opposing armies [1]. Those IoT devices may have other IoT entities as adversaries. An easy way to win a battle will be to manipulate the algorithm from the opposing IoT. The new and fertile field of adversarial machine learning is at the intersection of game theory and machine learning is promising.

REFERENCES

[1] A. Kott, A. Swami, B. West. "The internet of battle things". IEEE Computer, Dec. 2016.

[2] Edward J. M. Colbert and Alexander Kott, "Cybersecurity of SCADA and Other Industrial Control Systems", Springer International Publishing, 2016.

[3] Michael Hiltzik, "Apple, the FBI, and the Internet of Things: Your whole house is open to attack" Los Angeles Times, March 1, 2016, available online: http://www.latimes.com/business/hiltzik/la-fi-mh-apple-the-internet-of-things-vulnerable-to-attack-20160301-column.html [July 11, 2017]

[4] John R Johnson, et al., "A graph analytical metric for mitigating advanced persistent threat," 2013 IEEE Intelligence and Security Informatics Conference,

[5] E. Purvine, et al., "A graph based impact metric for mitigating lateral movement cyber-attacks," ACM Workshop on Automated Decision Making for Active Cyber Defense, 2016

[6] J. Vukalovic and D. Delija., "Advanced Persistent Threats, Detection and Defense" IEEE Information and Communication Technology, Electronics and Microelectronics (MICRO) 2015

[7] Stacey Higginbotham, "Samsung's smart fridge could be used to steal your Gmail login" Fortune, Aug 24, 2015, available online: http://fortune.com/2015/08/24/samsungs-smart-fridge-hacked/ [July 11, 2017].

[8] Roger B. Myerson, "Game Theory: Analysis of Conflict" Harvard University Press, (1991).

[9] Gael Kamdem, Charles A. Kamhoua, Yue M. Lu, Sachin Shetty, Laurent Njilla "A Markov Game Theoretic Approach for Power Grid Security" in the proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS 2017) workshop, Atlanta, GA, June 2017.

[10] Charles A. Kamhoua, Kevin Kwiat, Joon Park "Surviving in Cyberspace: A Game Theoretic Approach" in the Journal of Communications, Special Issue on Future Directions in Computing and Networking, Academy Publisher, Vol. 7, No 6, June 2012.

[11] M. Zhao, F. D'Ugard, K. A. Kwiat and C. A. Kamhoua, "Multi-level VM replication based survivability for mission-critical cloud computing," 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 1351-1356.

[12] Dave Climek, Anthony Macera, Walt Tirenin "Cyber Deception" Special Issue of Cyber Security and Information Systems Information Analysis Center (CSIAC) Journal, Focus on Air Force Research Laboratory's Information Directorate, Volume 4, Number 1, December 2015.

[13] Patrick McDaniel, Ananthram Swami "The Cyber Security Collaborative Research Alliance: Unifying Detection, Agility, and Risk in Mission-Oriented Cyber Decision Making" Cyber Science & Technology at the Army Research Laboratory (ARL), Volume: 5 Number: 1, January 2017.

[14] Charles A. Kamhoua, Patrick Hurley, Kevin Kwiat, Joon Park "Resilient Voting Mechanisms for Mission Survivability in Cyberspace: Combining Replication and Diversity" in the International Journal of Network Security and Its Applications (IJNSA), Vol.4, No.4, July 2012.

[15] Karel Durkota, Viliam Lisy, Branislav Bosansk and Christopher Kiekintveld, "Approximate Solutions for Attack Graph Games with Imperfect Information, in the proceedings of the International Conference on Decision and Game Theory for Security, London, UK, November 2015.

[16] Hayreddin Ceker, Jun Zhuang, Shambhu Upadhyaya, La Quang Duy, Soong Boon Hee "Deception-based Game Theoretical Approach to Mitigate DoS Attacks" in the proceedings of the International Conference on Decision and Game Theory for Security, New York, November 2016.

[17] Q Zhu, T Basar "Game-theoretic approach to feedback-driven multi-stage moving target defense" in the proceedings of the International Conference on Decision and Game Theory for Security, Fort Worth, TX, November 2013.

[18] Tansu Alpcan, Tamer Basar "A game theoretic analysis of intrusion detection in access control systems" in the proceedings of the 43rd IEEE Conference on Decision and Control, 2004.

[19] Muhammed Sayin, Tamer Basar "Secure Sensor Design for Cyber-Physical Systems Against Advanced Persistent Threats" in the proceedings of the International Conference on Decision and Game Theory for Security, Vienna, Austria, November 2017.

[20] Cuong T. Do, Nguyen H. Tran, Choongseon Hong, Charles A. Kamhoua, Kevin A. Kwiat, Erik Blasch, Shaolei Ren, Niki Pissinou, Sundaraja Sitharama Iyengar "Game Theory for Cyber Security and Privacy" ACM Computing Surveys (CSUR), Volume 50, Issue 2, Article No. 30, June 2017.

[21] Leslie, N., Singh, R., Rigaki, M., & Yang, S. (in press). "Applying Machine Learning Algorithms to Cyber-Physical System Security Challenges". NATO Technical Report for NATO STO IST-152 Task Group on Intelligent Autonomous Cyber Defence and Resilience. NATO.

[22] National Vulnerability Database, https://nvd.nist.gov/

[23] Common Vulnerability Scoring System, nvd.nist.gov/cvss

[24] Abhishek Roy, Charles A. Kamhoua, Prasant Mohapatra, "Game Theoretic Characterization of Collusive Behavior among Attackers" in the proceedings of the 2018 IEEE Conference on Computer Communications (INFOCOM), Honolulu, HI, April 2018.

[25] C. A. Kamhoua, N. Pissinou and K. Makki, "Game Theoretic Modeling and Evolution of Trust in Autonomous Multi-Hop Networks: Application to Network Security and Privacy," 2011 IEEE International Conference on Communications (ICC), Kyoto, 2011, pp. 1-6.

[26] C. A. Kamhoua, N. Pissinou, J. Miller and S. K. Makki, "Mitigating routing misbehavior in multi-hop networks using evolutionary game theory," 2010 IEEE Globecom Workshops, Miami, FL, 2010, pp. 1957-1962.

[27] W. Saad; A. Sanjab; Y. Wang; C. Kamhoua; K. Kwiat, "Hardware Trojan Detection Game: A Prospect-Theoretic Approach," in IEEE Transactions on Vehicular Technology , vol.PP, no.99, pp.1-1.

[28] Lee K, Caverlee J, Webb S. Uncovering social spammers: social honeypots+ machine learning. InProceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval 2010 Jul 19 (pp. 435-442). ACM.

[29] John DJ, Smith RW, Turkett WH, Cañas DA, Fulp EW. Evolutionary based moving target cyber defense. InProceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation 2014 Jul 12 (pp. 1261-1268). ACM.

[30] Zhuang R, DeLoach SA, Ou X. Towards a theory of moving target defense. In Proceedings of the First ACM Workshop on Moving Target Defense 2014 Nov 7 (pp. 31-40). ACM.

[31] Leslie, N., Harang, R. E., Knachel, L. P., & Kott, A. (2017). "Statistical models for the number of successful cyber intrusions". The Journal of Defense Modeling and Simulation, 1548512917715342.

[32] Alazab, M., Layton, R., Venkataraman, S. and Watters, P., 2010. Malware detection based on structural and behavioural features of API calls.

[33] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Celik, and A. Swami. The limitations of deep learning in adversarial settings. Proceedings of the IEEE European Symposium on Security & Privacy, 2016.

ABOUT THE AUTHORS

CHARLES A. KAMHOUA is a researcher at the Network Security Branch of the U.S. Army Research Laboratory (ARL) in Adelphi, MD, where he is responsible for conducting and directing basic research in the area of game theory applied to cyber security. Prior to joining the Army Research Laboratory, he was a researcher at the U.S. Air Force Research Laboratory (AFRL), Rome, New York for 6 years and an educator in different academic institutions for more than 10 years. He has held visiting research positions at the University of Oxford and Harvard University. He has co-authored more than 100 peer-reviewed journal and conference papers. He has presented over 40 invited keynote and distinguished speeches and has co-organized over 10 conferences and workshops. He has mentored more than 50 young scholars, including students, postdocs, and AFRL Summer Faculty Fellowship scholars. He has been recognized for his scholarship and leadership with numerous prestigious awards, including the 2017 AFRL Information Directorate Basic Research Award "For Outstanding Achievements in Basic Research," the 2017 Fred I. Diamond Award for the best paper published at AFRL's Information Directorate, 40 Air Force Notable Achievement Awards, the 2016 FIU Charles E. Perry Young Alumni Visionary Award, the 2015 Black Engineer of the Year Award (BEYA), the 2015 NSBE Golden Torch Award—Pioneer of the Year, and selection to the 2015 Heidelberg Laureate Forum, to name a few. He received a B.S. in electronics from the University of Douala (ENSET), Cameroon, in 1999, an M.S. in Telecommunication and Networking from Florida International University (FIU) in 2008, and a Ph.D. in Electrical Engineering from FIU in 2011. He is currently an advisor for the National Research Council, a member of the FIU alumni association and ACM, and a senior member of IEEE.

NANDI O. LESLIE is a senior principal engineer at Raytheon, serving as a researcher in the Network Science Division at the U.S. Army Research Laboratory. Her research interests are focused on cyber security and resilience quantification and assessments explaining what makes networked devices vulnerable to cyber-attacks—and on using machine learning and other computational modeling approaches to predict, detect, and/or block malicious or anomalous network traffic. At Systems Planning and Analysis, Inc. 2007-2015, she led and contributed to sensor performance projects for the Navy's Submarine Security and Technology Program, and she developed modeling approaches, using dynamical systems and stochastic processes, to understand search and detection processes and patterns in heterogeneous and dynamic oceanographic and atmospheric environmental conditions. She received her B.S. in Mathematics from Howard University in 1999 and her Ph.D. in Applied and Computational Mathematics from Princeton University in 2005. She was a Postdoctoral Researcher in Mathematics at University of Maryland, College Park from 2005 to 2007. Currently, she is a research adviser for the National Research Council.

MICHAEL J. WEISMAN received a BE in Electrical Engineering from the Cooper Union for the Advancement of Science and Art and a PhD in Applied Mathematics from Harvard University in the area of computer vision. He has previously been a member of the Senior Technical Staff at the JHU Applied Physics Laboratory and Technical Staff at MIT Lincoln Laboratory where he led research on multitarget tracking and data fusion for ballistic missile defense. Dr. Weisman is presently employed as a mathematician in the Computational and Information Sciences Directorate at the US Army Research Laboratory where he is leading research on machine learning and mathematical modeling for cyber security.

CSIAC Cyber Security & Information Systems Information Analysis Center

Call for Papers for Publication

CSIAC is chartered to ***leverage the best practices and expertise from government, industry and academia*** in order ***to promote technology domain awareness and solve the most critically challenging scientific and technical (S&T) problems.***

As part of that mission, CSIAC publishes the *Journal of Cyber Security and Information Systems*, focusing on scientific and technical research & development, methods and processes, policies and standards, security, reliability, quality, and lessons learned case histories. Contributing authors realize the benefits of being published in a highly-respected resource within the technical community, with an enormous reach across the Department of Defense and the broader scientific community (23k+ subscribers). Demonstrate your expertise and accomplishments or pose the challenging questions for further thought in the resource that reaches fellow Subject Matter Experts (SMEs) developing the solutions to support the warfighter.

To Submit an Article

Visit: **https://www.csiac.org/csiac-journal-article-submission-policy/**

CSIAC is currently accepting articles submitted by the professional community for consideration in the following topic areas:

- Advances in AI and machine learning, deep learning, cognitive computing, intelligent agent, chatbot
- AI applications in industry, business, healthcare, and education and training
- AI for enhancing information security and privacy
- Biometric Identity Management
- Trust, resilience, privacy and security issues in AI applications
- Testing and validation of AI and ML applications
- Security automation techniques/real-time incident response
- IoT, Smart Cities, Connected & Autonomous Vehicles (CAV)
- 5G networks/IoT critical use cases

(A non-exhaustive set of topics)

For other related topics on Artificial Intelligence (AI) and Machine Learning (ML), please visit:

https://www.csiac.org/ai
or
https://www.csiac.org/ml

WEBINAR

Leveraging Machine Learning: How to Achieve the Right Balance Between Humans and Automation to Optimize Outcomes

PODCAST

Tricking Neural Networks, CIA Impersonation, International Data Privacy, and Encrypted Malware Detection

REFERENCE DOCUMENT

Learning to Win: Making the Case for Autonomous Cyber Security Solutions

PROVIDING CYBER SITUATIONAL AWARENESS

on Defense Platform Networks

By: Patrick M. Hayden, David K. Woolrich, and Katherine D. Sobolewski

Modern defense platforms are at increasing risk of cyber-attack from sophisticated adversaries. These platforms do not currently provide the situational awareness necessary to identify when they are under cyber-attack, nor to detect that a constituent subsystem may be in a compromised state. Long-term improvements can be made to the security posture of these platforms by incorporating modern secure design best practices, but this is a time-consuming and costly task. Monitoring platform communication networks for malicious activity is an attractive solution for achieving improved cyber security on defense platforms in the near term. This article presents our research into the susceptibility of modern defense platforms to cyber-attack, and the suitability of platform-based intrusion detection systems in addressing this threat. We discuss risk factors contributing to cyber access, then describe a range of platform cyber-attack classes while considering the observables and indicators present on the embedded platform networks. Finally, we examine factors and considerations relating to implementation of a "Cyber Warning Receiver" solution approach for detection of such attacks.

THE THREAT IS REAL

For as long as weapons system platforms have been called upon to perform missions in contested spaces, the military has sought to protect the warfighter by equipping these platforms with survivability equipment. This equipment detects threats from across the various domains in which the platform operates, and alerts operators while taking appropriate response measures. As technology and connectivity of these platforms evolves and increasing sophistication is realized through automation, a new threat domain has emerged. This threat lurks in the dark, escaping detection by human eyes and ears, yet it has a clear potential for harm to the warfighter and to the mission. This is the cyber threat, and it is real.

Cyber-attacks become a credible threat if there is a reasonable expectation that a malicious actor could gain access to a defense platform, achieve a persistent malware presence, and subsequently trigger this malware to impart a damaging effect. In cyberspace, there are no concrete boundaries or borders. Cyber-attacks are not typically encumbered by range or timing. A malicious actor in a faraway land could achieve a latent presence and leverage it at a critical moment in the future to achieve their end goals. They could affect a single platform or an entire compromised squadron simultaneously.

LESSONS FROM INDUSTRY

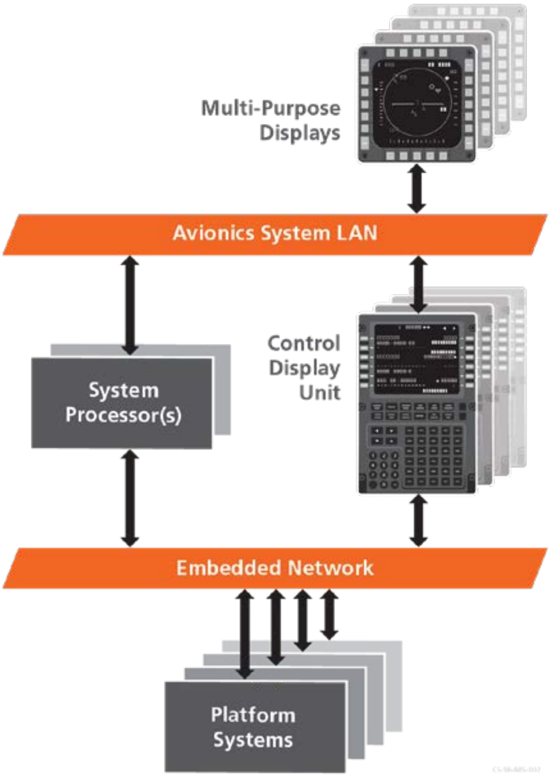
While there is a lack of openly documented cyber-attacks against Department of Defense (DoD) platforms, published examples against similar systems in other industries provide a compelling case for the feasibility of such attacks. We hear more and more about attacks against embedded systems and other smart devices. Attacks originate from threats that range from individual troublemakers to state-sponsored hacking groups. These attacks can be foul-mouthed hackers yelling at children via smart baby monitors [1], using SmartTVs as entrance points to home networks [2], entire automobiles being taken over remotely [3], or debilitating modification of industrial control processes [4].

In 2015, security researchers Dr. Charlie Miller and Chris Valasek were able to remotely access an unaltered SUV, controlling everything from the volume of the radio, to the transmission and steering systems. They first gained access to through a USB maintenance port, then eventually through its onboard cellular network. By traversing multiple subsystems, they ultimately controlled physical functions of the SUV from their hotel room while the vehicle was traveling on a highway.

In 2017, security consultants ARS were able to demonstrate the insertion of malicious code over a broadcasted TV signal. The transmitted code was able to exploit a vulnerability in the smart TV's web browser, enabling root access for the attacker. If a broadcast station were compromised, this attack could be delivered to any vulnerable TV within the broadcast towers' range.

As systems become more complex and gain more parts, supply chains for devices and systems become more spread out and global. This creates difficultly in validating the pedigree of 100% of the components on any one system. A 2017 Defense Science Board Task force on Cyber Supply Chain confirms the supply chain to be a real risk to DoD assets.

The examples above represent three distinct attack access vectors against embedded systems: supply chain compromise (microprocessor compromise), maintenance pathways (vehicle USB), and compromising data links (broadcasted malware in TV signal). Current trends in weapons system platform modernization suggest that these same vectors are also applicable to defense platforms.



PARALLEL SECURITY APPROACHES

The trends of increasing computer automation and platform interconnectivity are here to stay, as they enable distinct tactical advantages. Platform security must improve to address the associated risks head on.

Two complimentary approaches are common when it comes to traditional IT security measures. These apply in the world of defense platforms as well. The first is host-based security, where the security of the individual boxes on a network are improved to achieve increased security for the system overall. With defense platforms, the diversity of subsystems on a given platform means there is usually no single silver bullet solution for host-based protection. Although important, this complicating factor makes application to legacy platforms time-consuming and costly. The second approach is network-based security, where communications between hosts on a network are monitored to detect and potentially intercept malicious activity. We explore this alternative to address near-term improvement in platform security.

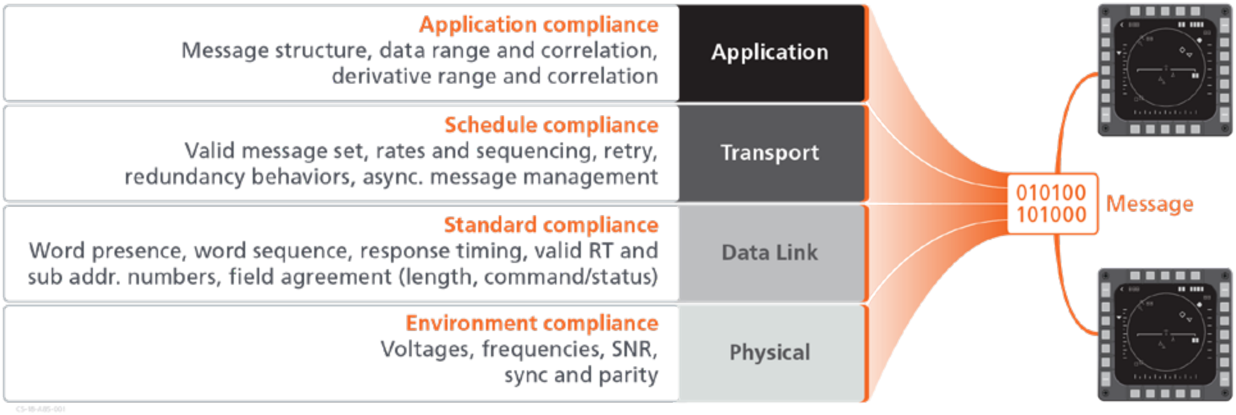

Application compliance
Message structure, data range and correlation, derivative range and correlation
Application
Schedule compliance
Valid message set, rates and sequencing, retry, redundancy behaviors, async. message management
Transport
Standard compliance
Word presence, word sequence, response timing, valid RT and sub addr. numbers, field agreement (length, command/status)
Data Link
Environment compliance
Voltages, frequencies, SNR, sync and parity
Physical
010100
101000
Message

Figure 2 – Common Embedded Network Layers and Observables

NETWORK LOCKDOWN

Embedded networks form the backbone for communications between platform subsystems. They provide the critical link between interface equipment like displays and keypads, and the endpoint devices that actually implement mission essential control or measurement capabilities. The actions necessary to conduct a cyber-attack, and the resulting effects will, in the majority of cases, be observable via these networks.

A common set of networks cover the vast majority of communications occurring on today's defense platforms. In particular, the U.S. Army's Common Avionics Architecture System (CAAS) depicted in Figure 1 relies heavily on Ethernet and MIL-STD-1553 (or fiber optic 1773) networks. Other common embedded interfaces include CANBus, ARINC 429, RS-232, RS-422, analog and discrete signals. A Cyber Warning Receiver, designed to look specifically for malicious activity on these networks can provide the broadly applicable solution necessary to achieve near-term game-changing platform security enhancement. A network focus enables rapid adaptation to various platforms, which would provide immense benefit to the cyber security posture of the overall fleet.

CYBER ATTACKS AND EMBEDDED NETWORKS

The breadth of published work on platform embedded network security is small in comparison to research for similar consumer, commercial, and industrial networks. Such networks are more openly accessible to security researchers for characterization. Our ongoing research has shown that many of the attack types conceived for other network types are also applicable to platform embedded networks. An overarching theme is that these networks do not provide any security features, such as authentication or encryption that would mitigate such misuse.

The attack types available depend on the specific foothold an attacker has achieved on a platform. In general, there are several positions an attacker might hold on a platform with respect to a system:

1. Attacker presence on systems outside the network that leverage data sent or received via the network;
2. Presence on a Remote Terminal / Slave / Receiving device connected to the network;
3. Presence on a Bus Controller / Master/ Transmitting device for the network; and
4. Multiple points of presence creating a combination of these states

Given this set of states, some of the attack types we've described and characterized are:

- Methods by which a compromised host could initiate new messages, remove existing messages, or intercept and modify data in transit between other terminals.
- Methods by which a compromised host could impersonate a different terminal or attempt to escalate its role in controlling the network.
- Methods by which any compromised host on the network could deny messaging between other terminals.
- Attacks in which basic rules and conventions of the data exchange protocol or application layer protocols in use are violated.
- Attacks in which a compromised host deliberately sends incorrect data to another host as part of the normal data exchange. This could include sensor data, control commands, system status or other information.

Consideration of possible attack types and characterization of their effects helps inform a robust design for a

> *"A Cyber Warning Receiver, designed to look specifically for malicious activity on embedded networks can provide the broadly applicable solution necessary to achieve near-term game-changing platform security enhancement."*

platform security detection system like a Cyber Warning Receiver.

ATTACK OBSERVABLES

As the attacks described above take place on an embedded network, they produce side effects that are observable to a high-fidelity monitor. Embedded networks can be logically organized into several network layers. It is convenient to apply these layers when considering the various observables present. Although observables may vary by specific network type, Figure 2 provides a general summary of some common examples.

The bottom layer is the physical layer, which contains observables relating to the fundamental electrical environment necessary for proper operation of the network. Certain attacks can cause disturbances at this level, especially in cases where misuse of the network causes message collisions.

The data link layer handles message addressing. At this level we can detect that only valid addresses and sub-addresses are present, and also that the expected message structure is intact, including allowed message types and expected word sequences for the hosts involved.

The transport layer handles details such as message delivery rates, schedules, and stateful transactions. At this layer we can verify that the system is using the set of messages expected to occur as part of the schedule, with the appropriate sequence and timing.

The application layer contains the core data of the message. The application layer format is often specific to the individual systems and their implementations, typically varying by vendor. Where data fields are specified or can be otherwise identified, a set of normal behaviors can be observed based on their values. For example, data may be known to have a limited range of values, to exhibit a known distribution, or to have a limited rate at which it can change. In other cases, multiple data fields might exhibit correlations, such as always moving together, or negating one another. Performance outside of these norms could be indicators of a cyber-attack.

DETECTING ANOMALIES

A Cyber Warning Receiver operates by monitoring traffic and discovering anomalies in the behavior of these observations and measurements. The normal set of behaviors for each of the measurements must be characterized before deployment based on the protocol specifications and platform tailored information. Examples of this tailored information could include host addresses in use, message schedule in different operating modes, and observations from collections of real world data.

"Leveraging observable side effects that are agnostic to the specific attack implementation details enables detection of attacks that haven't before been observed in the wild or preconceived by defenders."

In order to detect attacks that have not before been observed in the wild or preconceived by defenders, we must leverage observable side effects that are agnostic to specific attack implementation details. A subtle attack may impact only a small subset of the available observables, within only one of the network layers, while more aggressive attacks may have broader impacts. A robust solution must monitor across all observables and layers.

Anomaly detection at the application layer presents a particular challenge. For example, detecting malicious adjustment of a reported sensor value requires extracting that value, tracking it over time, and comparing it to a normalcy model. Given the variable formats of the application layer, detection of this important attack class requires sophisticated anomaly detectors that:

1. Scale to address the sheer volume of data relationships that would exist for all systems and messages across a complete defense platform.
2. Manage the specifics of the application layer message formats and field locations for dozens of devices and hundreds of unique messages.
3. Discover subtle or secondary correlations that might escape the intuitions of human cyber defense experts and therefore remain open to exploitation by malicious parties.

These limitations suggest the use of more automated techniques for anomaly detector creation.

MACHINE LEARNING AS A KEY ENABLER

Advances in machine learning innately address the three challenges described above. Powerful parameter estimation and model structure detection techniques from machine learning are beneficial for system identification. Multiple examples of using observations to establish normal behavior models for complex systems exist. Activity unexpected by the normal behavior models is thus anomalous and becomes a data point for cyber-attack investigation.

Modern machine learning approaches incorporate feature engineering and credit assignment as key elements. Deep machine learning techniques, for example, combine input observations (e.g., values in each message data field) into more abstract aggregate features that, while no longer representing actual physical measurements, provide an excellent basis for making decisions (i.e., normal behavior or not). Machine learning automatically selects which learned features contribute to making such decisions and which are essentially irrelevant – they assign credit to the various features. These characteristics also obviate the challenge of identifying the most important data fields within the application layer. This is a huge benefit over the alternative of manual specification of data fields and their relative importance.

Machine learning enables reasoning over much larger volumes of data than would be possible for human experts alone. Anomaly detectors increase the visible range of subtle interactions and mutual patterns of behavior exhibited by disparate elements on an embedded network. These patterns may seem innocuous to cyber defense experts trying to envision attack vectors. However, these are exactly the oversights that inevitably get exploited. Finding instances of such subtle relationships has enhanced situational awareness in other domains . Interestingly, insight into such patterns may also prove advantageous in system evaluation and trouble-shooting when non-attack anomalies surface.

TRAINING FOR CONTINUED SUCCESS

With machine learning comes a need for algorithm training, the process by which machine learning algorithms ingest relevant data, extract features, and build their representations of expected behavior. For a practical defense system, this training should not impose intensive requirements for data collection.

Suitable machine learning algorithms operate initially with bus data recorded during field trials and qualification testing and improve their performance upon acquisition of additional data. Once deployed, bus-recordings collected post mission would support incremental updates to training sets and learned behavior models. Distributing new models across platform instances at regular intervals enables all protected platforms to benefit continuously by learning from collective data. This provides a defense system that evolves with new threats and adapts to defeat them.

MEASURING MALICE

Not every anomaly means the platform is under attack. Systems are regularly entering and exiting new states and scenarios and experiencing abnormal conditions resulting from a range of incidental activities or failure modes. The key distinction between system glitches and cyber-attacks are the correlations that exist between observations, and the story they tell.

> *"The key distinction between system glitches and cyber-attacks are the correlations that exist between observations, and the story they tell."*

Any single cyber-attack step would generate a set of measurable side effects and artifacts. Multiple steps in sequence begin to form a picture of the current attacker presence and their objectives in an attack.

A data fusion system is the key element required to put these pieces together. Data fusion formulates the best possible estimate of the underlying system state based on observations, then determines the likelihood that anomalies are caused by an underlying failure, engagement in a scenario or operating mode not previously characterized, or an actual cyber-attack.

BUILDING A COMPLETE PICTURE

A final consideration in defining a Cyber Warning Receiver capability is the question of appropriate response. Among the options are event logging, operator notification, and active defense. Each has its own benefits and drawbacks.

Event logging during anomalous periods provides the capability to perform post-mission forensic data analysis. This low-impact activity is essential in order to provide better threat insights and preparedness for future engagements but offers limited protection for attacks as they are occurring.

Operator notification could help prompt a more immediate response but is not without risk. A notification should never distract a pilot or other key mission personnel unless the findings suggest an imminent survivability threat. Coordinated cyber and kinetic attacks in a combat situation would need to be prioritized to ensure a manageable feed of critical information to the operator. Providing too much information or generating excessive nuisance false alarms might be cause for an operator to disable a system, eliminating the protection and defeating the purpose.

Finally, the possibility of active defense is intriguing as it would allow immediate and automatic response for cyber-attacks, stopping them in their tracks. The risk with any active defense is that it could be tricked by attackers into providing an inappropriate response, in effect becoming a part of the attack itself. Design precautions would be necessary to ensure that attack suppression actions delivered by such an approach could not create consequences beyond what the original attack would have achieved by itself.

CONCLUSION

Modern weapons platforms continue to reach new heights of interconnectivity and software-defined automation. With these enhancements comes the need to address the increasing cyber security risks. Evidence from the commercial and industrial sectors suggests that many of the access vectors and attack methods observed there also apply to

> *"Cyber warning capabilities form a key addition to the suite of platform survivability equipment, providing visibility into the cyber domain and keeping the warfighter safe in the face of this emerging advanced threat."*

DoD platforms, with consequences that are potentially much more severe. Despite this reality, many modern weapons system platforms currently operate without sufficient means of providing detailed situational awareness into their cyber security state.

Embedded network monitoring enables a near-term capability to detect or prevent cyber-attacks that are a very real threat today. Through continuing research, we have characterized a wide range of embedded network-based attacks and established a corresponding set of observables. A Cyber Warning Receiver measures these observables over time and identifies anomalous or malicious activity. In addition to human-defined detection rules, it implements system behavior models derived using machine learning. The use of learned system behaviors enables deep inspection of messages traversing these interfaces to verify they are operating on schedule, that the expected correlations exist between various data fields, and that data ranges and rates of change are within their expected values.

When a cyber-attack occurs, the observations and anomalies that result are collected and examined using a data fusion process. This process estimates the underlying security state of the platform and tracks attacker actions. When critical systems are involved or a survivability risk is identified, a Cyber Warning Receiver can alert operators. Cyber warning capabilities form a key addition to the suite of platform survivability equipment, providing visibility into the cyber domain and keeping the warfighter safe in the face of this emerging advanced threat.

REFERENCES

[1] Gross, Doug. 2013. *Foul-mouthed hacker hijacks baby's monitor*. August 14. Accessed April 25, 2017. http://www.cnn.com/2013/08/14/tech/web/hacked-baby-monitor.

[2] Goodin, Dan. 2017. *Smart TV hack embeds attack code into broadcast signal—no access required*. March 31. Accessed April 25, 2017. https://arstechnica.com/security/2017/03/smart-tv-hack-embeds-attack-code-into-broadcast-signal-no-access-required/.

[3] Miller, Charlie, and Valasek, Chris. 2015. "Remote exploitation of an unaltered passenger vehicle." Black Hat USA 2015.

[4] Falliere, Nicolas, O Murchu, Liam, and Chien, Eric. 2011. W32.*Stuxnet Dossier* Version 1.4. Malware Analysis, Symantec.

[5] Clements, Paul, and John K. Bergey. 2005. *The U.S. Army's Common Avionics Architecture System (CAAS) Product Line: A Case Study*. Technical Report, Pittsburgh, PA: Carnegie Mellon Software Engineering Instutute.

[6] Ljung, Lennart, Hakan Hjalmarsson and Henrik Ohlsson, 2011. Four encounters with system identification. *European Journal of Control*, 5-6, 449-471; Pillonetto, Gianluigi. 2016. The interplay between system identification and machine learning. arXiv:1612.09158v1.

[7] Rhodes, B.J., Bomberger, N.A., Zandipour, M., Garagic, D., Stolzar, L.H., Dankert, J.R., Waxman, A.M., & Seibert, M. (2009). Automated activity pattern learning and monitoring provide decision support to supervisors of busy environments. Intelligent Decision Technologies, 3, 59–74; Rhodes, B.J., Bomberger, N.A., Zandipour, M., Stolzar, L.H., Garagic, D., Dankert, J.R., & Seibert, M. (2009). Anomaly detection & behavior prediction: Higher-level fusion based on computational neuroscientific principles. In N. Milisavljevic (Ed.), Sensor and Data Fusion (pp. 323–336). Croatia: In-Teh.

[8] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Trans. PAMI (Special issue: Learning Deep Architectures), 35, 1798–1828. doi:10.1109/tpami.2013.50; Hinton, G. E.; Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313 (5786), 504–507. doi:10.1126/science.1127647.

[9] Zandipour, M., Rhodes, B.J., & Bomberger, N.A. (2008). Probabilistic prediction of vessel motion at multiple spatial scales for maritime situation awareness. In *Proceedings of the 10th International Conference on Information Fusion*, Cologne, Germany, June 30 – July 3, 2008; Dankert, J.R., Zandipour, M., Pioch, N., Biehl, B., Bussjager, R., Chong, C.Y., Schneider, M., Seibert, M., Zheng, S., & Rhodes, B.J. (2010). MIFFSSA: A multi-INT fusion and discovery approach for Counter-Space Situational Awareness. In Proceedings of 2010 Space Control Conference (SCC), Lexington, MA, USA, May 1–3, 2010.

ABOUT THE AUTHORS

PATRICK M. HAYDEN is a Technology Development Manager for Platform Cyber Resilience programs at BAE Systems. Patrick has 10 years of experience in the cyber security field, performing systems assessments, software reverse engineering, and vulnerability research covering both offensive and defensive perspectives across a wide range of targets and applications.

DAVID K. WOOLRICH is a program manager in BAE Systems' Survivability, Targeting, and Sensing Solutions business area. His background is information security, information assurance, and cyber electronic warfare. He is focused on increasing awareness and survivability of DoD assets from cyber-attacks.

KATIE D. SOBOLEWSKI is a Technology Development Manager for Cyber Electronic Protection at BAE Systems with experience in cyber defense, cyber electronic warfare, and platform protection. Katie has a background in algorithm development, signal processing, and optimization for increased system performance.

CSIAC Cyber Security & Information Systems Information Analysis Center

Need Specialized Technical Support with Easy Contract Terms?

Core Analysis Task (CAT) Program

A Pre-Awarded, Pre-Competed Contract Vehicle.

CSIAC provides Subject Matter Expert (SME) support on an as-needed basis to quickly address technical requirements with minimal contracting effort. CSIAC provides such solutions via the utilization of our Core Analysis Task (CAT) service/capability. CSIAC is a competitively awarded contract with Indefinite Delivery/Indefinite Quantity (ID/IQ) provisions that allow us to rapidly respond to our users' most important needs and requirements. Custom solutions are delivered by executing user-defined and funded CAT projects without the need for further competition.

Through the CAT program, CSIAC is a pre-competed contracting vehicle, enabling the DoD and other agencies to obtain technical support for specific projects/programs that fall within one of the CSIAC technology areas. As with any inquiry, the first four hours are free. If the scope requires a CAT, CSIAC will assist with the development of a Performance of Work Statement (PWS) to be approved by the Contracting Officer's Representative (COR).

Key Advantages of working with CSIAC:

Expansive Technical Domain

The CSIAC's broad technical scope provides numerous pre-qualified resources for potential projects, and is especially valuable for today's information system challenges that frequently cross multiple domains.

Comprehensive STI Repositories

As a consolidation of three predecessor Information Analysis Centers (IACs), CSIAC has a wealth of expertise, data and information to support the successful completion of CATs.

Expansive Subject Matter Expert Network

CSIAC is able to leverage reach-back support from its expansive SME Network, including technical experts from the CSIAC staff, team members, or the greater community, to complete CATs.

Minimal Start-Work Delay

Not only does CSIAC provide DoD and other government agencies with a contract vehicle, but as a pre-competed single award CPFF IDIQ, work can begin in just a matter of weeks.

Apply the Latest Research Findings

CSIAC draws from the most recent studies performed by agencies across the DoD, leveraging the STI holdings of the Defense Technical Information Center (DTIC). The results of all CSIAC CATs and other DoD-funded efforts are collected and stored in DTIC's STI repository to support future efforts by the CSIAC and others.

How To Get Started

If you have a need for CSIAC technical support, the first step is to contact us. All Technical Inquiries are free to the customer for up to four hours of service. If the scope of the support is more extensive and requires a CAT, CSIAC will assist with the development and submission of the task description and related contract documents. CATs may be awarded as either Cost Plus Fixed Fee (CPFF) or Firm Fixed Price (FFP) delivery orders.

Inquiries may be submitted by email to **info@csiac.org**,
or by phone at **1-800-214-7921**.

Please visit our website for more information:
https://www.csiac.org/services/core-analysis-task-cat-program/

Who We Are

The Cyber Security Information Systems Information Analysis Center (CSIAC) is the DoD's Center of Excellence in Cyber Security and Information Systems, covering the following technical domains:

- Cybersecurity
- Software Engineering
- Modeling and Simulation
- Knowledge Management/ Information Sharing

CSIAC is chartered to leverage best practices and expertise from government, industry, and academia to solve the most challenging scientific and technical problems. The Center specializes in the collection, analysis, synthesis, and dissemination of Scientific and Technical Information (STI) to produce solutions in support of the defense community.

Our Team

Quanterion Solutions Incorporated is the prime contractor responsible for operating the CSIAC. In addition to Quanterion, customers also have access to the other members of the CSIAC team which include leading technology corporations as well as prestigious academic institutions that perform cutting edge research activities to expand our knowledge base.

CSIAC
Cyber Security & Information Systems
Information Analysis Center

266 Genesee Street
Utica, NY 13502

1-800-214-7921
https://www.csiac.org

Cyber Security and Information Systems
Information Analysis Center
266 Genesee Street
Utica, NY 13502

PRSRT STD
U.S. Postage
P A I D
Permit #566
UTICA, NY

THE CENTER OF EXCELLENCE IN CYBER SECURITY AND INFORMATION SYSTEMS

Leveraging the best practices and expertise from government, industry, and academia in order to solve your scientific and technical problems

Design and Development Process for Assured Software - Volume 1

Journal of Cyber Security & Information Systems

https://www.csiac.org/journal/

*To unsubscribe from CSIAC Journal Mailings please email us at **info@csiac.org** and request that your address be removed from our distribution mailing database.*