VOLUME 4 NUMBER 1
SPECIAL EDITION

JOURNAL OF CYBER SECURITY & INFORMATION SYSTEMS

FOCUS ON AIR FORCE RESEARCH LABORATORY'S INFORMATION DIRECTORATE

CSIAC

Introduction

By Juanita L. Riley, AFRL/RIGA

In this issue of the Journal of Cyber Security and Information Systems features the contributions of the scientists and engineers from the Air Force Research Lab Information Directorate in Rome, NY. The Information Directorate is focused on Information Technology which holds the key for the future of battlespace command and control, situation awareness of who the enemy is, real-time knowledge of what is happening, and exploiting techniques to rapidly transfer critical information to the decision makers. Information superiority will allow warfighters to dominate and control battlespace – control that is essential to virtually all joint warfighting capabilities in the 21st Century [1].

The Information Directorate expands its workforce knowledge base through partnerships with academia, industry, and representing on scientific national and international working groups. In addition to expanding the knowledge base of its workforce, the Information Directorate is committed to growing future leaders in Science, Technology, Engineering, and Mathematics (STEM) by participating in local community STEM activities and providing summer internship opportunities, such as the Advanced Course in Engineering (ACE) summer internship that develops the next generation of cyber-security leaders, with a particular emphasis on educating future military leaders. The Information Directorate aims to lead the Air Force and the Nation in Command, Control, Communications, Computers, and Intelligence (C4I) and Cyber Science, technology, research, and development [2].

The Information Directorate is a recognized leader in C4I and Cyber. In order to address the most critical C4I and Cyber needs of the Air Force, the Information Directorate has organized their science and technology portfolios into four Core Technical Competencies (CTCs) [2]:

(1) Autonomy, Command and Control, and Decision Support
(2) Processing and Exploitation
(3) Cyber Science and Technology
(4) Connectivity and Dissemination

Autonomy, Command and Control, and Decision Support

The Autonomy, Command and Control (C2), and Decision Support CTC is inventing technologies to realize truly integrated, resilient, and robust command and control systems. The mission is to deliver innovative trusted, affordable information technologies for agile, resilient, and distributed Air Force command and control systems [2].

Processing and Exploitation

The Processing and Exploitation CTC provides the computing and algorithms behind transforming big data into information. The mission is to lead the discovery, development, and transition of all-source processing and exploitation innovations for the Air Force and Joint communications [2].

Cyber Science and Technology

The Cyber Science and Technology CTC is leveraging and shaping the cyber domain to the nation's advantage. The mission is to design, develop and transition innovative cyber capabilities to the Air Force and Joint communities [2].

Connectivity and Dissemination

The Connectivity and Dissemination CTC is putting the right information into the right hands at the right time. The mission is to provide agile and secure mission-responsive communications and information sharing globally [2].

These CTCs have provided advanced research and transitioned technologies that are equipped with the capabilities to meet the operational needs of the Air Force and other military organizations. This issue will provide articles that explain the research and technology development that is occurring under each of these CTCs. In addition to, there is an article about the success and implementation the Information Directorate uses to develop junior employees to continue to push the envelope to lead, discover, develop, and deliver cutting edge research and technology to the 21st century warfighter.

References

[1] Air Force Research Laboratory Information Directorate, *Information Directorate Technical Brochure*. Rome, NY: Strategic Planning and Integration Division, 2012.

[2] Air Force Research Laboratory Information Directorate, A*nnual Review with an Economic Impact Analysis*. Rome, NY: Strategic Planning and Integration Division, 2014.

RELEASE STATEMENT

Approved for Public Release; Distribution Unlimited:
[88ABW-2015-5687] 19 Nov 15

Air Force Research Laboratory / Information Directorate (Rome NY)

By Charles Messenger, Quanterion Solutions Incorporated

AFRL's Information Directorate is located in Rome NY. Rome Air Development Center (RADC), the predecessor organization to AFRL Rome Research Site, began operations at Griffiss on June 12, 1951. RADC was the Air Force's research and development of ground electronics and intelligence systems. In 1990, RADC became Rome Laboratory as part of an Air Force Laboratory consolidation. In 1995, the Base Realignment and Closure Commission (BRACC) closed Griffiss Air Force Base but maintained Rome Laboratory as a "stand alone" facility. In 1997, the Air Force consolidated its laboratories into Air Force Research Laboratory and established the AFRL Rome Research Site. Rome Research Site draws on a 60 year tradition of excellence researching and developing revolutionary technologies such as troposcatter and satellite communications, long-distance radios, phased array radars, computer networks and software, electronic reliability tests and standards. RADC was one of the original 21 nodes of the ARPANET, the pioneering computer network that we know today as the internet. These advances became beneficial not only to the Nation's military, but its citizens' everyday lives as well. The transistor, the integrated circuit, the personal computer, the laser and the compact disc all advanced from the research at AFRL Rome Research Site.

Today, AFRL's Information Directorate is focused on Information Technology which holds the key for the future of battlespace command and control. Situation awareness of who the enemy is, real-time knowledge of what is happening, and exploiting techniques to rapidly transfer critical information to the decision makers are all crucial. Information superiority will allow warfighters to dominate and control battlespace – control that is essential to virtually all joint warfighting capabilities in the 21st Century.

Command, Control, Communications, Cyber, and Intelligence (C4I) is the key enabler of the Air Force's ability to conduct its mission to fly, fight, and win in air, space, and cyberspace. Air Force Laboratory/ Information Directorate's ability to conceive, develop, and transition compelling C4I capabilities provides the science and technology backbone to support the AF vision of Global Vigilance, Reach and Power for our Nation.

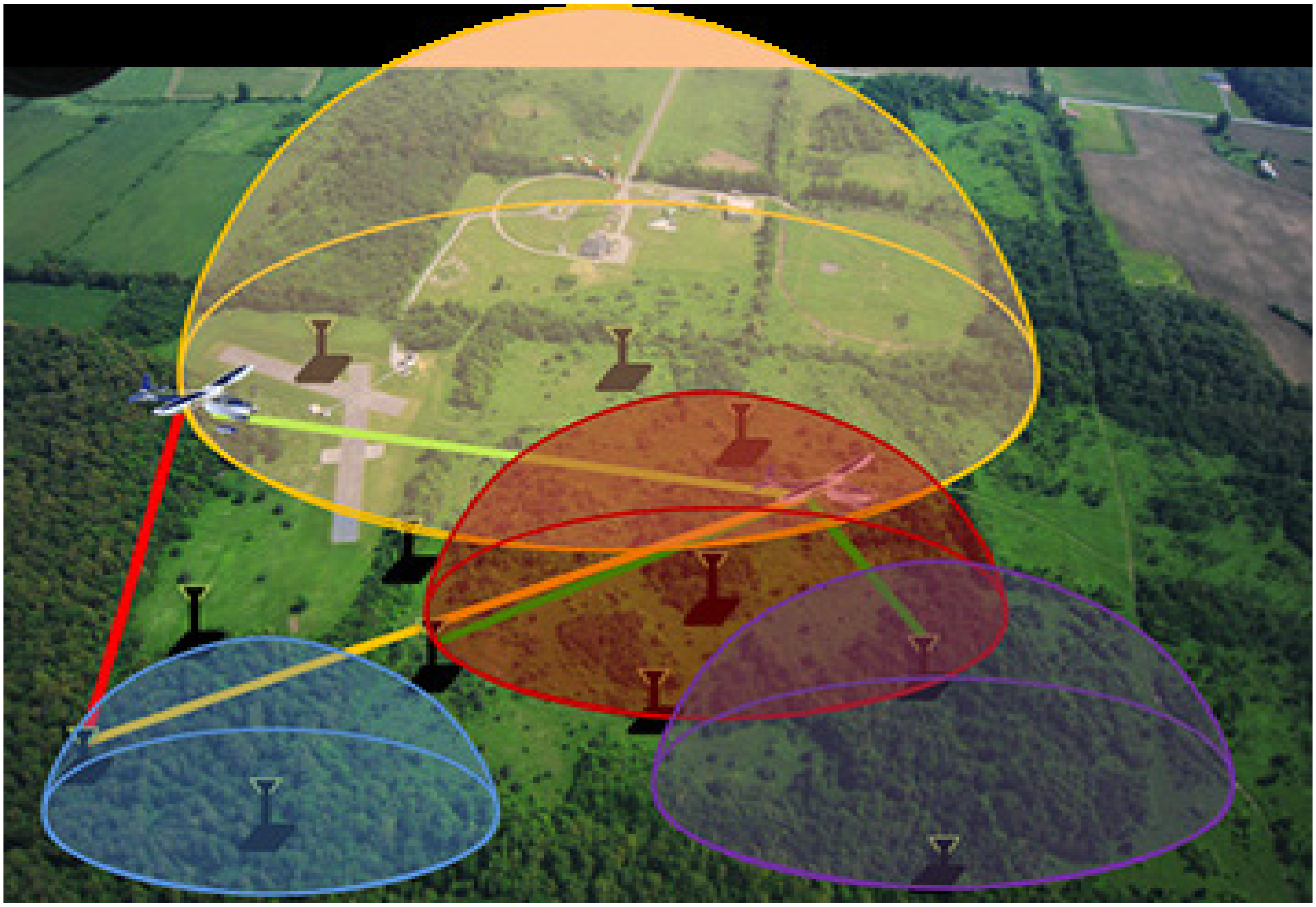
To achieve its mission, the Information Directorate focuses its research and development in four Core Technical Competency (CTC) areas; Autonomy, Command and Control (C2), and Decision Support, Processing and Exploitation, Connectivity and Dissemination, and Cyber Science and Technology.

The Autonomy, Command and Control (C2), and Decision Support CTC delivers distributed, resilient, timely, integrated C2 decision making technologies for the monitor, assess, plan and execute processes associated with Air Force command, control, and intelligence operations. Focus is on technology to present actionable information to military decision-makers and anticipate future adversarial and indigenous population activity. Being able to synchronize actions across air, space, and cyberspace and deliver agile C2 capabilities for future dynamic conflicts is critical. Research is focused on building trusted highly autonomous systems to enable machine-aided decision support.

The Processing and Exploitation CTC leads the discovery, development, and transition of all-source processing and exploitation innovations for the Air Force and joint communities. Focus is on creating advanced techniques, architectures and prototypes to intercept, collect, and process sensor and intelligence data; the computing and algorithms behind transforming raw data into information. Challenges center on managing, processing, and exploiting current massive amounts of ISR data flows to analyze Patterns of Life and infer relationships and assessment of the current situation.

The Connectivity and Dissemination CTC provides assured, mission-responsive communications and secure information exchange for the Air Force and joint communities; putting the right information into the right hands at the right time. A key challenge is developing layered communications and mission-aware networks, from platforms to capabilities, in congested and contested RF environment. Research involves cross-domain multimedia information sharing and mission-aware prioritized resource management.

The Cyber Science and Technology CTC creates the future Air Force and joint service assured operating environments that provide for mission aware and resilient full spectrum capabilities; leveraging and shaping the cyber domain to US advantage. The challenge is providing Mission Assurance while moving from cyber defense to resilience and developing trusted computing regardless of supply chain. Research is focused on mission modeling and cyber situational awareness for assuring effective missions,



cyber agility to disrupt/deny adversary attack planning, cyber resiliency to fight through and recover from attack, hardware & software "Root of Trust" for computational platform assurance, and full spectrum cyber operations for Cyberspace Superiority.

AFRL also has two very unique test ranges; the Stockbridge and Newport facilities. The Stockbridge Facility is used for development and evaluation of advanced RF/optical communications systems, radar imaging systems, foliage penetration studies and for communications link experiments with small unmanned aircraft systems. The facility provides a controllable RF interference environment for time varying analysis and evaluation of communications systems. A Small Unmanned Aerial System (SUAS) airfield is also operational within the facility.

The Newport Facility is comprised of five independent data acquisition facilities and eight measurement ranges. All ranges and both hills are interconnected with a fiber optic network with an interface to instrumentation and a high data rate link to AFRL Rome Research Site. The five primary ranges are fully instrumented with signal sources, antennas, amplifiers, receivers, computers, displays, recording systems, fiber optic interfaces, positioned controllers and high speed multiplex systems. Simultaneous operation of four ranges is possible. Automated data acquisition allows data to be available in real-time for analysis and recorded digitally for future off-line analysis. The facility is used primarily to obtain antenna patterns and to perform isolation measurements on full size tactical aircraft such as the F-35, F-22, A-10, F-15, F-16, various helicopters (Blackhawk/Seahawk), remotely piloted aircraft (RPAs) sections of the B-1B, KC-135, C-130, and future aircraft prototypes. Other types of systems such as ground vehicles, specialized aircraft, and satellites are also evaluated in accordance with the needs of their specific programs.

In this issue of the Cyber Security and Information Systems Information Analysis Center (CSIAC) Journal we present several articles on the technologies and capabilities being developed at AFRL Rome Research Site. ✈

Command and Control of Proactive Defense

By David Last, David Myers, Matthew Heffernan, Meghan Caiazzo, and Capt. Nicholas Paltzer

Missions are under constant threat of cyber-attacks that can cause the denial of critical services and the loss of data confidentiality. The application of proactive cyber defenses can help prevent these attacks, but may also endanger the mission by exhausting system resources when the defenses are not optimally implemented. The potential for cyber friendly-fire increases when adding moving-target defenses (MTDs) to the defensive posture of the mission system. The Command and Control of Proactive Defense (C2PD) program provides a capability to balance cyber security with mission assurance by generating the optimal defensive posture for a cyber security administrator (CSA) to deploy based on metrics of the mission system's attack surface, mission requirements, and the combination of proactive cyber defenses.

Introduction

In today's cyber environment, attackers have an asymmetrical advantage over cyber defenders. This advantage comes from the idea that perfect security does not exist without hindering the system's usability. Cyber defenders must lock down every entry point and attempt to account for undiscovered vulnerabilities, while an adversary must only find one way to breach the attack surface, which is the attacker's view of a system. With current defense deployment, an adversary sitting on a host or network has virtually unlimited time to perform reconnaissance and plan attacks. The adversary's unequivocal advantage makes the cyber defender's task of deploying and configuring defenses quite challenging.

A CSA ideally wants their system to appear to be nondeterministic to an attacker, however this conflicts with static defense approaches. The new defense classification of MTDs changes the attack surface over time, which makes the system less predictable. MTDs create command and control (C2) challenges for a CSA. The deployment of any cyber defense consumes resources needed for mission execution. A CSA must maintain mission assurance while providing cyber security. Although a CSA is concerned with system security, they are equally as concerned about mission assurance, which is dependent on a predictable system. A CSA must balance system security with system

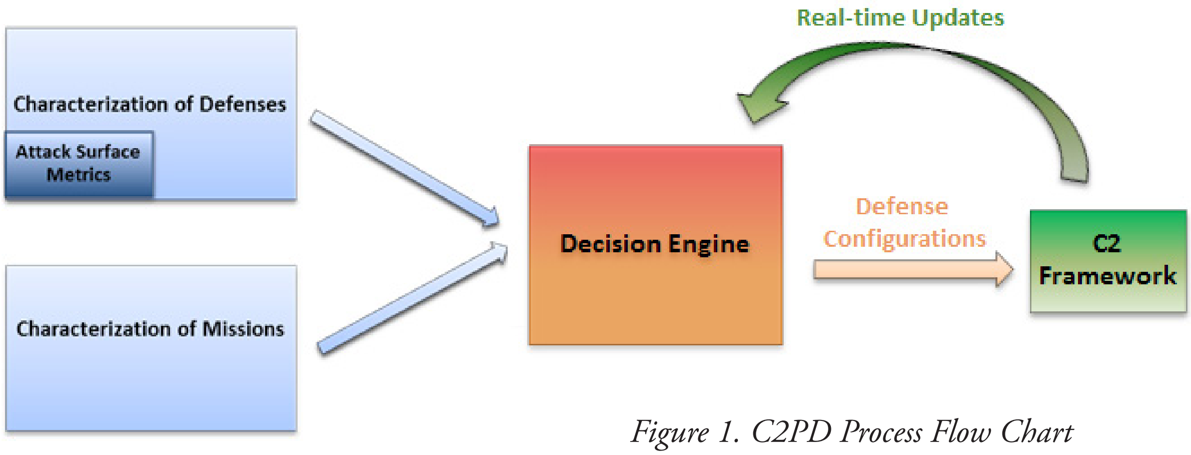


Figure 1. C2PD Process Flow Chart

resource consumption, mission execution, and defense interoperability. This information overload makes it difficult for a CSA to make an intuitive decision about deploying available cyber defenses.

Command and Control of Proactive Defense (C2PD) is an Air Force Research Laboratory program to provide a decision-support capability for automated deployment of MTDs and other proactive defenses. C2PD determines the optimal defense configuration based on metrics of attack surface, defense characterization, and mission requirements. It generates metrics for different defensive postures and determines the optimal configuration to present to a CSA, which is automatically deployed to the system via an integrated C2 framework upon selection. Figure 1. C2PD Process Flow Chart shows the inputs and process for C2PD's defense determination.

The objective of this research is to produce an automated procedure for producing defensive configurations that allows a CSA to maximize non-deterministic system appearance from the perspective of an attacker while maintaining deterministic system behavior for mission assurance. A product of this research is the ability to show that this automated process of generating and implementing a defensive posture significantly increases the difficulty of any attack against a mission system compared to a manual version of this process. Additionally, the speed of the automated course of action (COA) generation outperforms an intuitively designed manual configuration. Most importantly, the resulting defensive posture provides both mission and information assurance through the provision of a deterministic quality of service while using MTDs. The remainder of the paper is organized as follows. Section 2 provides background on MTD research. Section 3 defines the attack surface and security metrics relating to overall system security and resource consumption. Section 4 details the generation of defense configurations. Section 5 describes how a modular framework unifies communications across a system and deploys cyber defenses. This paper does not discuss mission characterization; however, it is required to generate defense configurations and prevent cyber friendly fire.

Background

Recognizing the attacker's advantage gained with unlimited reconnaissance time, the cyber security research community has responded with the development of MTDs to mitigate this advantage. MTDs provide security by shifting the target system's attack surface over time. With the target system's attack surface changing over time, the adversary cannot rely on information gained from previous reconnaissance efforts.

A foundational survey of MTDs by Lincoln Laboratories categorizes these defenses by the system resources they manipulate [1]. Table 1. Description of the five MTD categories in regard to the modification defense type, what type of attack they were designed to defend against and the associated overhead for the general case [1]. shows each MTD category and its associated security benefits and resource impact. This information influences the development of defense configurations.

Table 1. Description of the five MTD categories in regard to the modification defense type, what type of attack they were designed to defend against and the associated overhead for the general case [1].

MTD Category	Attack Thwarted	Modification Targets	Resources Impacted
Dynamic Runtime Environment	Injection	Memory Layout, Interfaces presenting processor and system calls	Execution, Memory
Dynamic Software	Injection, Exploitation of Trust/Privilege	Program Instructions, including format, grouping, and order	Execution, Hardware
Dynamic Platforms	Injection, Exploitation of Trust/Privilege, Scanning, Resource, Supply Chain	Operating System Version, Build Instance, CPU Architecture	Execution, Memory, Network, Hardware
Dynamic Data	Injection, Resource, Exploitation of Authentication	Format, Encoding, Syntax, Representation	Execution, Memory
Dynamic Networks	Exploitation of Trust/Privilege, Scanning, Resource, Spoofing, Data Leakage	Protocols, Addresses	Network

MTDs create new C2 challenges for mission assurance. Mission execution depends on deterministic system behavior, while MTDs create a non-deterministic attack surface. A CSA responsible for the security of a mission system currently does not have quantitative information about the effects of a defensive posture, mission resource requirements, or system vulnerabilities. Therefore, a CSA depends on intuition to develop a defensive posture COA to provide mission assurance. MTDs have the potential for providing enhanced cyber security. However, ad hoc defense deployments are as likely to create an internal denial of service as they are to prevent an external one. This inherent risk requires that various cyber defenses are quantified and characterized prior to deployment.

Metrics

There are many factors to consider when generating a cyber defense configuration. One of the most important factors is a measure of security or resistance to attack. Researchers have tried to develop a generalized method for measuring the security of an information system; Manadhata and Wing developed one of the most comprehensive approaches and codified it in terms of a measurement of the attack surface of the system [2] [3]. In their approach, an attack surface metric for an information system is based on an enumeration of all possible entry and exit points into the system, with each point weighted according to the ease of penetration and the consequences (to the defender) of penetration. This paper leverages this definition of an attack surface.

This attack surface measurement is generated by reasoning over models of a system. Models of the network, available defenses, and information flows that are part of the cyber mission are composed to represent the defender's area of responsibility. Models of the adversary capabilities and available attack vectors in the system represent threats to system security. The attack vector model represents all possible adversary actions; they are combined to generate an attack graph that describes the system's vulnerabilities. Different cyber defenses, including MTDs, disrupt different attack steps in the attack graph, reducing the number of attack paths available to the adversary to reach his goal. This attack surface measurement capability is used to reason over these models to characterize different defense configurations.

One of the limitations of building an attack surface metric as described above is the challenge of enumerating all possible attack steps available to an attacker. Attack step models must be based on known software vulnerabilities; however, vulnerabilities discovered in the future will result in new attack steps or change the attacker cost or defender consequences of an existing attack step. Any new attack step changes the attack surface measurement. Therefore, the attack step model must also account for zero-day attacks enabled by previously undiscovered vulnerabilities.

This research also addresses the forecasts for discovering the number, type, and severity of zero-day vulnerabilities. This work leverages previous research on Software Vulnerability Discovery Models [4] [5] to generate zero-day forecasts; Last details the current state of this research [6].

In order to ensure the validity of the attack surface measurements, the defense models must accurately describe the performance and behavior of defenses in an active system. Characterization profiles of these defenses include an analysis of the security they provide, measurement of their impact on system resources, and their potential interoperability issues with other defenses. This process generates characterization profiles for all defenses available to a CSA. These characterization profiles, along with mission information, aid in the generation of defense configurations.

Defense Configuration Development

At the basis of this C2 problem is a decision made by a CSA. A CSA must decide where to utilize available defenses. This defense configuration development problem is a multi-criteria decision making problem.

A CSA must balance network defense priorities with mission priorities. This requires a full understanding of available defense capabilities. The defense characterization process described in Section provides this vital information. A CSA must also understand mission priorities, critical network assets, and services. Understanding these three components allows the C2PD program to develop a decision-support tool for assigning, deploying, and orchestrating multiple defenses simultaneously.

This decision-support for assigning defense techniques is inherently an optimization problem. Multi-criteria optimization allows for the balancing of multiple objectives, maximizing the defense provided to network assets, and minimizing resource consumption that affects mission priorities. A decision engine utilizing this technique provides several mathematically optimal defense configurations to a CSA. Multi-criteria decision-making techniques also allow for CSA interaction with the optimization process in order to allow for human-in-the-loop decision-support.

After a CSA has selected the most preferred defense configuration for implementation, the defense deployment framework described in Section provides the ability to deploy these defenses onto the network. This allows for an initial human-in-the-loop decision and enables future autonomous behavior where the decision engine could suggest changes to defense configurations based on observations of network activity or changes in mission priority.

Framework

Current manual methods for installation, configuration, and activation of cyber defenses are labor-intensive. This slow process does not allow a CSA to manipulate defenses more quickly than an attacker can adapt. C2PD provides an automated method for deployment of defense configurations, drastically shortening a CSA's response time.

CSAs protect the network by utilizing a diverse collection of sensors, defenses, and other assets installed throughout the network. C2PD provides a common communication framework to integrate all of these tools into a C2 system. The use of botnets is an example of this concept of centralized control and decentralized execution. Botnets connect a diverse array of computing assets by standardizing communication and enabling distributed C2.

The C2PD framework provides common communication and distributed C2 for cyber defenses. This framework incorporates sensors, cyber defenses, and other cyber assets into modules. A CSA adds or removes modules to the framework based on mission requirements. Distributed control of modular defenses for rapid deployment provides a scalable defensive posture.

This framework is a distributed multi-agent system [7]. All defenses, defense assets, and C2 interfaces associate with their own agent in the framework through a common Application Program Interface (API). Defenses and defense assets report to the C2 interface via communication by their respective agents. This method of control conceals the implementation details of cyber defenses from the controller.

The framework has a set of core services. First, the registration service provides naming and location services for agents within the network. Second, a message service is required to allow communication between agents. A message encryption service provides encryption for messages transmitted within the network over any message transport service. Additionally, an audit service is available for system integrity. This auditing service records all events with timestamps within the framework. A logging service outputs a record of events to administrators. This logging service can correlate logs from multiple hosts across the network and present a single log to a CSA. The publish-and-subscribe service enables the specification for types of input and output for agents or environments. A

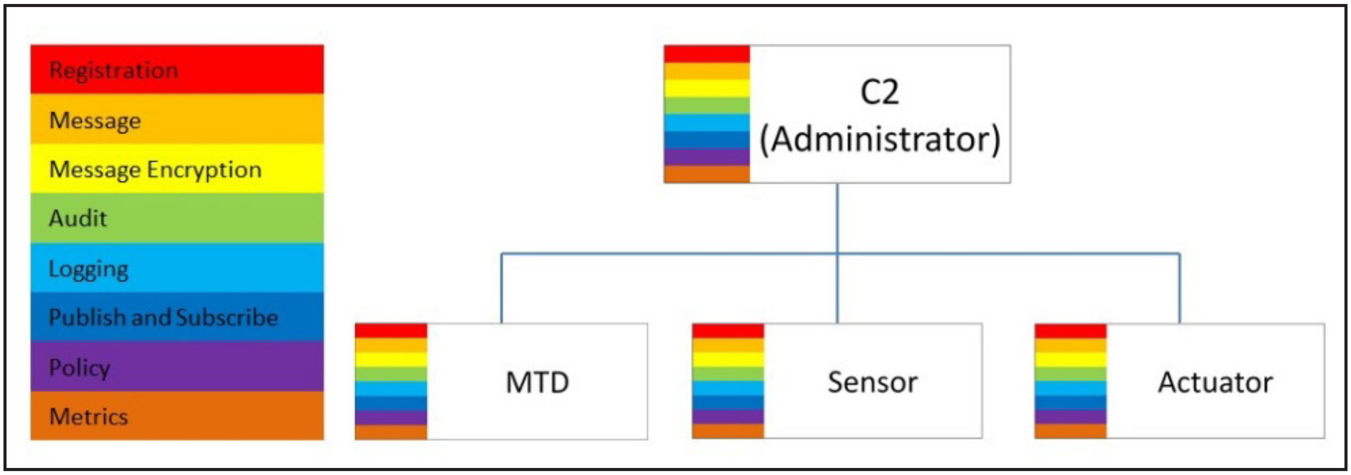


Figure 2. Core services within API framework

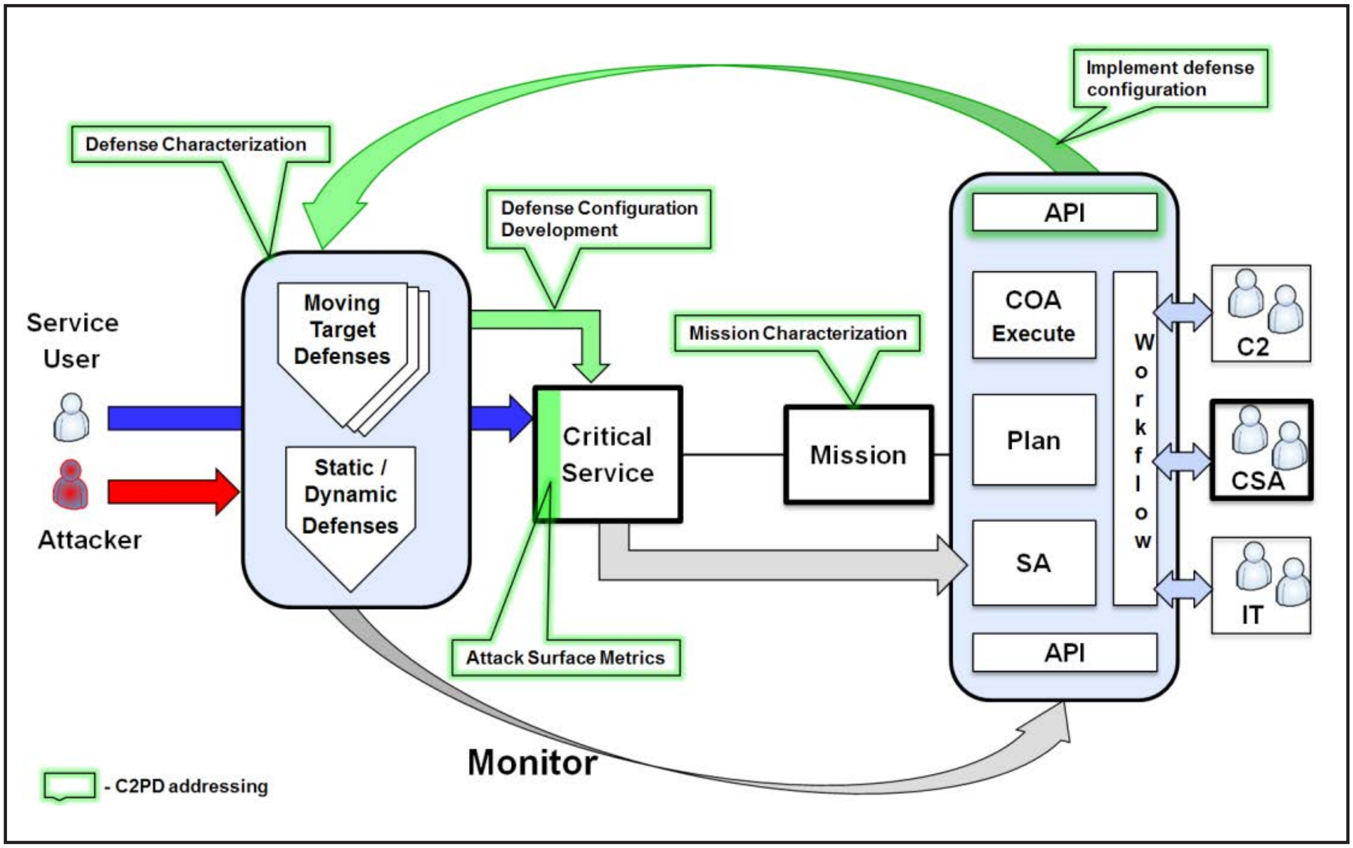


Figure 3. C2PD Technical Scope

policy service allows for granular control of the various services running on the framework. Finally, a metrics service reports performance and resource usage statistics from across the network hosts and agents. Figure 2. Core services within API framework represents the use of these services within the framework by each agent.

Conclusion

In the current state of the art of network defense, a CSA must overcome the attacker's asymmetric advantage. Proactive application of defenses puts the attacker and defender on equal footing. In order to generate effective defense plans, it is vital to characterize available defenses. Configurations generated based on these characterizations maximize security while minimizing impact on mission-critical resources. The C2PD program, as illustrated in Figure 3. C2PD Technical Scope, generates these configurations and provides them to a CSA for human-in-the-loop decision making. The selected defense configuration is automatically deployed on the network via the C2PD framework. C2PD advances the state of the art of network defense by greatly decreasing the time required to develop a defensive posture as well as increasing the effectiveness of these postures.

Bibliography

[1] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard and W. Streilein, "Survey of Cyber Moving Target Techniques," Massachusetts Institute of Technology Linconln Laboratory, 2013.

[2] P. K. Manadhata and J. M. Wing, "An attack surface metric," *Software Engineering, IEEE Transactions* on, vol. 37, no. 3, pp. 371-386, 2011.

[3] P. Manadhata and J. M. Wing, "Measuring a system's attack surface," DTIC Document, 2004.

[4] O. H. Alhazmi and Y. K. Malaiya, "Prediction capabilities of vulnerability discovery models.," in *Reliability and Maintainability Symposium*, 2006. RAMS'06. Annual, IEEE, 2006, pp. 86-91.

[5] J. Kim, Y. Malaiya and I. Ray, "Vulnerability discovery in multi-version software systems," in *High Assurance Systems*

Engineering Symposium, 2007. HASE'07. 10th IEEE, IEEE, 2007, pp. 141-148.

[6] D. Last, "Using Historical Software Vulnerability Data to Forecast Future Vulnerabilities," in *Resilience Week 2015, Proceedings of* , Philadelphia, 2015.

[7] M. Carvahlo, T. C. Eskridge, K. Ferguson-Walter, N. Paltzer, D. Myers and D. Last, "MIRA: A Support Infrastructure for Cyber Command and Control Operations," in *Resilience Week 2015, Proceedings of*, Philadelphia, 2015.

About the Authors

David Last earned his Bachelor's and Doctorate degrees in Electrical and Computer Engineering from Auburn University in Auburn, Alabama. His research interests include computer and network security and resiliency and moving target defenses.

David Myers received his B.S., M.S., and Ph.D. in Industrial and Systems Engineering from the University at Buffalo, The State University of New York. His research interests are in the application of multi-criteria decision making for military and government domains.

Matthew Heffernan received his B.S. in Computer Engineering from Rochester Institute of Technology and is currently pursuing an M.S. in Computer Science from Syracuse University. His most recent research has been in data visualization and command and control systems.

Meghan Caiazzo received her B.S. in Computer Science and Mathematics from St. Joseph's College and her M.S. in Cyber Security from New York University Polytechnic School of Engineering.

Captain Nicholas Paltzer (U.S. Air Force) has been a member of the United States Air Force for more than 20 years and has an extensive background in telecommunications and networking in both fixed base and deployed environments. He received his B.S. from the University of South Florida and his M.S from the Air Force Institute of Technology, both in Computer Engineering.

RELEASE STATEMENT

Distribution A: Approved for public release; distribution is unlimited. Case number 88ABW-2015-4680.

SOAR
Security Risk Management for the Off-the-Shelf (OTS) Information and Communications Technology (ICT) Supply Chain
CSIAC

Cyber Deception

By Dave Climek, Anthony Macera, and Walt Tirenin

The Department of Defense currently depends upon static cyber defense systems. Adversaries can plan their attacks carefully over time by relying on the static nature of our networks, and launch their attacks at the times and places of their choosing. The DoD needs new tools and technologies to reverse the current asymmetry that favors our cyber adversaries, by forcing them to spend more time and resources, cope with greater levels of complexity and uncertainty, and accept greater risks of exposure and detection due to the significantly increased requirements for reconnaissance and intelligence collection of our networks. Throughout history the military has employed deception as a counter-intelligence mechanism, but thus far it has been minimally employed for tactics and strategies in cyberspace to counter cyber exploitation and attack. The best known attempts at cyber deception in the commercial realm are honeypots and honeynets. These passive decoy technologies rely on effective intrusion detection, and if implemented inappropriately, can be easily detected and avoided by attackers. Modern day military planners need a capability that goes beyond the current state-of-the-art in cyber deception to provide a system or systems that can be employed for defensive purposes by a commander when needed, to enable proactive deception to be inserted into cyber operations.

Significance

Cyber deception is a deliberate and controlled act to conceal our networks, create uncertainty and confusion against the adversary's efforts to establish situational awareness, and to influence and misdirect adversary perceptions and decision processes. Defense through deception can potentially level the cyber battlefield by altering an enemy's perception of reality through delays and disinformation which can reveal attack methods and provide the attributions needed to identify the adversary's strategy. Delaying and dissuading also provides the essential time for forensics teams to analyze, identify, and mitigate attack vectors that could expose inherent vulnerabilities to operational and support systems.

Background

Military deception is defined as "actions executed to deliberately mislead adversary military, paramilitary, or violent extremist organization (VEO) decision makers, thereby causing the adversary to take specific actions (or inactions) that will contribute to the accomplishment of the friendly mission." [1]. Military forces have used techniques such as camouflage, feints, chaff, jammers, fake equipment, false messages or traffic, etc. for thousands of years to alter an enemy's perception of reality.

Whether intended for exfiltrating data or destroying systems, an attacker generally follows a typical sequence of steps: reconnaissance, weaponization, delivery, exploitation, control, execute mission, and maintaining access [2]. Attacks generally target configurations, interfaces, and applications that are exposed at the host and network levels.

It is believed that deception techniques, as part of an overall moving target defense and in conjunction with other normal cyber defense methods, can alter the underlying attack process. They can create uncertainty to delay and disrupt the attacker's ability to determine the status, location, or implementation details of the configurations, interfaces, and applications they are trying to target. This will make the attack attempt much more difficult, time consuming, risky, and cost prohibitive.

Much work has already been done in cyber deception technologies. Honeypots are computers designed to attract attackers by impersonating another machine that may be worthy of being attacked. Honeynets take that further by simulating a number of computers or a network, and products such as the Deception Toolkit [5] convey an impression of the defenses of a computer system that are different from what they really are by creating phony vulnerabilities. Honeytokens are false data implanted within systems to confuse an attacker or to serve as a security trigger when they are detected as being exfiltrated.

Low-Interaction honeypots utilize emulated or virtualized software that is usually inexpensive and easy to set up, but often are unable to provide the full functionality of a real computer system. The use of computer virtualization allows a single host computer to simultaneously run a number of resident virtual machines each with identifiable features as unique as actual systems, including different operating systems, file systems, network settings, and some hardware. More complex honeypots, known as High-Interaction honeypots, can provide the complete set of functionality found on a normal system, but typically require the use of more hardware and can be complicated to set up. The current state of the art in deception technology implements and assumes a static configuration. While this static configuration is helpful for administration and management, it is also “helpful” to the attacker [3]. The static nature of networks allows an attacker to employ various means of information collection that can be done slowly enough, and across a long enough time period, to hide these reconnaissance activities in the “noise” of normal day-to-day operations. The application of honeypots can help protect a network by providing false information to distract an attacker, and cause time and effort to be wasted during the course of an attack. However, an attacker can also utilize a compromised honeypot system to carry out other attacks on neighboring systems or to participate in a distributed attack.

Advanced techniques are needed with a focus on introducing varying deception dynamics in network protocols and services which can severely impede, confound, and degrade an attacker’s methods of exploitation and attack, thereby increasing the costs and limiting the benefits gained from the attack. Forcing changes in the attacker’s behavior or actions can also serve to highlight and expose his activities for enhanced detection, deriving intent, as well as improved forensics and remediation for actions already taken. The combination of these effects can form a strong basis for deterrence. [4]

The DoD operates within a highly standardized environment. Any technology that significantly disrupts or increases the cost to the standard of practice will unlikely be adopted. If the technology is adopted, the defense system must appear legitimate to the adversary trying to exploit it.

AFRL/RIGA Approach

AFRL/RIGA initiated a deception effort under the Cyber Agility program in FY15.

This exploratory effort will have an emphasis on technologies that delay the attacker in the reconnaissance through weaponization stages of an attack, and also aid defenses by forcing an attacker to move and act in a more observable manner. This technology seeks to provide deception in our systems and networks at multiple levels and in multiple forms, recognizing that attackers target our cyber infrastructure across the various protocol and system layers. Techniques across the host and network layers or a hybrid thereof, will be explored in order to provide AF cyber operations with effective, flexible, and rapid deployment options.

Network-based deception approaches may focus on manipulating network activities to mask, fabricate, or simulate authentic operational networks. For example, they may generate displays or ruses in terms of fake “mirage” networks, or attribute characteristics to real networks that mislead the attacker about their structure, critical nodes, etc. These techniques may be particularly effective for deceiving attackers during the reconnaissance stage of the attacker model. Host-based approaches can be utilized to isolate critical resources while exposing falsified resources to an adversary as a facade, creating the impression of authentic information with associated processes where none will actually occur.

The techniques we develop should be capable of being operated in a proactive mode providing a constant

Table 1: AFRL/RIGA FY15 Deception Efforts

TITLE: CINDAM (Customized Information Networks for Deception and Attack Mitigation) PERFORMER: Applied Communication Sciences OBJECTIVE: The objective of this effort is to research and develop a proof-of-concept capability to create individualized deceptive environments that present unique and dynamic changing views of the network to each host in order to impede an adversary from mounting a successful attack with minimal impact to mission services provided to legitimate users. SCOPE: The scope of this effort is to develop a proof-of-concept Customized Information Networks for Deception and Attack Mitigation (CINDAM) capability by leveraging and building on Software Defined Networking (SDN) and other standard network services. CINDAM leverages SDN, network virtualization, and standard network services to create individualized deceptive environments with illusory and continuously shifting topologies for each network host.
TITLE: KAGE (Keeping the Adversary Guessing and Engaged) PERFORMER: Raytheon/BBN OBJECTIVE: Research, develop, demonstrate, and evaluate technologies and mechanisms that keep the attacker engaged and make him believe that he is succeeding, but not allowing him to impact mission critical functions of the protected enclave under attack. SCOPE: Core R&D activities such as design, development and testing of new algorithms and software components, integration with other COTS or research products in the area of software defined networks and virtualization, as well as transition-focused demonstration and evaluation. One research focus of the proposed effort is to apply the general idea of deception and manipulating the adversary's decision loop in the context of cyber-attacks and cyber-defense. Topics of investigation include deception maneuvers at various system layers, and at different attack stages. Technologies that will be explored in this focus include software defined networking and end point virtualization.
TITLE: Megatron (Advanced Deception Concepts to Support Defensive Cyber Operations) PERFORMER: Assured Information Security OBJECTIVE: The objective of this effort is to execute research and development that will result in proof-of-concept deception techniques that are capable of delaying and impeding the adversary's activities (thereby prompting them to invest more time in the reconnaissance and weaponization stages of attack and less time in the execution of their mission), discouraging further exploits by providing misinformation that leads to outdated or ineffective exploits, and informing defenses and prompting predictable adversarial behavior. SCOPE: Two phased approach; 1) a research phase and, 2) rapid development phase. The first phase will foster a detailed understanding of the various aspects of cyber deceptions and their applicability to proactive cyber defense. This research will include extending previous work regarding taxonomies that allow for cyber deception techniques to be categorized, evaluating additional CONOPS for deceptions, and broadening the application of metrics to evaluate deception techniques. The second phase of this effort will create several proofs-of-concept for cyber deceptions that show the most promise for meeting program requirements and increasing defensive cyber capabilities.

confusion component, or may be employed by a commander only when additional obfuscation is required. Any techniques employed must appear to be genuine to an attacker, but at the same time be transparent to authorized users such that they do not waste unnecessary time, effort, or resources. Techniques at any layer of the protocol stack may be explored and implemented, but should be complementary to and/or leverage other DoD developed technologies.

AFRL/RIGA initiated three deception contract awards under the Cyber Deception project in FY15. These efforts are summarized in Table 1. Additionally, there are six Phase 1 Small Business Innovative Research (SBIR) efforts that fall under the Cyber Deception project. The topic areas these efforts relate to are summarized in Table 2.

References

[1] Joint Publication 3-13.4, Military Deception, 26 January 2012

[2] Bodeau, D., & Graubart, R. (2013, November). Intended effects of cyber resiliency techniques on adversary activities. In Technologies for Homeland Security (HST), 2013 IEEE International Conference on (pp. 7-11). IEEE.

[3] J. Lowry, R. Valdez, B. Wood, "Adversary Modeling to Develop Forensic Observables." Digital Forensic Research Workshop, 2004.

[4] W. Tirenin and D. Faatz, "A Concept for Strategic Cyber Defense," Military Communications Conference (MILCOM) '99, 1999.

[5] The Deception Toolkit Home Page and Mailing List, **http://www.all.net/dtk/**.

[6] Thwarting cyber-attack reconnaissance with inconsistency and deception. Rowe, N and Goh, HC, Information Assurance and Security Workshop, 2007. IEEE SMC, 2007.

[7] Chabrow, E. Intelligent Defense Against Intruders. Government Information Security. [Online] May 23, 2012. **http://www.govinfosecurity.com/interviews/intelligent-defense-against-intrud ers-i-1565**.

Table 2: AFRL/RIGA FY15 SBIR Cyber Deception Topics

TECHNOLOGY AREA: Cyber Deception for Network Defense OBJECTIVE: Research and develop technology to provide a cyber deception capability that could be employed by commanders to provide false information, confuse, delay, or otherwise impede cyber attackers to the benefit of friendly forces. SCOPE: Examination of typical attack steps of reconnaissance (where the enemy researches, identifies and selects the target), scanning (where detailed information about the target is obtained allowing a specific attack to be crafted), gaining access (where the attack is carried out), and maintaining access (where the attack evidence is deleted and information is exfiltrated or altered/destroyed) to identify where and how deception technologies can be brought to bear to thwart the objectives of an attack. It is believed that deception techniques, working in conjunction with normal cyber defense methods, can alter the underlying attack process, making it more difficult, time consuming and cost prohibitive. Some work has already been done in cyber deception technologies; i.e., honeypots are computers designed to attract attackers by impersonating another machine that may be worthy of being attacked, honeynets take that further by simulating a number of computers or a network, and products such as the Deception Toolkit conveys an impression of the defenses of a computer system that are different from what they really are by creating phony vulnerabilities. Modern day military planners need a capability that goes beyond the current state-of-the-art in cyber deception to provide a system or systems that can be employed by a commander when needed to enable additional deception to be inserted into cyber operations.
TECHNOLOGY AREA: Host-Based Solutions for Anti-Reconnaissance and Cyber Deception OBJECTIVE: New and novel approaches to reduce the adversary's ability to gain an accurate and comprehensive picture of a target environment. SCOPE: There is a need for solutions capable of, at the host-level, increasing the complexity of the target surface to the attacker and limit the exposure of vulnerabilities. Attackers are capable of observing crucial components and configurations of static target operational environments and the information that is available through public fingerprinting technologies. Much of this information is communicated through standard Internet browsing technologies available to users; to an attacker this is crucial information about a system that can lead to successful exploitation. The proposed solution must falsify externally reported settings and provide a method to randomize the applications utilized. By exposing attackers to a dynamic environment, their ability to perform reconnaissance on a target system will be greatly reduced, while the cost of weaponization and delivery of an exploit will increase, thereby significantly decreasing the likelihood of exploitation [7].
TECHNOLOGY AREA: Infrastructure Agnostic Solutions for Anti-Reconnaissance and Cyber Deception OBJECTIVE: This topic seeks to provide new and novel approaches to delaying, disrupting and deceiving adversaries engaged in active network reconnaissance. SCOPE: Secure, infrastructure agnostic, solutions designed for cyber agility and anti-reconnaissance. Such solutions must effectively prevent traffic analysis, and must implement evasive and deceptive techniques such as misreporting source and destination IP and/or MAC addresses, and intermittently changing those addresses. The technology must be capable of preventing an adversary from accurately determining the direction or volume of information moving within the network, or the size or topology of the network itself, and must be capable of taking measures to prevent, detect, and cease communication with non-compliant or rogue clients within the environment.

[8] U.S. Naval Academy. Phases of a Cyber-Attack / Cyber-Recon. US Naval Academy. [Online] http://www.usna.edu/CS/si110arch/si110AY13F/lec/l32/lec.html

About the Authors

Dave Climek is Deputy Branch Chief and Technical Advisor in the AFRL Cyber Assurance Branch Rome, NY. He has over 35 years of experience in C4ISR Systems Engineering, Military Communications Systems and Cyber Defense technologies. He has earned certifications in CISSP and CEH and Master's Degrees in Information Assurance, Business Management and Telecommunications.

Anthony Macera, PMP, is the Deputy Program Manager for AFRL's Cyber Agility Program Rome, NY. He has over 25 years of experience in Information Management and Cyber Defense. Mr. Macera has a Bachelor's Degree in Electrical Engineering and Master's Degree in Computer and Information Science.

Walt Tirenin is the Program Manager for AFRL's Cyber Agility Program, Information Directorate, Rome, NY. He has 31 years of experience in communications and information assurance. Mr. Tirenin has a Bachelor's Degree in Electrical Engineering and a Master's Degree in Management Science, Systems Management.

RELEASE STATEMENT

Approved for Public Release; Distribution Unlimited: 88ABW-2015-4691 20151002

A Science of Network Configuration

By Sanjai Narain, Dana CheeBrian Coan, Ben Falchuk, Samuel Gordon, Jaewon Kang, Jonathan Kirsch, Aditya Naidu, Kaustubh Sinkar, Simon Tsang, Sharad Malik, Shuyuan Zhang, Vahid Rajabian-Schwart, and Walt Tirenin

Configuration is the glue for logically integrating network components to satisfy end-to-end requirements on security and functionality. Every component has a finite number of configuration variables that are set to definite values. It is well-documented that configuration errors are responsible for 50%-80% of network vulnerabilities and downtime and it can take months to set up and adapt networks. This is because the large conceptual gap between requirement and configuration is manually bridged. This paper presents a Science of Configuration to automatically bridge this gap. It contains tools for requirement specification, configuration synthesis, repair, vendor-specific adaptation, visualization, emulation, verification, distributed configuration, in-band configuration, reconfiguration planning and moving-target defense. The Science leverages modern SMT solvers that can solve a million constraints in a million variables in seconds, and group communication protocols that provide total-ordering message delivery guarantees. The Science is motivated by the same problems that Software-Defined Networking is, but unlike SDN, exploits the full power of conventional networking devices that don't separate the control and data planes. Applications of the Science are sketched to cyber defense exercises and network planning.

1. Introduction

Configuration is the glue for logically integrating network components to satisfy end-to-end requirements on security and functionality. Every component has a finite number of configuration variables that are set to definite values. It is well-documented that configuration errors are responsible for 50%-80% of network vulnerabilities and downtime and it can take months to set up new networks or adapt them to changing requirements and state. This is because the large conceptual gap between requirements and configuration is manually bridged. Requirements induce complex dependencies or constraints between configurations within and across components at and across multiple protocols. The number of requirements, components, configuration variables and possible values is large, so the spaces over which one must search for a satisfying configuration are astronomical. Manual search through these spaces is infeasible.

This paper presents a Science of Configuration to automatically bridge the above gap. It leverages modern SMT solvers [10, 28] that can solve a million constraints in a million variables in seconds, and thus efficiently search through the above spaces. The Science also leverages group communication protocols that provide total-ordering message delivery guarantees [11, 9]. The Distributed Assured and Dynamic Configuration System is an implementation of this Science. In its simplest form a DADC system consists of a single controller as shown in Figure 1. DADC offers the following tools:

- **Intuitive requirement specification language.** This allows one to precisely specify requirements in about the same time it takes to visually depict these. It contains a catalog of fundamental logical structures and relationships at and across multiple protocol layers. These can be composed to specify a very large class of requirements. A visual version of this language is under development.

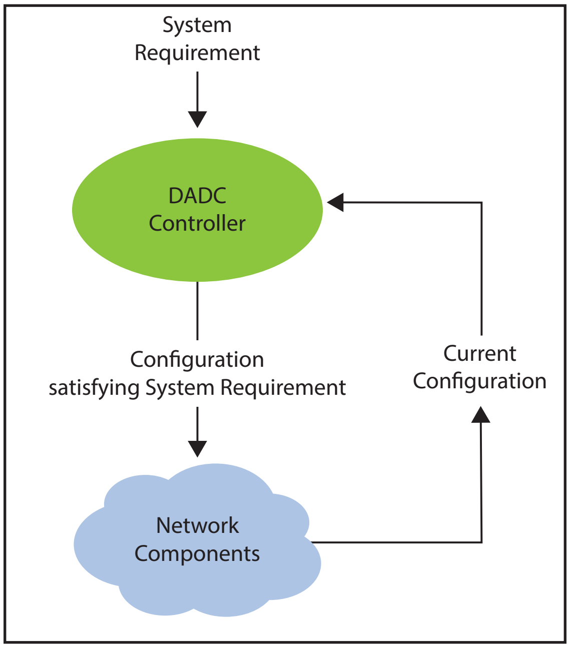


Figure 1. DADC single controller architecture

- **Configuration synthesis.** This computes configurations satisfying requirements and thus eliminates errors due to manual computation. Thus, it automates a central intellectual task that, currently, is manually accomplished.
- **Configuration repair.** This identifies configurations that are non-compliant with requirements and calculates the minimum-cost configuration changes to bring these into compliance.
- **Vendor-specific adapters.** These parse configuration files of components from different vendors into an abstract, vendor-neutral information model, generate these files from abstract configurations, and apply (download) files to components.
- **Visualization.** This provides a conceptual understanding of the network via visualizations of a large number of logical structures and relationships latent in the current configuration. It makes visible the presence or absence of structural defects.
- **Emulation.** This allows a network planner to evaluate complex architectural concepts in a Cisco or Linux network in minutes rather than the days or months it takes with physica`l networks. Emulation is *not* simulation. It reproduces the behavior of physical networks with perfect fidelity, except possibly for performance behavior.
- **Verification.** This evaluates the correctness of requirements and the inclusion and equivalence of access-control policies. This also evaluates the propagation of an adversary's influence through a network with an algorithm for path finding in the presence of access-control lists and path constraints.
- **Distributed configuration.** This enforces global configuration consistency in the absence of a centralized configuration authority.
- **In-band configuration.** This removes the need to create an out-of-band network for configuration management.
- **Reconfiguration planning.** This computes the order in which to reconfigure components without violating safety requirements during transition.
- **Moving-target defense.** This periodically changes network configuration in such a way that legitimate users continue to obtain service yet the adversary is confused about what configurations are new and old. This technique is called configuration-space randomization. Configuration is a network's "DNA," so its knowledge can allow an adversary to identify high-value targets such as single points of failure. Configuration-space randomization makes it harder for an adversary to gain such knowledge.

At present, DADC supports Cisco IOS and ASA, Linux, Juniper, Vyatta and Palo Alto for IP, IPv6 IPSec, RIP, OSPF, static routing, HSRP, VLAN, GRE, mGRE, QoS and access-control lists.

Section 2 illustrates the large gap between requirement and configuration and why it is hard to manually bridge. Section 3 sketches the design of DADC tools in the context of this example. Section 4 shows that the performance of DADC is adequate for networks of realistic size and complexity. Section 5 outlines applications of DADC to network planning and cyber defense exercises. Section 6 outlines the relationship of DADC to current work, especially, Software-Defined Networking. Section 7 contains an overview of SAT and SMT solvers and is followed by references.

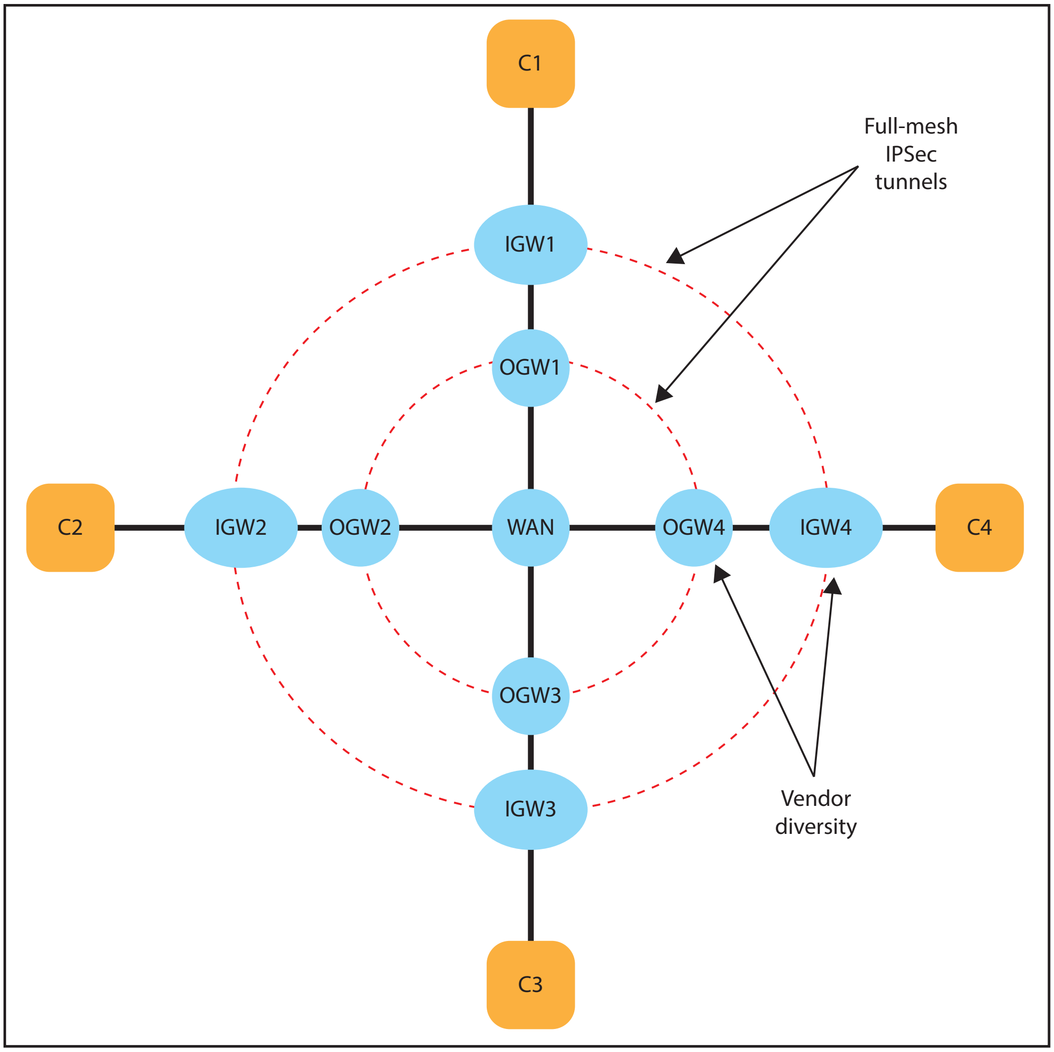


Figure 2. Network for secure transmission over public networks using commercial components

2. The gap between requirement and configuration

Figure 2 shows a network architecture for securely transmitting information between hosts C1-C4. The architecture is inspired by the Commercial Solutions for Classified standard [2] permitting the use of commercial equipment and public networks for such transmission. The requirements that the network needs to satisfy are as follows:

Structural requirements

- There are four enclaves
- Each enclave has a client, an inner gateway and an outer gateway
- In each enclave, inner and outer gateways are from different vendors
- Outer gateways are connected to the WAN router

Security requirements

- There is a full-mesh of IPSec tunnels between the inner gateways
- There is another full-mesh of IPSec tunnels between the outer gateways

Routing requirement

- Static routes are used to forward all traffic from clients to inner gateways, from inner gateways to outer gateways, and from outer gateways to the WAN
- Dynamic (OSPF) routing is used in the WAN router

Even for this small example, it is quite hard for a human to translate the requirements into concrete configurations. He cannot choose values of configuration variables independently of each other. In fact, he needs to satisfy 503

Table 1. Cisco configuration file for single router OGW1 in Figure 2

```
! Preamble
hostname OGW1
version 12.4
no ip domain-lookup

! IPSec configurations
crypto isakmp policy 10
  authentication pre-share
  group 5
  encryption 3des
  hash sha

crypto ipsec transform-set esp-3des-esp-sha-
hmac esp-3des esp-sha-hmac
  mode tunnel

crypto isakmp key 0 1234567890 address 20.0.0.1

crypto map on_eth2 10 ipsec-isakmp
  set peer 20.0.0.1
  set transform-set esp-3des-esp-sha-hmac
  set pfs group5
  match address OGW1eth2OGW2eth3

crypto isakmp policy 10
  authentication pre-share
  group 5
  encryption 3des
  hash sha

crypto ipsec transform-set esp-3des-esp-sha-
hmac esp-3des esp-sha-hmac
  mode tunnel

crypto isakmp key 0 1234567890 address 30.0.0.1

crypto map on_eth2 20 ipsec-isakmp
  set peer 30.0.0.1
  set transform-set esp-3des-esp-sha-hmac
  set pfs group5
  match address OGW1eth2OGW3eth4
```

```
crypto isakmp policy 10
  authentication pre-share
  group 5
  encryption 3des
  hash sha

crypto ipsec transform-set esp-3des-esp-sha-hmac
esp-3des esp-sha-hmac
  mode tunnel

crypto isakmp key 0 1234567890 address 40.0.0.1
crypto map on_eth2 30 ipsec-isakmp
  set peer 40.0.0.1
  set transform-set esp-3des-esp-sha-hmac
  set pfs group5
  match address OGW1eth2OGW4eth5

ip access-list extended OGW1eth2OGW2eth3
  permit ip 1.0.0.0 0.0.0.255 2.0.0.0 0.0.0.255

ip access-list extended OGW1eth2OGW3eth4
  permit ip 1.0.0.0 0.0.0.255 3.0.0.0 0.0.0.255

ip access-list extended OGW1eth2OGW4eth5
  permit ip 1.0.0.0 0.0.0.255 4.0.0.0 0.0.0.255

! Interface configurations

interface eth1
  no shutdown
  ip address 1.0.0.4 255.255.255.0

interface eth2
  no shutdown
  ip address 10.0.0.1 255.255.255.0
  crypto map on_eth2

! Routing configurations

ip route 0.0.0.0 0.0.0.0 10.0.0.128
```

constraints over 237 configuration variables across the 13 components. Examples of constraints are: keys, encryption and hash algorithms should be the same at both ends of every IPSec tunnel; IPSec tunnel local addresses should be equal to the IP addresses of originating interfaces, tunnel peer values should be symmetric and static routes should direct traffic into IPSec tunnels. The total number of configuration commands is 2239 with Linux hosts, Juniper inner gateways and Cisco outer gateways and WAN. The Cisco configuration file for OGW1 is listed in Table 1. Cisco configuration file for single router OGW1 in Figure 2. It lists values of IPSec configurations (peers, keys, encryption and hash algorithms) for the three outer tunnels, interface configurations (addresses, masks and originating tunnels), and static routing configuration.

The next section outlines how DADC automatically bridges the gap between requirements and configurations.

3. DADC Design

This section sketches the design of DADC tools.

3.1 Intuitive requirement specification language

DADC allows one to specify requirements or properties that a network should satisfy. It offers a Requirement Library of useful constraints. The Library contains logical structures and relationships that are typically used in network architecture diagrams. One heuristic for identifying these is by formalizing the notion of "correct configuration." For each protocol, we ask how a group of agents executing that protocol should be configured so they accomplish a joint goal associated with that protocol. Answers to this question are encoded as constraints in the Requirement Library. Library constraints can be composed with logical operators to specify a very large class of requirements. In particular, the AND operator formalizes the superposition of logical structures in typical network architecture diagrams. For example, the *entire* network in Figure 2 is specified in DADC as the conjunction of requirements in Table 2. DADC specification of entire network of Figure 2:

The `component` requirements declare component vendors. Together they satisfy diversity requirements. The `enclave` requirement means there is linear IP connectivity

Table 2. DADC specification of entire network of Figure 2

-- **Structural requirements**

```
component type linux
       C1 C2 C3 C4
component type junos
       IGW1 IGW2 IGW3 IGW4
component type cisco
       OGW1 OGW2 OGW3 OGW4 wan

enclave 201.0.0.0 24 C1 eth0 IGW1 eth0
        1.0.0.0 24 IGW1 eth1 OGW1 eth1
        10.0.0.0 24 OGW1 eth2 wan eth2

enclave 202.0.0.0 24 C2 eth0 IGW2 eth0
        2.0.0.0 24 IGW2 eth1 OGW2 eth1
        20.0.0.0 24 OGW2 eth3 wan eth3

enclave 203.0.0.0 24 C3 eth0 IGW3 eth0
        3.0.0.0 24 IGW3 eth1 OGW3 eth1
        30.0.0.0 24 OGW3 eth4 wan eth4

enclave 204.0.0.0 24 C4 eth0 IGW4 eth0
        4.0.0.0 24 IGW4 eth1 OGW4 eth1
        40.0.0.0 24 OGW4 eth5 wan eth5
```

-- **Inner IPSec tunnels**

```
full mesh ipsec tunnels
     eth0 IGW1 eth1
     eth0 IGW2 eth1
     eth0 IGW3 eth1
     eth0 IGW4 eth1
```

-- **Outer IPSec tunnels**

```
full mesh ipsec tunnels
     eth1 OGW1 eth2
     eth1 OGW2 eth3
     eth1 OGW3 eth4
     eth1 OGW4 eth5
```

-- **WAN routing**

```
ospf domain 0 0 0
     wan eth2
     wan eth3
     wan eth4
     wan eth5
```

between the client, inner gateway, outer gateway and the wan, and that all packets originating from a node are forwarded to its successor. The `full mesh ipsec tunnels` requirement means there is a full mesh of IPSec tunnels between the interfaces on the right, encrypting all traffic originating at the subnet of the left interfaces. Finally, the `ospf domain` requirement means OSPF is enabled at all interfaces of the WAN router and that all interfaces are in area 0 with default hello and dead timer values. Surprisingly, no additional routing is needed. Packets originating at clients are forwarded to inner gateways, encrypted, forwarded to outer gateways, re-encrypted and forwarded to the WAN router. That router, using OSPF-learned routes, redirects packets to the remote interface of outer tunnels where they are decrypted, forwarded to remote interfaces of inner gateways, decrypted and forwarded to the clients.

No IP addresses, static routing, or IPSec or OSPF configurations are explicitly specified. These are all computed when the requirement is solved. Conventional configuration languages force one to specify all values of configuration variables. By contrast, DADC allows one to specify just the constraints that these variables must satisfy. DADC solves these constraints to compute the values. This transition from specifying explicit variable values to specifying the conditions that these must satisfy marks a major increase in expressive power.

The entire network is specified in a single file. This is a major simplification over current practice in which a separate configuration file has to be created for each component. This makes it much harder to enforce complex dependencies across multiple files because of the context switching between different files.

In conventional configuration languages one is often forced to write requirements in a definite order. For example, static routes or firewall rules cannot be written unless the IP addresses in their fields are determined. If these addresses change, then these routes and rules have to be manually updated with the new addresses. DADC eliminates such ordering and manual updates by allowing requirements to contain variables. When variable values are computed, the requirements are automatically updated with the new values. No special manual action needs to be taken when values change. The constraint solver automatically accomplishes the update.

Not only are the semantics of the language simple, so is the syntax. A requirement is a sequence of identifiers separated by white spaces. There are no special symbols such as commas, colons, semicolons, curly, round or square braces. Except for those in IP addresses, there are no dots either. Requirements can be split across multiple lines. Each unindented line is assumed to start a requirement. All indented lines following it are assumed to belong to that requirement. The sequence of all identifiers in these lines is accumulated and parsed. All requirements in a specification file are assumed to be composed by conjunction. Thus, one does not have to write the conjunction operator for these top-level requirements. The order in which requirements are written in a file is immaterial. DADC checks whether a requirement is syntactically correct, and if not, outputs an error message.

3.2 Configuration synthesis

DADC transforms requirements into primitive constraints in the language of an SMT solver. For example, the requirement:

```
enclave 201.0.0.0 24 C1 eth0 IGW1 eth0
        1.0.0.0 24 IGW1 eth1 OGW1 eth1
        10.0.0.0 24 OGW1 eth2 wan eth2
```

is transformed into the conjunction of the following requirements:

```
subnet 201.0.0.0 24 C1 eth0 IGW1 eth0
subnet 1.0.0.0 24 IGW1 eth1 OGW1 eth1
subnet 10.0.0.0 24 OGW1 eth2 wan eth2
next hop C1 eth0 = ip address IGW1 eth0
next hop IGW1 0.0.0.0 0 = ip address OGW1 eth1
next hop OGW1 0.0.0.0 0 = ip address wan eth2
```

The first is further transformed into the conjunction of primitive constraints that the IP addresses of `C1 eth0` and `IGW1 eth0` are distinct, are in the range `201.0.0.0/24`, but are not equal to the first and last addresses in the range.

For configuration synthesis, DADC simply solves the accumulated primitive constraints using an SMT solver [17]. An excerpt from the solution for the CSfC requirements is:

```
ip address C1 eth0 = 201.0.0.5
ip address IGW1 eth0 = 201.0.0.4
next hop C1 0.0.0.0 0 = 201.0.0.4
```

3.3 Configuration repair

DADC parses configuration files of an existing network into a large constraint `Current Configuration` of the form $x_1=c_1, .., x_k=c_k$ where each x_i is a configuration variable in the requirement and c_i is its value in one of the uploaded configuration files. If the conjunction (`System Requirement ∧ Current Configuration`) is unsolvable, then the solver produces an "unsat-core." This is a typically small constraint that is itself unsatisfiable and whose unsatisfiability causes that of the conjunction.

The unsat-core can be taken to be a root-cause of the unsolvability of the conjunction. If in the unsat-core there is an equation of the form `x=c` that occurs in `Current Configuration`, then this equation represents a configuration error. This error can be repaired by deleting this equation from `Current Configuration` and reattempting the solution to (`System Requirement ∧ Current Configuration`). This step is repeated until either a solution is obtained or it is no longer possible to find an equation of the form `x=c` in the unsat-core. At that stage, the algorithm halts and outputs the unsat-core. That unsat-core represents a design flaw in the `System Requirement` itself. This flaw is best resolved by the user because the `System Requirement` represents his intent.

Another repair option is to use a MaxSAT solver [22]. One can associate weights with variables representing the cost of changing their values. Then, MaxSAT can find the minimum cost change to values in `Current Configuration` so that `System Requirement` becomes true. If no weights are associated, MaxSAT can find the minimum number of variables to change.

3.4 Vendor-specific adapters

From the solution produced by the constraint solver, DADC generates configuration files for all components referenced in `System Requirement`. When these are successfully applied to components, their joint configuration would satisfy `System Requirement`. For example, the Cisco configuration file in Section was automatically generated by DADC.

DADC parses configuration files of different vendors to produce `Current Configuration`. However, it is infeasible to base parsing on writing grammars for vendor-specific configuration languages. Instead, DADC creates a database of the configuration file and then queries it to obtain the values of configuration variables in `System Requirement` [23]. This approach avoids the need to model the entire language when we are only interested in a subset of it.

DADC also defines an internal, abstract, vendor-neutral information model for configuration. All of its algorithms work on this model. Adapters for configuration generation and parsing are developed for each vendor. DADC communicates with devices using SNMP for Cisco and SSH for other vendors. It applies and reads entire files rather than individual commands.

3.5 Visualization of current configuration

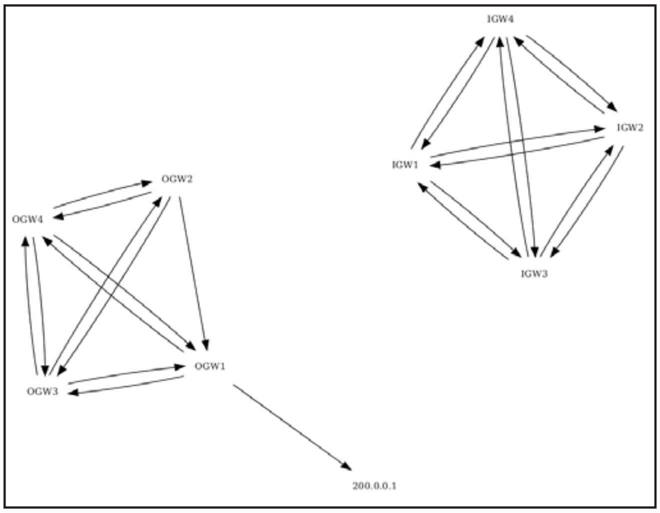
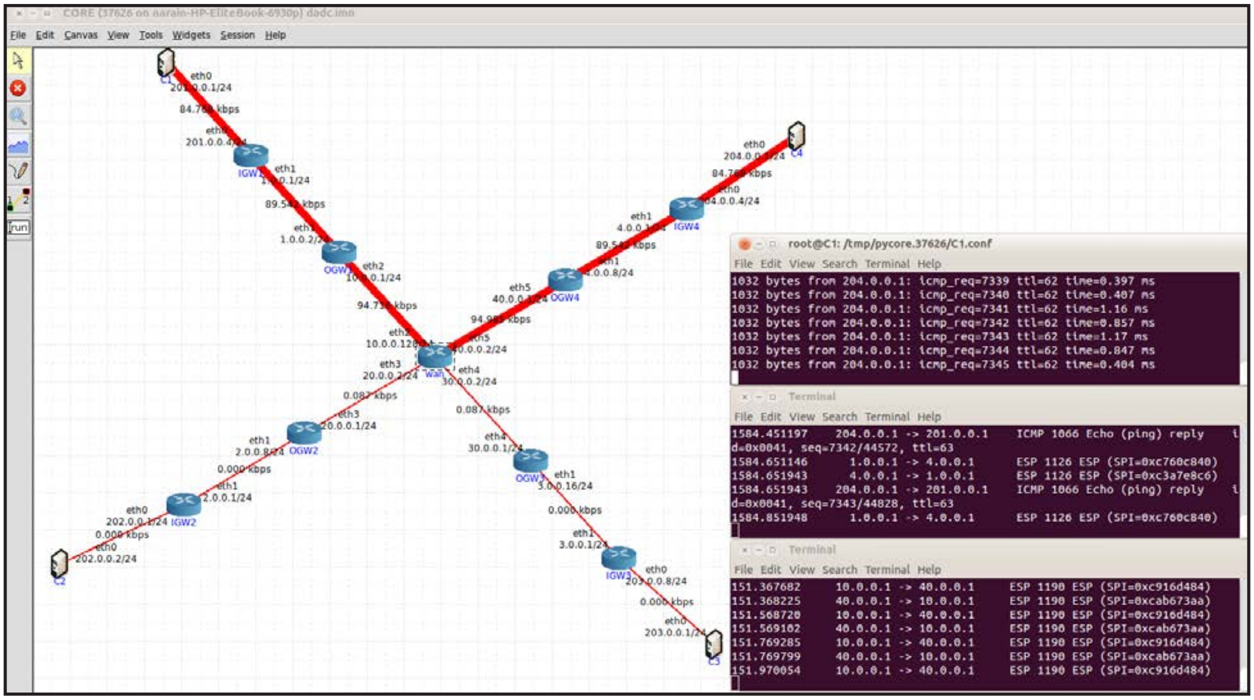


Figure 3. Visualization of current configuration shows a misconfigured outer gateway tunnel

DADC produces visualizations of a number of logical structures latent in the current network configuration. DADC can either read the configuration directly from the components. Or, an administrator can gather the configuration files, zip them, and input them to DADC. In many cases, the latter method is preferable because of its non-invasiveness. Visualizations can provide a good conceptual understanding of the network. These can also uncover structural defects. For example, suppose that in the configuration of OGW1 in Section , we change the peer value from `20.0.0.1` to a non-existent `200.0.0.1`. DADC then produces the visualizations for IPSec tunnels shown in Figure 3. The inner gateway tunnels form a full-mesh as expected. The other gateway tunnels don't. What should have been a tunnel from OGW1 to OGW2 is now pointing to the non-existent address.

3.6 Emulation

DADC has been integrated with two network emulation systems, GNS3 [7] and CORE [3]. In addition to generating component configurations satisfying a requirement, DADC also generates a configuration file for emulation. When the appropriate emulator is started with this file, the observed



```
1032 bytes from 204.0.0.1: icmp_req=7339 ttl=62 time=0.397 ms
1032 bytes from 204.0.0.1: icmp_req=7340 ttl=62 time=0.407 ms
1032 bytes from 204.0.0.1: icmp_req=7341 ttl=62 time=1.16 ms
1032 bytes from 204.0.0.1: icmp_req=7342 ttl=62 time=0.857 ms
1032 bytes from 204.0.0.1: icmp_req=7343 ttl=62 time=1.17 ms
1032 bytes from 204.0.0.1: icmp_req=7344 ttl=62 time=0.847 ms
1032 bytes from 204.0.0.1: icmp_req=7345 ttl=62 time=0.404 ms
```

```
1584.451197    204.0.0.1 -> 201.0.0.1    ICMP 1066 Echo (ping) reply    i
d=0x0041, seq=7342/44572, ttl=63
1584.651146      1.0.0.1 -> 4.0.0.1      ESP 1126 ESP (SPI=0xc760c840)
1584.651943      4.0.0.1 -> 1.0.0.1      ESP 1126 ESP (SPI=0xc3a7e8c6)
1584.651943    204.0.0.1 -> 201.0.0.1    ICMP 1066 Echo (ping) reply    i
d=0x0041, seq=7343/44828, ttl=63
1584.851948      1.0.0.1 -> 4.0.0.1      ESP 1126 ESP (SPI=0xc760c840)
```

```
151.367682     10.0.0.1 -> 40.0.0.1      ESP 1190 ESP (SPI=0xc916d484)
151.368225     40.0.0.1 -> 10.0.0.1      ESP 1190 ESP (SPI=0xcab673aa)
151.568720     10.0.0.1 -> 40.0.0.1      ESP 1190 ESP (SPI=0xc916d484)
151.569102     40.0.0.1 -> 10.0.0.1      ESP 1190 ESP (SPI=0xcab673aa)
151.769285     10.0.0.1 -> 40.0.0.1      ESP 1190 ESP (SPI=0xc916d484)
151.769799     40.0.0.1 -> 10.0.0.1      ESP 1190 ESP (SPI=0xcab673aa)
151.970054     10.0.0.1 -> 40.0.0.1      ESP 1190 ESP (SPI=0xc916d484)
```

Figure 4. CORE Emulation of network in Figure 2

behavior of the network is indistinguishable from that of a physical network satisfying the same requirement except, perhaps, for performance. The net result is that complex requirements can be specified and evaluated in minutes rather than the days or months it currently takes to compute configurations, build a physical network, configure it and run tests. Figure 4. CORE Emulation of network in Figure 2 shows the CORE emulation of the network of Figure 2. It shows that ping from C1 to C4 succeeds, as we expect. It also shows that these packets are encrypted, i.e., encapsulated inside ESP packets originating at IGW1/eth1 with address 1.0.0.1 and destined to IGW4/eth1 with address 4.0.0.1. Finally, it shows the second layer of encryption inside ESP packets originating at OGW1 eth2 with address 10.0.0.1 and destined to OGW4/eth5 with address 40.0.0.1.

3.7 Verification

Given a requirement, DADC guarantees that any solution it generates satisfies `System Requirement`. However, the requirement itself may be incorrect. Incorrectness can take several forms.

3.7.1 Unsatisfiable requirement

One form of incorrectness is that `System Requirement` itself is unsatisfiable. Then, DADC produces an unsat-core. For example, if to the requirement of Section we add the constraint `ip address IGW1 eth0 = 1.1.1.1`, DADC produces the unsat-core below stating that it is not possible for the address of IGW1/eth0 to be `1.1.1.1` and yet belong to the range `201.0.0.0/24`.

```
ip address IGW1 eth0 = 1.1.1.1
bitwise and 255.255.255.0 ip address IGW1
eth0 = 201.0.0.0
```

DADC does not attempt to repair such a requirement as it represents a design error that is best addressed by the designer.

3.7.2 Requirement does not satisfy intent

Another form of incorrectness is that while `System Requirement` is correct, it does not satisfy a design intent. One way of checking if it does is to express a simpler form of intent and check if the requirement implies the simpler form. This can be accomplished by checking that the requirement and the negation of the simpler form is unsatisfiable. In other words, any configuration that satisfies the requirement also satisfies the simpler form.

3.7.3 Firewall verification

DADC allows one to check the inclusion and equivalence between firewall policies. Policy `P1` is included in a policy `P2` provided every packet permitted by `P1` is permitted by `P2`. `P1` and `P2` are equivalent if each is included in the other. We assume that policies

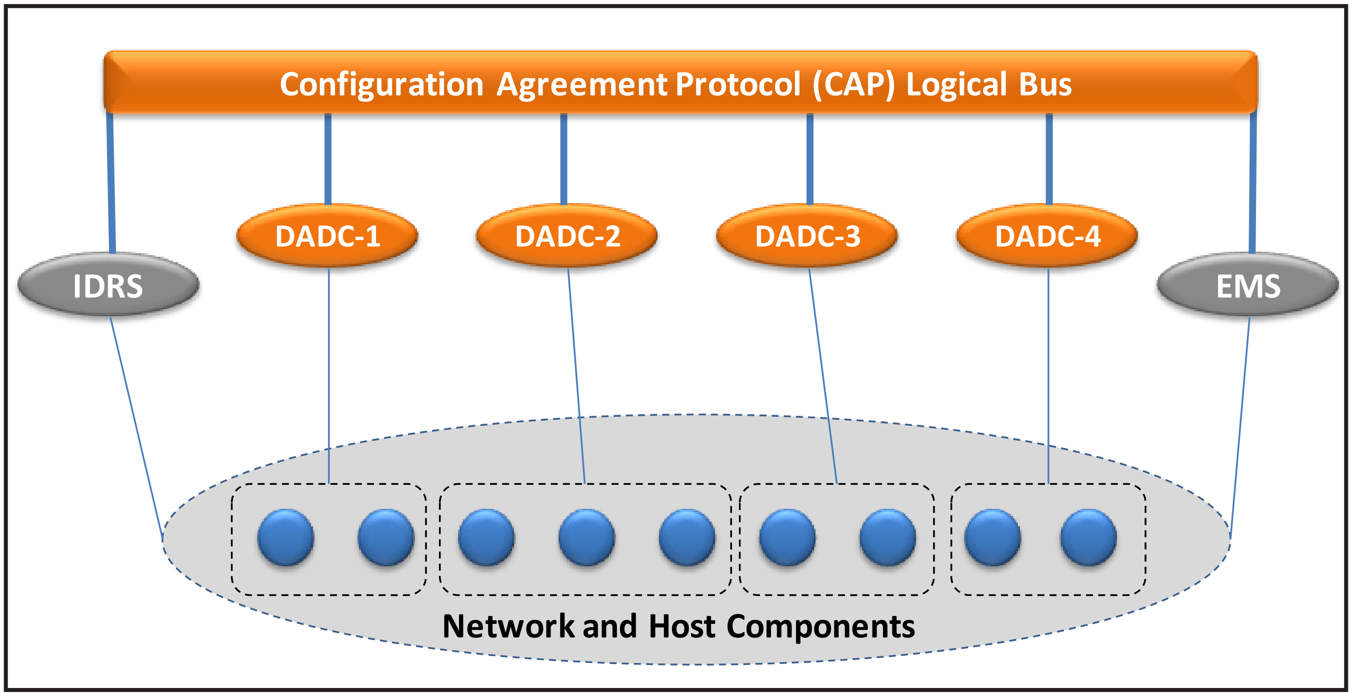
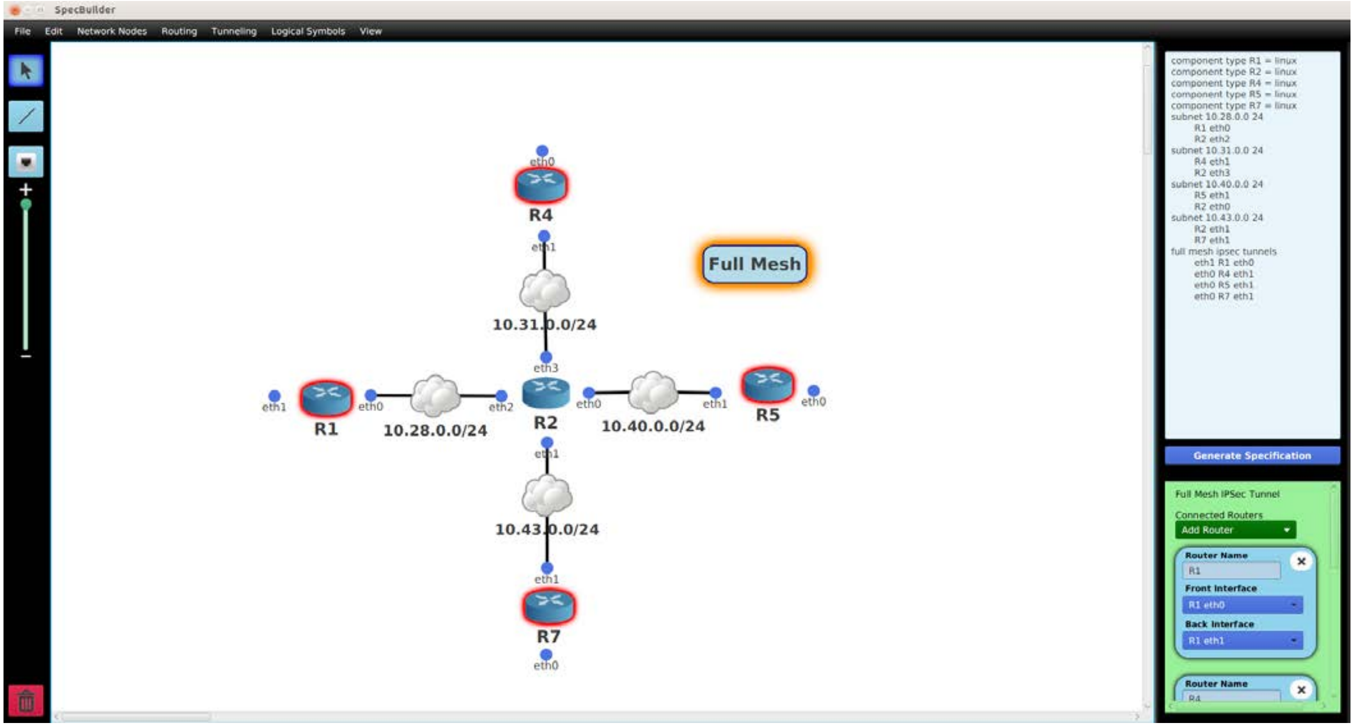

Configuration Agreement Protocol (CAP) Logical Bus
DADC-1
DADC-2
DADC-3
DADC-4
IDRS
EMS
Network and Host Components

Figure 5: Distributed ADC system architecture



```
component type R1 = linux
component type R2 = linux
component type R4 = linux
component type R5 = linux
component type R7 = linux
subnet 10.28.0.0 24
        R1 eth0
        R2 eth2
subnet 10.31.0.0 24
        R4 eth1
        R2 eth3
subnet 10.40.0.0 24
        R5 eth1
        R2 eth0
subnet 10.43.0.0 24
        R2 eth1
        R7 eth1
full mesh ipsec tunnels
        eth1 R1 eth0
        eth0 R4 eth1
        eth0 R5 eth1
        eth0 R7 eth1
```

Router Name: R1
Front Interface: R1 eth0
Back Interface: R1 eth1
Router Name: R4

Figure 6. Visual specification of IP connectivity and a full-mesh of IPSec tunnels

are evaluated on five fields of a packet header: source address, source port, destination address, destination port and protocol. The brute-force approach to solving this problem by enumerating all packet headers and checking whether it is permitted (or not) by each policy is computationally infeasible. There are 2^104 packet headers to enumerate with 32 bit source and destination addresses, 16 bit source and destination ports and 8 bit protocol. DADC solves the inclusion (and equivalence) problem by converting a policy into a constraint on a generic packet header consisting of five variables one for each field. For given values of these variables, the constraint is true iff the packet header is permitted by the policy. Now, `P1` is included in `P2` if the constraint for `P1` and the negation of the constraint for `P2` is unsatisfiable. In other words, it is not possible to find a packet header that is permitted by `P1` but not by `P2`. More details can be found in [21, 23].

3.7.4 Path planning

The problem of finding a path between a source and destination in a graph can be solved in time linear in the size of the graph. However, the problem is much harder if the graph contains firewalls or routers with access-control lists or we allow the placement of additional constraints on paths. The straightforward approach of enumerating all paths and finding one that satisfies constraints is infeasible. The number of paths in a graph can be exponential in the size of the graph. DADC converts this problem into a constraint satisfaction one and thus improves our chances of solving it. It models a path as a constraint on the generic packet header fields, and labels on nodes and edges indicating if they are on the path. The constraint is derived from the topology of the network, from access-control lists on the path, and other user-supplied constraints on the packet and path. The constraint generation algorithm makes use of the one outlined above for representing a firewall policy as a constraint. The path-finding algorithm is inspired by [5].

If a node is compromised then we can use this algorithm to find a new path between a source and destination such that path that avoids this node, and all access-control lists along the path permit the client-server flow. The algorithm can also be used to verify that there are no paths between a compromised node and a sensitive server that permit a given flow. We would check that the solver returns an unsat-core for the requirement that there be such a path.

3.8 Distributed configuration

DADC was originally designed as a centralized system that communicated with network components over an out-of-band network. The distributed version of it [14] removes both of these assumptions. As shown in Figure 5: Distributed ADC system architecture, the set of network components is partitioned into enclaves each controlled by a DADC controller. Each controller has the full functionality of a centralized DADC controller. Controllers communicate with each other over a Configuration Agreement Protocol (CAP) bus. Also communicating over this bus are Enterprise Management Systems and Intrusion Detection and Response Systems that provide information about the dynamic state of components: up, down, compromised.

CAP guarantees that messages are delivered to all controllers in the same order. Therefore, it presents to each controller an identical view of the dynamic state of all components. Each controller also has the identical System Requirement governing the whole network. Upon receipt of a message, each controller solves the System Requirement in the context of the current dynamic state. Since SAT or SMT solvers that we use are deterministic, each controller arrives at identical conclusions about the new configurations of all components, not just its own. Each controller then applies configurations relevant to its enclave to the enclave components, and the entire network converges to a new configuration satisfying System Requirement.

3.9 In-band configuration

The simplest way for a DADC controller to configure network components is over an out-of-band network. If using such a network is infeasible then the controller can try to use the data network itself, i.e., configure in-band. The central challenge of in-band configuration is computing the order in which to reconfigure components. The controller should never be locked out of reaching a component before it has been reconfigured. Routing can be affected not just because routing protocol configurations can redirect traffic but also because access-control lists can block traffic. An algorithm for in-band configuration has been described in [30] where only static routing is assumed. A much more general algorithm (e.g., in the presence of dynamic routing) is possible if the network is assumed to be purely IPv4. Then, we can enable IPv6 at all interfaces of all components and let the controller communicate over the IPv6 network. No additional physical resources are

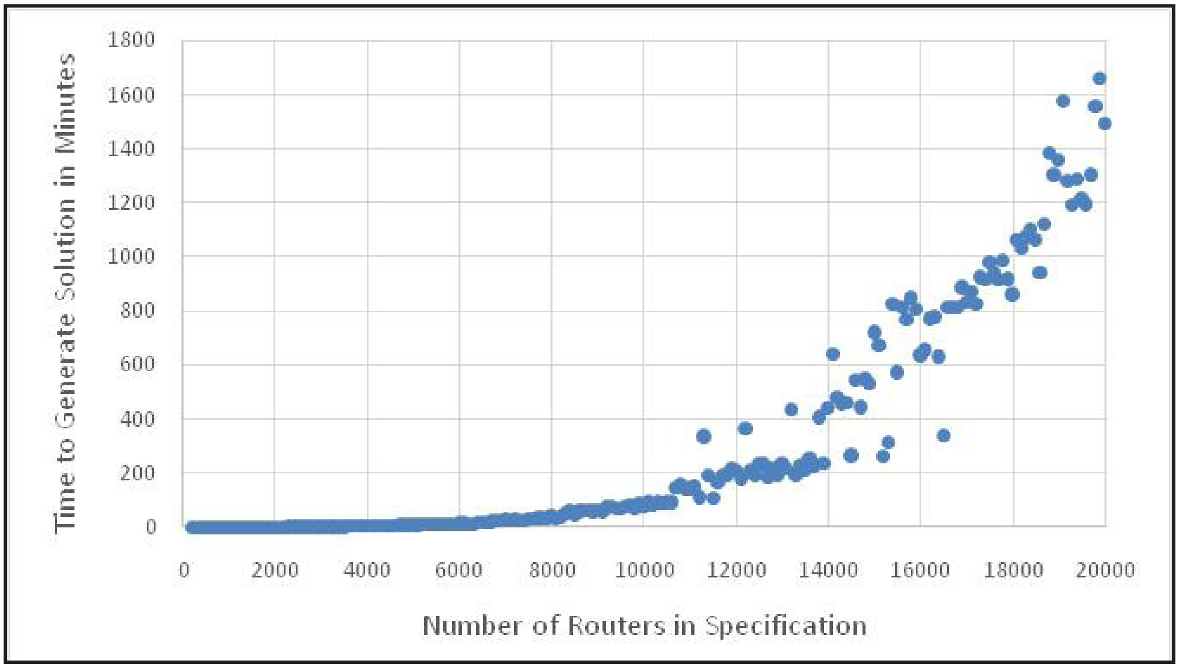


Figure 7. Measuring DADC scalability for a fault-tolerant VPN

required. All modern components support IPv4 and IPv6. The controller can change IPv4 configurations in any order since IPv6 reachability is unaffected by these changes. Since DADC also supports IPv6, the logical out-of-band control network configurations can also be automatically generated!

3.10 Visual specification language

While DADC's specification language is semantically and syntactically simple, a new visual interface simplifies specification even further. One can drag and drop network and relationship objects onto a canvas and specify their attributes. At the click of a button, the text version of the requirement is generated. At the click of another button, the requirement is solved, configurations are generated and an emulation of the network is started up. Thus, one can draw a network concept in DADC and then test it under emulation in a few seconds. For example, a network of outer gateways connected to a WAN router and a full-mesh of IPSec tunnels between them is drawn in Figure 6. Visual specification of IP connectivity and a full-mesh of IPSec tunnels. Clicking the Full Mesh icon highlights the four outer gateways in the full mesh with red circles, but not the WAN router. Clicking on Generate Specification button shows the equivalent text specification. Clicking on "Run CORE Emulation" (not shown) starts up the emulation similar to that in Figure 4. CORE Emulation of network in Figure 2.

3.11 Reconfiguration planning

Once component configurations have been computed, the problem still remains: in what order should these be applied to the components so that an invariant is never violated during the transition? For example, in the network of Figure 2, we may want to apply IPSec configurations before the static routing ones so that when packets flow, they are encrypted. The reconfiguration planning algorithm allows one to specify an invariant as a requirement and transforms it into a constraint on the times at which the invariant variables should change so that the invariant remains true at all times. This constraint is then solved to compute a safe schedule of changes to the variables. Details are available in [20].

3.12 Moving-target defense

Moving-target defense is accomplished by finding a new solution to `System Requirement` subject to the additional constraint that values of some critical variables be different than those in the `Current Configuration`. A set of values is critical if their knowledge would enable an adversary to plan an effective attack. Periodically, DADC selects a critical variable `x`, generates the constraint `not(x=c)` where c is the current value of `x`, strengthens `System Requirement` with it, and reattempts a solution [26].

4. DADC Scalability

To evaluate scalability of DADC, we measured its performance against large network specifications. The network is a fault-tolerant VPN connecting multiple sites as described in [[16]]. Each site contains a gateway router that is connected to a WAN router. All gateway routers are linked in a ring of GRE/IPSec tunnels with OSPF running over them. All WAN routers are also connected in a ring with RIP running over them. Thus, if there are N sites then there are 2*N routers in the specification. We installed DADC on an Ubuntu 14.01 server consisting of two Intel Xeon E5-2697 2.70GHz processors, 54 GB RAM, and a 100 GB SSD Hard Drive. We scaled the number of routers from 20 to 20,000 in increments of 100 and measured the synthesis time, i.e., the time to generate the solution and the Cisco configuration files for all routers. Figure 7. Measuring DADC scalability for a fault-tolerant VPN shows our results. The time to generate a 1,000 router solution was only 10 seconds, while a 10,000 router solution took 76 minutes. Our largest test case of a 20,000 router specification required almost 25 hours.

The synthesis time does eventually scale non-linearly with the number of routers, perhaps because SAT is NP-complete. However, the results are deemed favorable given that there is a very large number of enterprise networks with a few thousand routers. These networks can be efficiently synthesized and managed with DADC in minutes. The time taken for larger networks is still orders of magnitude smaller than with manual practice.

5. DADC Applications

DADC is being transitioned to real enterprises for network planning and cyber defense exercises. In current practice of network planning, one draws network requirement diagrams such as Figure 2, then translates those into abstract configurations, then translates those into vendor-specific configurations, then creates a physical network and applies these configurations to the components. If the network does not work as expected, the error can be at any of the above stages. Thus, it can take a very long time to resolve and fix it. With DADC, one can specify the requirements in its high-level text or visual language and then automatically generate a working network under emulation in under a minute. If the network does not work, it is definitely a problem with the requirements. The translation of requirements into abstract and vendor-specific configurations and the application of these to emulated components are guaranteed to be correct. Thus, network planning becomes far more efficient than with current practice.

In cyber defense exercises, blue teams defend against attacks by red teams. One use of DADC is to evaluate the effectiveness of blue team defensive actions. For example, the blue team may cut off an attacker's access by placing such a stringent access-control list on a router that even legitimate users are blocked. Such blockage may not be obvious as legitimate users are simulated by computers who may not complain when their traffic is disrupted. DADC's diagnosis, visualization and path planning algorithms can be used to check if legitimate services have been disrupted. DADC could, of course, be used to build the cyber defense exercise networks of realistic scale and complexity. Finally, emulation could be used to conduct the entire exercise in a virtualized environment to make it much more efficient for users to try out new attack and defense maneuvers.

6. Relationship with previous work

The first use of SAT solvers for configuration synthesis was reported in [27]. The Alloy [1] system provides a first-order logic language (Boolean logic with individuals and quantifiers over finite domains). Statements in this language are verified by transforming these into Boolean formulas and solving these with a SAT solver. However, quantifier removal, an essential part of translating first-order logic into Boolean, can lead to very large Boolean formulas. Thus, this approach does not scale to networks of realistic size. DADC addresses this problem by preprocessing constraints to solve as much of these as possible using algorithmic methods, leaving behind a constraint that truly requires the power of an SMT solver [18].

The systems described in [8], [12], [6], are only for verifying reachability properties of a network. They are not intended for the other tasks that DADC accomplishes such as configuration synthesis and repair.

DADC has been motivated by the same problems that Software-Defined Networking has been: it is hard to conceptualize networks as a whole, configuration is hard, and networks are not programmable. SDN's approach to solving these problems is to separate the data and control planes. The network fabric contains utterly simple networking devices with only data-plane features such as forwarding and access control. All control-plane features such as routing protocols, tunneling and encryption are abstracted away into a logically centralized controller. The controllers communicate with devices over an out-of-band network using a well-defined API such as Openflow [13]. The biggest concern about this approach is that the powerful control-plane protocols have to be reimplemented from a centralized standpoint. If configuration is hard, programming is a lot harder!

DADC solves the first problem by allowing one to specify network-wide requirements. In other words, the conceptualization of the network as a whole is the set of requirements that it should satisfy. DADC solves the second problem by solving requirements using SMT solvers. What makes configuration hard is that requirements induce complex constraints between configuration variables and these constraints have to be manually solved. By using DADC to automatically and correctly configure existing control-plane protocols, one can fully exploit their power. For the third problem, it relies on interfaces provided by vendors, e.g., SNMP or SSH. The granularity of these interfaces is indeed coarse. However, well-defined APIs are now being offered in the new generation of components. DADC will be extended to use these in the future. A description of the application of DADC to specifying and emulating hybrid networks i.e., with both pure SDN and legacy components, is described in [15].

7. Overview of SAT and SMT solvers

Boolean logic is the most primitive language for modeling constraints. Examples of Boolean constraints are $p \vee q$, $p \wedge q$, $\neg p$, $p \supset q$ where p, q are propositional variables. The satisfiability problem (SAT) is to find values of propositional variables so a given constraint becomes true. For example, $p \wedge q$ has only one solution, $p=t, q=t$, whereas $p \vee q$ has three solutions: $p=t,\ q=t;\ p=t, q=f;\ p=f, q=t$. Even though SAT is NP-complete, modern solvers [28] can often solve millions of Boolean constraints in millions of variables in seconds. The techniques behind these solvers were pioneered by Professor Sharad Malik, one of our coauthors. If a constraint is unsolvable, SAT solvers output an unsat-core, a typically small part of the constraint that is itself unsatisfiable. An unsat-core can be taken to be the "root-cause" of unsatisfiability. For example, the constraint $p \supset q \wedge p \wedge \neg q \wedge u \wedge v \wedge w \wedge x \ \wedge y \wedge z$ has unsat core $p \supset q \wedge p \wedge \neg q$. The variables u, v, w, x, y, z do not contribute to unsatisfiability.

However, Boolean logic is too low-level a language for modeling network constraints. We need to be able to talk about things like routers, interfaces, addresses and relationships between these. While these things and relationships can, in principle, be expressed in Boolean logic, a much more expressive option is to use the languages offered by Satisfiability Modulo Theories (SMT) solvers such as Z3 [33], CVC4 [4] and Yices [32]. These combine SAT solvers with domain-specific ones. Three domains, and their solvers, used in DADC are EUF (Equality of Unintepreted Functions), linear arithmetic and bitvector logic. EUF can be used to model data structures.

References

[1] Alloy: A language and tool for relational models. http://alloy.mit.edu/alloy

[2] Commercial Solutions for Classified Program. https://www.nsa.gov/ia/programs/csfc_program/

[3] Common Open Research Emulator (CORE). http://www.nrl.navy.mil/itd/ncs/products/core

[4] CVC4 SMT solver. http://cvc4.cs.nyu.edu/web/

[5] Fadi A. Aloul, Bashar Al Rawi, Mokhtar Aboelaze. Identifying the Shortest Path in Large Networks using Boolean Satisfiability. http://www.cse.yorku.ca/~aboelaze/publication/ABA06.pdf

[6] George Varghese, Yahoo Labs, Nick McKeown, Peyman Kazemian. Header Space Analysis: Static Checking For Networks. 9th USENIX Symposium on Networked Systems Design and Implementation, 2012. http://cseweb.ucsd.edu/~varghese/PAPERS/headerspace.pdf

[7] GNS3 Emulator. http://www.gns3.com

[8] Haohui Mai, Ahmed Khurshid, Rachit Agarwal, Matthew Caesar, P. Brighten Godfrey, Samuel T. King. Debugging the Data Plane with Anteater http://pbg.cs.illinois.edu/papers/anteater-sigcomm2011.pdf

[9] Jgroups. http://www.jgroups.org

[10] Leonardo De Moura, Nikolaj Bjørner. Satisfiability modulo theories: introduction and applications. Communications of the ACM. Volume 54 Issue 9, September 2011. http://goo.gl/27P0wG

[11] Leslie Lamport. Time, clocks and the ordering of events in a distributed system. Communications of the ACM, Volume 21 Issue 7, July 1978.

[12] Mohammed Noraden Alsaleh, Ehab Al-Shaer, Adel El-Atawy. Towards A Unified Modeling and Verification of Network and

System Security Configuration. 5th Symposium on Configuration Analytics and Automation (SafeConfig 2012) http://goo.gl/bBdq3H

[13] Openflow SDN standard. https://www.opennetworking.org/sdn-resources/openflow

[14] Sanjai Narain, Dana Chee, Chung-Min Chen, Brian Coan, Ben Falchuk, Dov Gordon, Jon Kirsch, Siun-Chuon Mau, Aditya Naidu, Simon Tsang. Declarative, Distributed Configuration. PODC 2014 Distributed Software-Defined Networking Workshop, Paris, France, 2014. http://www.argreenhouse.com/papers/narain/DSDN.pdf

[15] Sanjai Narain, Dana Chee, Sharad Malik, Shuyuan Zhang. Planning Hybrid SDN and Legacy Networks. Open Networking Users Group Conference, Research Track, Columbia University, New York, May 14, 2015. http://www.argreenhouse.com/papers/narain/DADC-ONUG-2015-1.pdf

[16] Sanjai Narain, Dana Chee, Sharad Malik. Demonstrating Assured and Dynamic Configuration over a live, emulated network. http://www.argreenhouse.com/papers/narain/ADC-Live-Demo.pdf

[17] Sanjai Narain, Gary Levin, Vikram Kaul, Rajesh Talpade. Scalable and interactive method of generating and modifying network configurations to enforce compliance with high-level requirements. US 8,315,966. Granted 2012

[18] Sanjai Narain, Gary Levin, Vikram Kaul. Declarative Infrastructure Configuration Synthesis and Debugging. Journal of Network Systems and Management, Special Issue on Security Configuration. 2008.

[19] Sanjai Narain, Gary Levin. Query-based semantic analysis of ad hoc configuration languages for networks. US 8,554,796. Granted 2013.

[20] Sanjai Narain, Gary Levin. Router route reconfiguration planning. US 8,805,770 B2. Granted 2014

[21] Sanjai Narain, Gary Levin. Verifying access control policies with arithmetic quantifier-free form constraints. US 8,826,366 B2. Granted 2014

[22] Sanjai Narain, Konstantin Arkoudas. Optimal network configuration repair. US 8,725,902 B2. Granted 2014.

[23] Sanjai Narain, Rajesh Talpade, Gary Levin. Network Configuration Validation. Chapter in Guide to Reliable Internet Services and Applications, edited by Chuck Kalmanek, Richard Yang and Sudip Misra. Springer Verlag, 2010

[24] Sanjai Narain, Sharad Malik, Shuyuan Zhang. Planning Hybrid SDN and Legacy Networks. Open Networking Users Group Conference, Research Track, Columbia University, New York, May 14, 2015

[25] Sanjai Narain. BGP Stable Path Problem Specification in Alloy . Formal Methods in Networking Class, Princeton University, 2010

[26] Sanjai Narain. Moving-Target Defense by Configuration-Space Randomization. Filed 2014.

[27] Sanjai Narain. Network Configuration Management Via Model Finding. Proceedings of USENIX Large Installation System Administration (LISA) Conference, San Diego, CA, 2005. Also in Proceedings of ACM Workshop on Self-Managing Systems, Newport Beach, CA, 2004. Full report.

[28] Sharad Malik, Lintao Zhang. Boolean Satisfiability from theoretical hardness to practical success. Communications of the ACM. Volume 52 Issue 8, August 2009. http://goo.gl/2Grdy6

[29] Shuyuan Zhang, Abdulrahman Mahmoud, Sharad Malik. Verification and synthesis of firewalls using SAT and QBF. IEEE International Conference on Network Protocols, October 2012, Austin, TX

[30] Shuyuan Zhang, Laurent Vanbever, Sharad Malik, Sanjai Narain. In-Band Update for Network Routing Policy Migration. IEEE International Conference on Network Protocols, October 2014, Research Triangle, NC.

[31] Software Defined Networking. https://ee.stanford.edu/research/software-defined-networking

[32] Yices SMT solver. http://yices.csl.sri.com/

[33] Z3 SMT solver. https://z3.codeplex.com/

About the Author(s)

Sanjai Narain is a Fellow and Chief Scientist in Information Assurance and Security Department at Applied Communication Sciences (ACS). ACS was earlier called Telcordia Research and Bellcore Applied Research. Currently, he leads a Science of Configuration project. He has served on editorial boards and program committees of major journals, conferences and workshops. In 1990, he joined Bellcore where he developed SONET, ATM and DSL network management tools. From 1981 to 1990 he worked at RAND Corporation where he developed technologies to reason about discrete-event simulation models. His formal training is in mathematical logic, programming languages, and electrical engineering. He studied logic with Professor Alonzo Church at UCLA. He has obtained funding from government agencies such as AFRL, IARPA, DARPA and DHS. His current and past collaborators are **Professors Sharad Malik**, **Mung Chiang** and **Prateek Mittal** at Princeton, **Professor Trent Jaeger** at Penn State, Professor **Daniel Jackson** at MIT, **Professor Bart Selman** at Cornell, and **(Retd.) Col. Kevin Jordan** at PACOM.

RELEASE STATEMENT

Distribution A: Approved for Public Release;
Distribution Unlimited : 88ABW-2015 4645 20150929.

Networking Modeling and Simulation: Bridging the Gap from Theory to Field Tests

By Elizabeth Serena Bentley, Sunil Kumar, Joel Dallaire, and Jerry Reaper

The use of live field testing for new data links, protocols, waveforms, radios, and algorithms is the traditional best-practice. With that said, the cost, time, effort and complexity involved in large-scale field tests often makes these tests unaffordable to many levels of testers. A glaring example is the gap that exists between basic research algorithm development and the testing of those algorithms in operationally relevant environments, where the number of nodes extends into hundreds of individual systems and radios. Modeling, Simulation and Analysis (MS&A) bridges this gap, offering an attractive, cost-effective, and readily scalable means to evaluate newly developed theories and technologies with a full complement of scenarios not easily achievable through empirical methods. MS&A is a viable tool that helps gain insight into large-scale system performance in ways that are affordable at all levels of test

Inserting the proper MS&A steps into design processes of algorithms and technologies results in reduced cost, schedule, and risk for the systems and algorithms under development, the ability to test the operational characteristics of new systems and algorithms in complex environments, the capability of reconfiguring and customizing simulations for a wide variety of Air Force (AF)-relevant scenarios, large-scale systems and network models. This approach provides the methods needed to fine-tune systems and algorithms, and validate test plans before going out to the field to test. It also provides a central visualization for systems and networks spread over a wide geographical area that is not possible in empirical testing.

The work presented here was completed using Riverbed Modeler (formerly OPNET Modeler). Riverbed Modeler is a commercial discrete event simulator with a rich feature set that is used for the modeling, simulating, and analysis of communications networks, protocols, devices, and applications. The graphical editor interface encompasses a three-part modeling mechanism: the process model, the node model and the network model. The process model is a finite state machine that uses C programming to represent the process flow. There is a hierarchical network model structure that covers the Application, TCP, IP, MAC, and PHY layers. Riverbed Modeler includes an extensive range of protocols and standards with up to 400 libraries and standards implemented and specific model descriptions for popular networking equipment including operational parameters. There is also a variety of application models such as Poisson distribution, Binomial distribution, etc. Riverbed Modeler is well-suited for protocol and algorithm development, since each level of the model is completely customizable, allowing users to tailor their models to their specific needs and requirements for various topologies and scenarios.

Networking M&S in Riverbed Modeler

Two efforts in Riverbed Modeler will be discussed in this article: (A) a cross-layer ROuting and Spectrum Allocation (ROSA) implementation and (B) directional networking using multi-beam directional antennas.

A. ROSA

As spectrum becomes a scarce resource, cognitive radio networks, where secondary users can opportunistically access the spectrum without disrupting primary users' existing transmissions, are a feasible solution. AFRL/RI (the Air Force Research Laboratory, Information Directorate), in collaboration with the University at Buffalo, developed a cross-layer ROuting and dynamic Spectrum Allocation (ROSA) algorithm that maximizes the network throughput by performing joint routing, dynamic spectrum allocation, scheduling in a distributed way, and transmit power control. It utilizes distributed routing decisions based on the differential backlog and channel capacity to choose the optimal frequency, next hop, and the session to be transmitted to maximize the spectrum utility function.

Each node is assumed to be equipped with two transceivers. One channel is reserved for the time-slotted Common Control Channel (CCC), employed by all secondary users for negotiations for spectrum access. The second channel is used for Data Communication (DC). Handshakes on the two channels are performed independently, allowing parallelism. When the CCC is sensed to be idle, every backlogged node that is not already transmitting performs the ROSA algorithm as follows:

(i) Calculate the spectrum, corresponding transmit power, and capacity for each link for the set of next hops, which includes neighbors that are closer to the destination for the backlogged session. The spectrum and power allocation algorithm maximizes the capacity link which translates into selecting the spectrum and the corresponding transmit power on each frequency to maximize the Shannon capacity subject to the spectrum conditions (including the presence of a spectral hole) and the hardware limitations of the radio.

(ii) Schedule the session with the maximum differential backlog on that link with the next hop that maximizes the spectrum utility for that link. Calculate this maximum spectrum utility. This ensures that nodes with smaller backlogged queues with more spectrum receive more traffic.

(iii) Calculate the probability of accessing the medium based on this maximum spectrum utility on the next hop link. Links with a higher differential backlog may have a higher spectrum utility. This leads to having a higher probability of being scheduled for transmission. This probability is implemented by varying the size of the contention window at the media access control (MAC) layer. The transmitter generates a back-off counter that is uniformly chosen from a range that depends on the contention window of that transmitter. Nodes with smaller back-off counters will have higher priorities in allocating resources, so heavily back-logged links with more spectrum are given a higher probability of transmitting.

The steps listed above are fundamental steps of the ROSA algorithm, provide a general idea on what the implementation of the algorithm entails, and were taken from [1] where the full theoretical explanation can be found.

The MAC layer for ROSA is similar to the IEEE 802.11 two-way request-to-send (RTS) and clear-to-send (CTS) handshake with an additional control packet, the data transmission reservation (DTS), that is used by the transmitter to announce its spectrum reservation and its transmit power to its neighbors. To allow nodes to learn about the spectral nature and the queue length information of their neighbors, the RTS/CTS/DTS packets are modified to include the address fields of the sender and receiver, spectrum reservation, reservation duration field, queue length information, and power constraints.

The performance of ROSA was compared with two alternatives which rely on the same knowledge of the environment. Routing with Fixed Allocation (RFA) uses routing based solely on the differential backlog with a predefined channel and transmit power. Routing with Dynamic Allocation (RDA) uses routing based on the shortest path (with no consideration of the differential backlog,) with dynamic channel selection and transmit power allocation [2]

The Riverbed Modeler ROSA model was designed with several goals in mind. First, the model needed to be flexible with regard to the underlying RF parameters. It should be able to support the representation of a generic radio, the USRP radios used by AFRL/RI, and future radios based on existing military data links. Each of these could have different frequencies, bandwidths, powers, modulation, data rates, etc., and the model must not be hard-coded to work with fixed values for any of these parameters. Another design goal was to implement the core ROSA algorithm such that it could be used outside of Riverbed Modeler. This allows the algorithm to be developed and tested even without a Riverbed Modeler license available, testing to be performed outside the constraints of a Riverbed Modeler simulation, and for the algorithm code to potentially be used in other models or even an actual radio.

The remainder of this paper presents the implementation of ROSA in Riverbed Modeler.

The application layer is modeled as simple traffic generators with fixed rates, fixed-sized packets, and either a fixed, single destination or randomized source-destination pairs. A standard Riverbed Modeler sink is used to record receive statistics and destroy the packet. Packet streams connect the application to the DC and the DC to the sink. Node addresses can be set manually or be automatically assigned. The sink provides statistics on end-to-end delay (seconds) and the traffic received in bits, bits/sec, packets, and packets/sec.

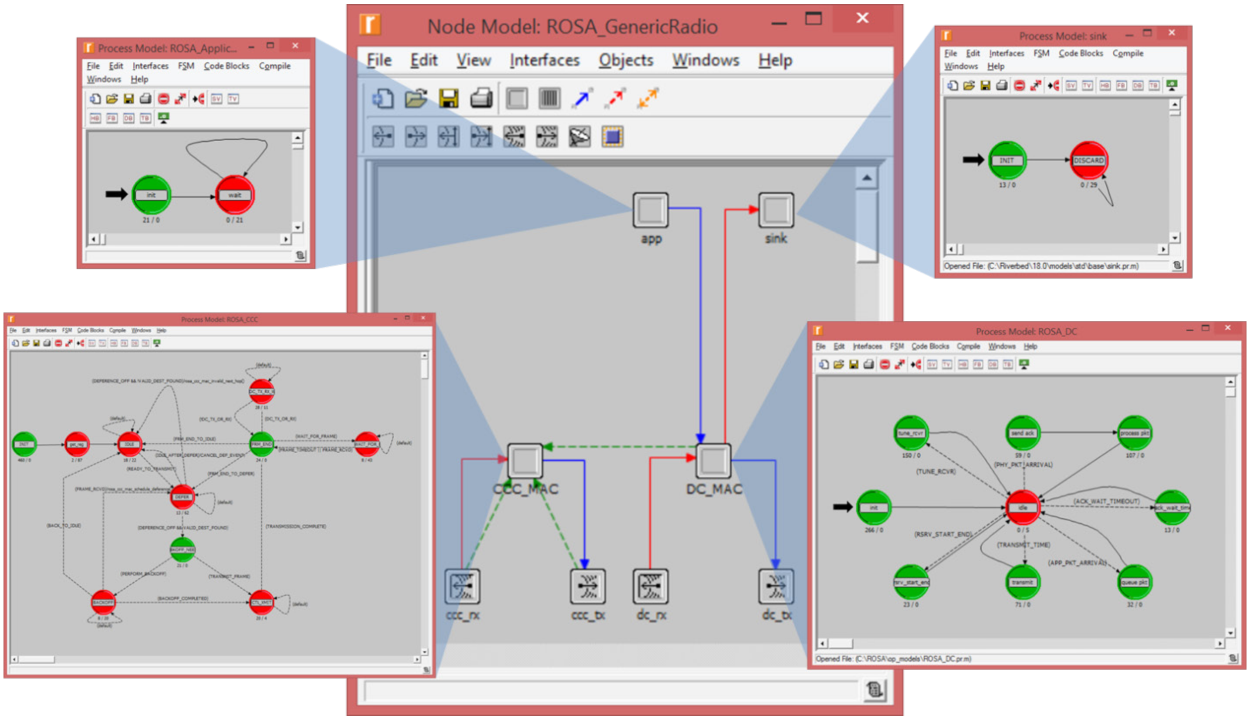
The MAC layers are where the ROSA algorithm resides. The node model of ROSA and the process models of the CCC MAC, DC MAC, application and sink are shown in Figure 1. The CCC is responsible for spectrum reservation negotiations between nodes, storing neighbor information used by the algorithm, calling external C++ code to execute the ROSA algorithm to determine its next hop and RF transmission parameters and beginning the RTS/CTS/DTS handshaking to acquire the data channel reservation. Once acquired, it passes the RF parameters and session to transmit to the DC layer to start the data packet transmissions. The CCC provides reservation information for both the transmit and receive to the DC using a Remote Interrupt with Event State. The DC queues packets from the application, transmits when directed by the CCC, decapsulates the received physical layer packets, forwards to the sink (if for this node), queues routed packets, and provides session backlog information to the CCC.

Statistic wires from the transmitter and receiver to the CCC MAC indicate when the CCC transmitter stops transmitting, when the receiver becomes busy, and when the receiver is no longer busy. A statistic wire from the DC MAC indicates when a packet is added to an empty queue (first active session) or when the queue becomes empty (no active sessions).

The RTS/CTS/DTS packet formats and the data/ACK packets were based on Riverbed Modeler 802.11 formats and customized to include all information required by the ROSA algorithm. The packet formats for the data and control packets are shown in Figures 2 and 3.

To support a generic ROSA radio, many of the CCC and DC transmitter and receiver parameters are set based on global attribute settings. The settings include maximum transmitter power, base frequency, bandwidth of each mini-band, the maximum number of mini-bands, and data rate. During a simulation, as the ROSA algorithm executes and different DC power and frequencies are selected, the DC transmitter and receiver parameters are modified. Additionally, a packet field for the data rate is added to the packet to support variable data rates. This field is used in the radio transceiver pipeline stages. The ROSA model pipeline stages are largely based on Riverbed's WLAN (802.11) implementation, though some are based on the Riverbed's default stages. Using the WLAN stages provided to the ROSA model support for handling jamming and using the data rate packet field.

The global attributes also include a mode field that changes the routing and spectrum allocation algorithm to use. Possible values are ROSA, RFA and RDA. The latter two were implemented in Riverbed Modeler as well, for comparison with ROSA.

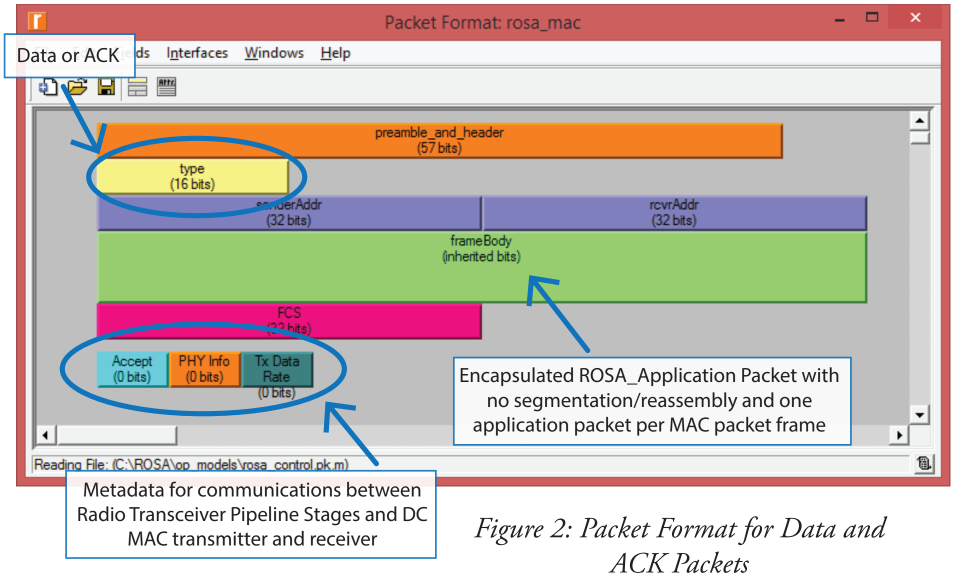


Opened File: (C:\Riverbed\18.0\models\std\base\sink.pr.m)

Opened File: (C:\ROSA\op_models\ROSA_DC.pr.m)

Figure 1: Riverbed Modeler ROSA Node Model with the Process Models

In the future, AFRL/RI will examine how well ROSA would perform in contested environments where undesired emitters may block certain frequencies from usage. The Riverbed Modeler ROSA model and the Stockbridge Controllable Contested Environment (CCE) test site (that incorporates terrain effects) will be used together to guide the design of the AFRL field test experimentation plan. This test plan will be executed as the existing hardware lab set-up, with the ten USRP N210s + SBX/CBX daughterboards shown in Figure 4, is moved to the Stockbridge CCE test site. The existing hardware testbed implements the ROSA algorithm using RTS/CTS/DTS control packets, a wired CCC Collaborative Virtual Sensing (CVS) where nodes obtain spectrum information based on a combination of physical sensing and local exchange of information, neighbor discovery via beacon packets, and the RDA/RFA alternatives for comparison. Analysis in Riverbed Modeler will consist of examining how ROSA performs as the number of nodes increases, with different propagation effects, with various jamming scenarios, and with node mobility. Newly developed neighbor discovery methods in GPS-denied environments and methods to make the CCC more robust in contested environments will also be tested and analyzed in Riverbed Modeler. The use of the ROSA model provides a means to affordably complete a thorough analysis of the strengths of ROSA, as well as determining any breaking points, in a wide variety of geometries, environments, topologies, and traffic loads.

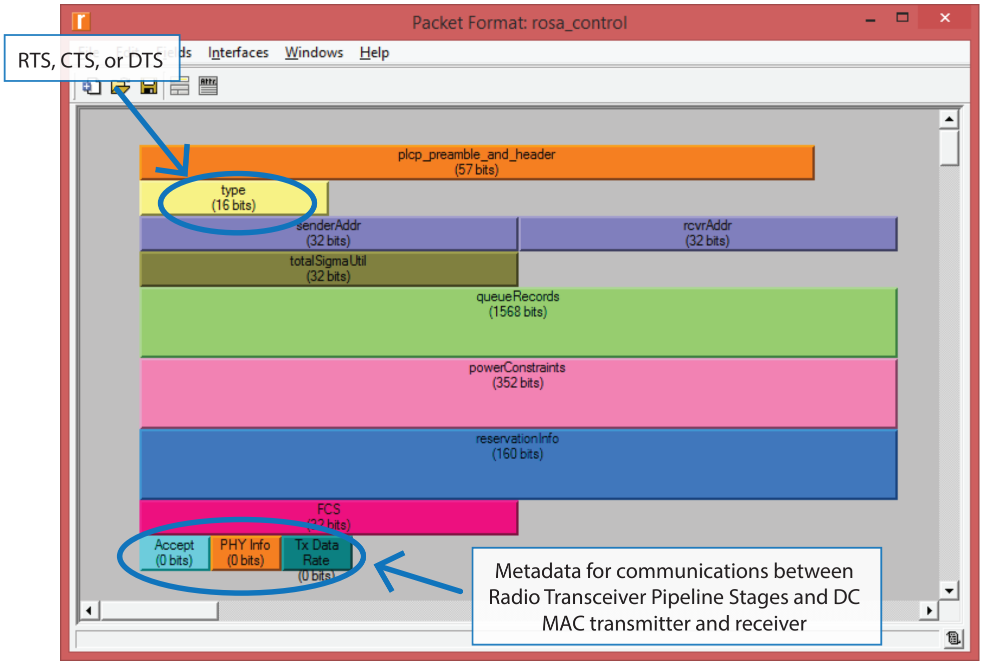


Data or ACK

Encapsulated ROSA_Application Packet with no segmentation/reassembly and one application packet per MAC packet frame

Metadata for communications between Radio Transceiver Pipeline Stages and DC MAC transmitter and receiver

Figure 2: Packet Format for Data and ACK Packets



RTS, CTS, or DTS

Metadata for communications between Radio Transceiver Pipeline Stages and DC MAC transmitter and receiver

Figure 3: Packet Format for Control Packets

B. Multi-beam Directional Antennas and Directional Networking

Traditionally, medium access control (MAC) protocols are designed for nodes, which are equipped with omni-directional antennas. Some disadvantages of using omni-directional antennas are poor data throughput, lower network and power efficiency due to interference resulting from the transmission

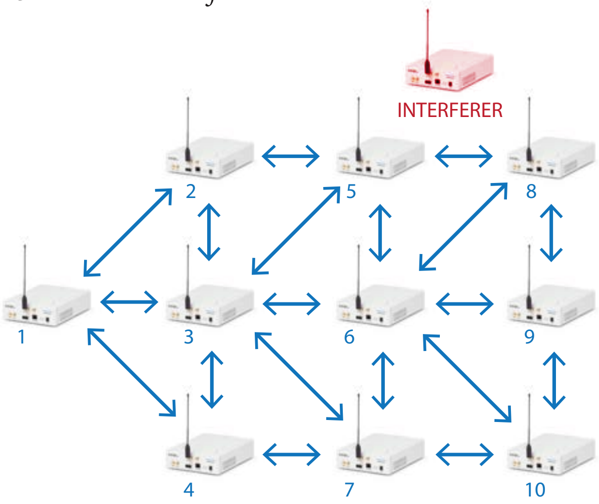


Figure 4: ROSA hardware testbed

of packets in undesired directions. Using directional antennas and the concept of sending packets in specific directions has a significant impact on spectrum efficiency and power consumption. Using multiple RF chains at each antenna element, separate beams can be formed simultaneously. The use of beamforming directional antennas capable of adaptively configuring multiple narrow beams and nulls enables *(i)* multiple simultaneous directional transmissions, *(ii)* spatial isolation and frequency reuse, *(iii)* LPD/LPI qualities since transmissions are not broadcast in an omni-directional manner for anyone to hear/intercept, *(iv)* large data rates, *(v)* long transmission ranges, *(vi)* graceful degradation with failure and combat damage, *(vii)* interference avoidance, and *(viii)* anti-jam capabilities for contested environments by null formation in the direction of jammers and other undesired emitters.

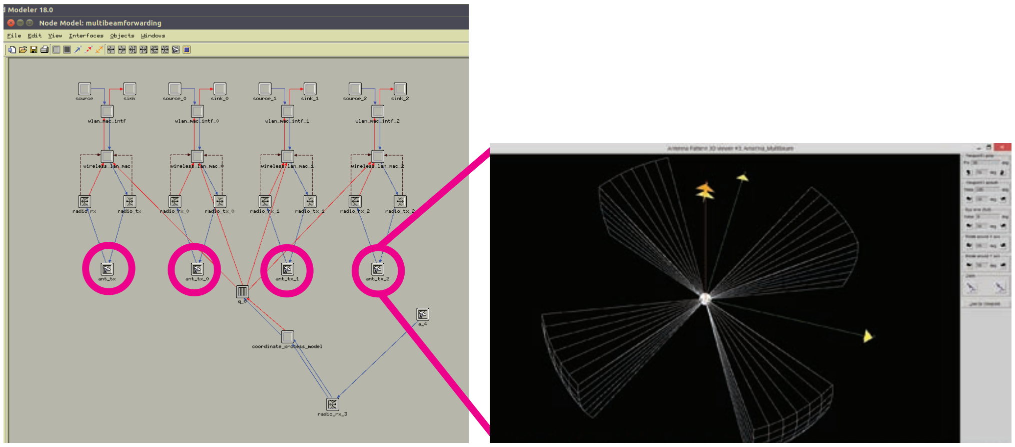


Figure 5: Multi-beam Directional MAC with multi-beam directional antenna with 4 beams

However, there are challenges associated with the use of directional antennas that require more investigation. Using a directional antenna (i) requires the design of neighbor and topology discovery techniques for mobile nodes, (ii) introduces new hidden node problem due to node deafness, and (iii) requires distributed scheduling schemes to avoid performance degradation due to hidden nodes. Since antenna direction also impacts the routing path, a cross-layer design among routing-MAC-PHY is required.

Most of the published literature on directional antennas assumes a selectable main beam gain and no side lobe interference irrespective of beam width and number of beams, assumes non-overlapping beams, ignores range extension due to different beams widths, and does not consider terrain effects or contested environments. Riverbed Modeler is grounded in the math and physics of antenna and channel propagation. This makes the model results one step closer to reality because Riverbed Modeler prevents users from arbitrarily selecting the main-beam gain while ignoring the side lobe interference. Likewise, Riverbed Modeler correctly calculates the impact of beam width and number of beams on the transmission range, allows for importing a user-defined antenna gain table, and incorporates terrain and other propagation effects. Through preliminary comparative testing, AFRL/RI, in collaboration with San Diego State University (SDSU), has found that Riverbed Modeler simulations with directional antennas are far more realistic than the Matlab and ns-2 based simulations.

While Riverbed Modeler allows the use of directional antennas at the PHY layer, the implemented MAC protocols use omni-directional antennas. Consequently, the implementation of directional protocols required significant modifications to the existing Riverbed Modeler code. For the IEEE 802.11 MAC layer protocol, omni-directional RTS/CTS/Data/ACK were modified to omni-directional (or directional) RTS, omni-directional (or nearly omni-directional) CTS, and directional Data/ACK. This required writing additional code in the MAC process model and including cross-layer interaction between the PHY and MAC processes that did not exist in the standard 802.11 models. Further, the use of a multi-beam antenna equipped node required additional code for extending the existing omni-directional (or single beam) nodes in Riverbed Modeler. Since multi-beam nodes can simultaneously transmit and/or receive data on each of their beams, concurrent packet transmissions and receptions have been implemented, requiring additional code for simultaneous (or synchronized) RTS, CTS, Data and ACK, including addressing node back-off issues and alternate packet transmission-reception in relay nodes. The node model of a multi-beam antenna with four simultaneous beams and the multi-beam directional MAC is shown in Figure 5.

Currently, Riverbed Modeler uses a video data model where a fixed number of packets per second (corresponding to a given video bit rate and packet size) are assumed. Additional code was written to transmit H.264/AVC video bitstreams. H.264/AVC video has a variable number of packets, varying packet sizes, and a packet loss distortion value in each unit time. This enables the examination/comparison of performances of different protocols and network topologies for the compressed full motion videos.

In the future, Riverbed Modeler will be used to implement the HMAC cross-layer protocol which allows for simultaneous transmission and reception of multiple packet on different beams, to design a MAC scheme for concurrent transmission and reception using nodes equipped with multi-beam antennas, to develop a new reactive routing protocol for finding multiple paths between source-destination pairs using multi-beam antennas in wireless mesh network architectures for comparing the new protocol's performance with existing Riverbed Modeler protocols, to design a new MAC protocol to study the performance of TCP for long-distance links, and to examine protocol performance with respect to mission effectiveness in AF relevant environments containing high speed links, long distance links, interference, high node mobility, and jammers.

Conclusions

This paper has presented two efforts in networking MS&A. The discussion began with the basic theory, continued through the current implementations, and finished with planned future steps. As noted, planned laboratory and field tests will utilize this modeled performance analysis to reduce costs and increase to scope of available testing. The major components of a joint routing and spectrum allocation algorithm, ROSA, were presented, as well as current test results demonstrating that ROSA possesses high throughput, low delay, and fair bandwidth allocations in dynamic, ad-hoc networks., Finally, this paper presented the initial steps for developing a multi-beam directional antenna and networking capability in Riverbed Modeler. ✈

References

[1] http://www.riverbed.com/products/steelcentral/steelcentral-riverbed-modeler.html

[2] L. Ding, T. Melodia, S. Batalama, j. Matyjas and M. Medley, "Cross-Layer Routing and Dynamic Spectrum Allocation in Cognitive Radio Ad Hoc Networks", in *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, May 2010, pp. 1969-1979.

About the Authors

Elizabeth Serena Bentley has a B.S. degree in Electrical Engineering from Cornell University, a M.S. degree in Electrical Engineering from Lehigh University, and a Ph.D. degree in Electrical Engineering from University at Buffalo. She was a National Research Council Post-Doctoral Research Associate at the Air Force Research Laboratory in Rome, NY. Currently, she is employed by the Air Force Research Laboratory in Rome, NY, performing in-house research and development in the Networking Technology branch. Her research interests are in cross-layer optimization, wireless multiple-access communications, wireless video transmission, modeling and simulation, and directional antennas/directional networking.

Sunil Kumar received his Ph.D. from the Birla Institute of Technology and Science, Pilani (India) in 1997. Currently, he is a Professor and Thomas G. Pine Faculty Fellow in the Electrical and Computer Engineering department at San Diego State University, CA, USA. His research interests include robust video compression (including H.264 and HEVC), and QoS-aware and cross-layer protocols for wireless networks. He has published 130 research articles in refereed journals and conferences, seven books and book chapters, and recently co-edited two books. His research has been funded by the U.S. NSF, DOD, DOE, and California Energy Commission.

Joel Dallaire is a senior engineer with Clear Creek Applied Technologies. Mr. Dallaire has over 17 years of experience in modeling and simulation, system software, and enterprise software development. He has a B.S. in Computer Science from Wright State University.

Jerry Reaper is a senior engineer with Clear Creek Applied Technologies with over 35 years of experience in applying modeling, simulation and analysis to address technical questions and extend available testing and analysis avenues.

RELEASE STATEMENT

Distribution Statement A: Approved for Public Release;
distribution unlimited: 88ABW-2015-4795

Bridging Fault Tolerance and Game Theory for Assuring Cyberspace

By Kevin Kwiat & Charles Kamhoua, AFRL/RIGA

Two Air Force Office of Scientific Research (AFOSR)-funded in-house efforts have shaped the way that AFRL/RI has bridged fault tolerance and game theory: "Fault Tolerance for Fight-Through (FTFT)" and "STORM: Survivability Through Optimizing Resilient Mechanisms". FTFT was the forerunner of STORM. This was also a logical ordering from a historical perspective because fault tolerance is an older discipline than game theory. Fault-tolerant computing formally originated when John von Neumann introduced the concept to electronic computers. Although the introduction of game theory is also credited to von Neumann, it was much earlier, in 1837, that Charles Baggage gave evidence of fault tolerance's existence. In [1], he wrote that a complicated formula could be algebraically arranged in several ways such that if the same values are assigned to the variables and the results agree, then the accuracy of the computation is secure. Babbage, of course, was referring to the work of clerical staff – the "computers" of his time. Note that Babbage advocated the use of diversity to secure a computation [1]. As digital computers developed, diversity became a key consideration when seeking fault tolerance, and throughout the history of computers, fault tolerance was often coupled with diversity for added assurance to computing [2-3]. In FTFT we used fault tolerance and diversity to address the more contemporary concern of cyber defense.

Fault–tolerant computing shares conceptual similarities with cyber defense. For example, fault tolerance deals with the detection and treatment of failures whereas cyber defense deals with the detection and treatment of compromises – both of which can cause a computer to deviate from its specification. Traditionally, fault-tolerant computing dealt with deviations stemming from randomly occurring faults and not faults resulting from intelligent attack. Whereas faults caused by natural-occurring phenomena are tolerable using established, standard approaches, attacker-induced faults require a more aggressive approach that also ushers-in cyber defense. New challenges arise in the area of transforming fault tolerance to attack tolerance. As information systems become ever more complex and the interdependency between these systems increases, it is beyond the abilities of most system developers to predict or anticipate every type of component failure and cyber-attack. Attempting to predict and protect against every conceivable failure and attack soon becomes exceedingly cumbersome and costly. Therefore, the more realistic goal became the design of a fight-through capability that can absorb the damage and then rebound so that it can be the basis for restoration of critical services. We sought adaptations of fault-tolerant computing concepts to address this need in cyber defense. An optimum decision has to be made early in the design phase and during mission execution to maximize fault-tolerance. To achieve that, we found an appropriate source in military strategist John Boyd who conceived and developed the Observe, Orient, Decide, and Act Loop (OODA Loop) [4]. He applied the OODA Loop to the combat operations process including the engagement of fighter aircraft in aerial combat. Figure 1 shows a basic OODA Loop.

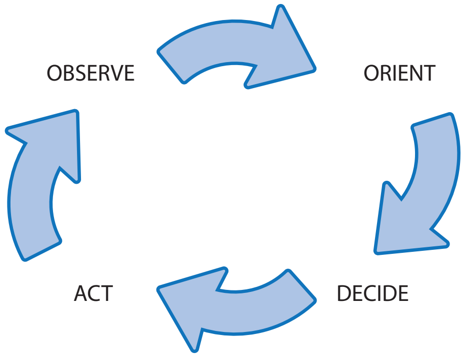


Figure 1: A Basic Observe, Orient, Decide and Act (OODA) Loop

We saw a close similarity between the fight through problem and the OODA loop that allowed us to create a fight-through OODA loop. The fight-through OODA loop [6] shown in Figure 2 is aimed at outperforming the adversary's OODA loop. It absorbs the damage inflicted by the attacker to ultimately prevail. The mainstay for our fight-though OODA loop is fault tolerance. Invariably, fault tolerance calls upon some form of redundancy. Spatial, temporal, and information redundancy [5] are stood-up concurrently so that the loss of resources in one dimension is withstood by the other dimensions. These dimensions of redundancy supply the reinforcing resources for a fight-through capability. FTFT spans multiple dimensions of redundancy to form an OODA loop for fighting-through. Redundancy, as the underpinning of fault tolerance, is not placed haphazardly; instead, redundancy is strategically placed to counter the attacker. When computer replication is employed as a form of redundancy it is infused with diversity so that the replicas would be functionally-equivalent but present the attacker with different targets. Virtual machines in a cloud computing environment are a contemporary source of viable spatial replication. Such replication can overwhelm an attacker with too many targets; however, replication is more than merely providing sacrificial targets. By being able to observe an attacker's actions aimed at depleting the number of replicas, the fight-through OODA loop can: orient the other replicas; decide on their deployment; and then act against the attack. The fight through OODA loop in Figure 2 depicts such a scenario. It shows concentric loops. The attacker's outer loop strives to compromise those replicas that comprise a critical application by monitoring the replicas' communications. Similarly, the defender's inner loop has the replicas monitoring their own communications. However, the inner loop decides when the information divulged by the replicas' communications is approaching a critical level. Before the critical level is reached, the communicating replicas agree to change roles. This disrupts the previous communication pattern such that the knowledge that the attacker had derived from it is now seriously diminished. The tighter diameter of the defender's OODA Loop illustrates the defender's more timely completion of the cycle. FTFT's multi-dimensional redundancy permits the defender to "get inside the enemy's decision cycle."

In a larger sense, our transformation of fault-tolerant computing concepts into a fight through capability is a blending of the reactive and the proactive: our proposed fight through OODA Loop is reactive to faults yet strives to proactively anticipate the attacker's next action.

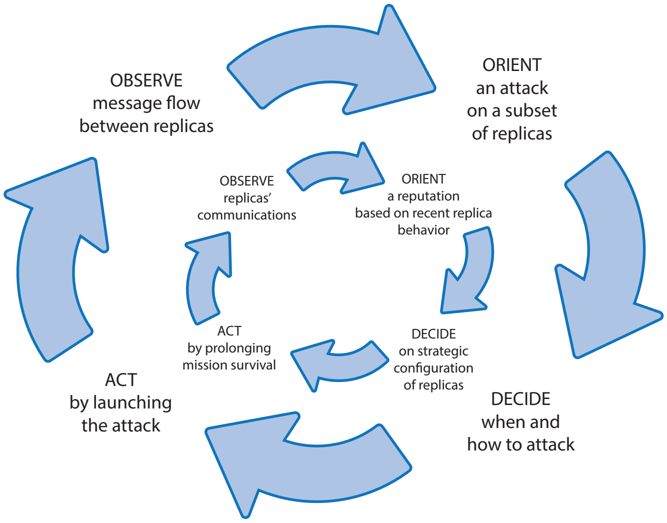


Figure 2: Concentric OODA Loops: Attacker's Outer Loop, Defender's Inner Loop

The FTFT effort investigated models, algorithms, and protocols to support the creation of an OODA loop for fighting-through [6-7]. An important step forward for FTFT became in-depth strategic consideration of cyber conflict. FTFT provided a fight through mechanism, but a mechanism, however, is merely a trigger; procedures must be used in conjunction with the mechanism to face the attack more strategically. For these procedures we turned to game theory.

Game theory is the branch of applied mathematics [8] that analizes conflict and strategic interactions among intelligent rational agents. With such a broad scope, game theory became syngistic with a contested cyberspace. For example, game theory has been applied to network security [9-10]. The synergy we observed compelled us to investigate a game theoretic framework and bridging it with fault tolerance. The most dangerous system failures typically originate from intelligent attackers instead of random faults, and game theory enables modelling the behavior of intelligent adversaries. Thus, a game theoretic model, if properly applied, might be the best one to deal with the worst case scenario, i.e., an inteligent attacker with detailed knowledge of the system. Furthermore, we believe that game theory is a promising framework because game theory has had marked success for over six decades in modeling other complex systems such as economics and biology. Finally, to the best of our knowledge, a game theoretic modeling of fault tolerance capabilities for cyber assurance was a new and open problem.

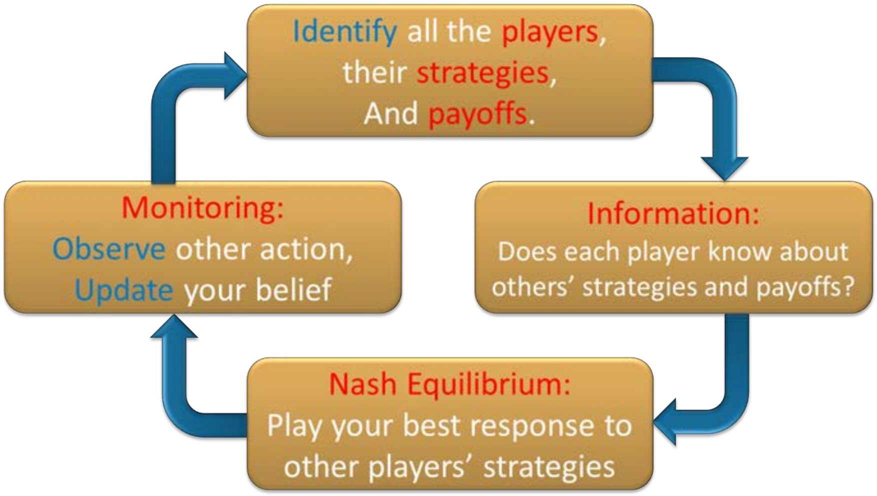

Identify all the players,
their strategies,
And payoffs.
Information:
Does each player know about others' strategies and payoffs?
Nash Equilibrium:
Play your best response to other players' strategies
Monitoring:
Observe other action,
Update your belief

Figure 3: Game Theory Decision Loop

A strategic interaction is any interaction in which the behavior of one agent affects the outcome of others. First, the optimum defensive strategy should depend on the attacker's behavior. Second, several protocols and security policies, including diversity, cannot be unilaterally implemented. Cyber diversity, like numerous other protocols, requires the collaboration of several users in several organizations in order to be successful. Finally, cyberspace is interconnected and the data collected from one vulnerable computer can be used to compromise others. Using the framework of game theory, the cyber defender has a path to optimize his resources and defensive strategy while simultaneously taking into account those actions from other users including the attackers. A small sampling of our early papers [11-13] documents some of our accomplishments in using a game theoretic framework to embrace fault tolerance and diversity. Most recently, we have used the framework for assuring cloud computing [14-17].

A key component of game theoretic modeling of cybersecurity is to find the Nash equilibrium of the cybersecurity game. At a Nash equilibrium profile, no player's payoff is increased by a unilateral deviation. Also, each player is playing their best response to other players' strategies. As a consequence, the cyber defender can use the Nash equilibrium profile to predict the attacker's behavior. These actions are depicted in the decision loop of Figure 3 whose 4 stages are analogous to those depicted in the loops of Figures 1 and 2.

The STORM effort aims to capture the mechanisms that move this loop. With the ability to control this loop, STORM strives to develop dynamic and unpredictable schemes which, like a storm, can disrupt the adversaries' plans and advances. Strategically, by storming the attacker in this way we embrace Boyd's notion of a best stragtegy: to win without ever engaging in a fight at all [18].

John Nash's famous proof that there is an equilibrium for every finite game is a strong motivation to adopt game theory: once obtained, an equilibrium brings a "stategic pause" to the continuous revolutions depicted in the loops of Figures 1-3. Without such a pause a loop can become like a vortex continually drawing in resources – presenting a challenge to not only sustaining the loop but to the mission itself. Instantiating these loops becomes an engineering enterprise that calls upon sound judgement of the human, software, hardware and communciations resources required to execute them. They are strategic loops, so they too are the outcome of a strategy. Our formation of this underlying strategy calls upon this fact: is not unknown in war for a side to win every battle, but, through flawed strategy, to lose the war [18]. Therefore, the building of our bridge between fault tolerance and game theory spans 1) discovering the most promising strategy and 2) applying the engineering principles so that even if a fault occurs, the strategy does not become flawed. These discovery and engineering processes for assuring cyberspace continue with STORM.

References:

[1] Babbage, C., "On the Mathematical Powers of the Calculating Engine," (Unpublished Manuscript) Buxton MS7, Museum of the History of Science, Oxford, December 1837, Printed in *Origins of Digital Computers: Selected Papers*, B. Randell (ed), Springer, 1974, pp. 17-52.

[2] Avizienis, A., Kelly, J. P. J. "Fault Tolerance by Design Diversity: Concepts and Experiments," Computer, vol. 17, no. 8, pp. 67-80, August 1984.

[3] Avizienis, A., Laprie, J.-C., "Dependable Computing: From Concepts to Design Diversity," IEEE, vol. 74, no. 5, pp. 629-638, May 1986.

[4] Frans P.B. Osinga, Science, Strategy and War: The Strategic Theory of John Boyd, Routledge Publishing, 2006.

[5] Johnson, B., *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley, 1989.

[6] *Fault Tolerance for Fight Through (FTFT)*, AFRL Final Report, AFRL-RI-RS-TR-2013-039, February 2013.

[7] Kwiat, K., "Fault Tolerance for Fight-Through: A Basis for Strategic Survival," Proceedings of the ACM 4th International Conference on Security of Information and Networks (SIN) held in Sydney, Australia, November 2011.

[8] Myerson, R. "Game theory: analysis of conflict", Harvard University Press, 1997.

[9] Roy, S. Ellis, C. Shiva, S. Dasgupta, D. Shandilya, V. Qishi, W. "A Survey of Game Theory as Applied to Network Security", 43rd Hawaii International Conference on System Sciences (HICSS). Honolulu, HI, USA. March 2010.

[10] Alpcan, T. and Basar T. "Network Security: A Decision and Game-Theoretic Approach", Cambridge University Press; 1 edition (November 30, 2010)

[11] Kamhoua, C., Kwiat, K., Chatterjee, M., Park, J., and Hurley, P., "Replication and Diversity for Survivability in Cyberspace: A Game Theoretic Approach," in Proceedings of the International Conference of information Warfare (ICIW 2013) Denver, Colorado, USA, March 2013.

[12] Kamhoua, C., Kwiat, K., and Park, J., "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.

[13] Kamhoua, C., Hurley, P., Kwiat, K., and Park, J., "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.

[14] Kamhoua, C., Kwiat, L., Kwiat, K., Park, J., Zhao, M., Rodriguez, M., "Game Theoretic Modeling of Security and Interdependency in a Public Cloud" in the proceedings of IEEE International Conference on Cloud Computing, (IEEE CLOUD 2014) Anchorage, Alaska, June 2014.

[15] Kwiat, L., Kamhoua, C., Kwiat, K., Tang, J., and Martin, A., "Security-aware Virtual Machine Allocation in the Cloud: A Game Theoretic Approach" in the proceedings of IEEE International Conference on Cloud Computing, (IEEE CLOUD 2015) New York, New York, June-July 2015.

[16] Kamhoua, C., Martin, A., Tosh, D., Kwiat, K., Heitzenrater, C., Sengupta, S., "Cyber-threats Information Sharing in Cloud Computing: A game Theoretic Approach" in the proceedings of the IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2015), New York, November 2015.

[17] Kamhoua, C., Ruan, C., Martin, A., Kwiat, K., "On the Feasibility of an Open-Implementation Cloud Infrastructure: A Game Theoretic Analysis" in the proceedings of the 2015 IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2015), Limassol, Cyprus, December 2015.

[18] Richards, C., *Certain to Win: The Strategy of John Boyd Applied to Business,* Xlibris, 2004.

About the Authors

Kevin A. Kwiat is a Principal Computer Engineer with the Air Force Research Laboratory's Cyber Assurance Branch. He received the B.S. in Computer Science and the B.A. in Mathematics from Utica College of Syracuse University, and the M.S. in Computer Engineering and the Ph.D. in Computer Engineering from Syracuse University. His main research interest is dependable computer design.

Charles A. Kamhoua received his B.S. in Electronic from the University of Douala (ENSET), Cameroon in 1999, and the M.S. in Telecommunication and Networking and PhD in Electrical Engineering from Florida International University in 2008 and 2011 respectively. In 2011, he joined the Cyber Assurance Branch of the U.S. Air Force Research Laboratory (AFRL), Rome, New York, as a National Academies Postdoctoral Fellow and became a Research Electronics Engineer in 2012. Prior to joining AFRL, he was an educator for more than 10 years. His current research interests cover the application of game theory and mechanism design to cyber security and survivability, with over 50 technical publications in prestigious journals and International conferences including a Best Paper Award at the 2013 IEEE FOSINT-SI. Dr. Kamhoua has been recognized for his scholarship and leadership with numerous prestigious awards including 15 Air Force Notable Achievement Awards, the 2015 AFOSR Windows on the World Visiting Research Fellowship at Oxford University, UK, an AFOSR basic research award of $645K, the 2015 Black Engineer of the Year Award (BEYA), the 2015 NSBE Golden Torch Award - Pioneer of the Year, a selection to the 2015 Heidelberg Laureate Forum, a 2011 NSF PIRE award at Fluminense Federal University, Brazil, and the 2008 FAEDS teacher award. He is an advisor for the National Research Council, a Senior Member of IEEE, a member of ACM, the FIU alumni association, and NSBE.

RELEASE STATEMENT

Approved for Public Release; Distribution Unlimited: 88ABW-2015-5203 20151026

Physics of Information Assurance

By Donald Telesca, Ph.D., AFRL/RITB

Information Assurance (IA) is the application of this directive in the cyber domain. IA activities include measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. IA is the practice of assuring information and managing risks related to the use, processing, storage, and transmission of information or data and the systems and processes used for those purposes.[2] It can use physical, technical and administrative controls to accomplish these tasks.

In accordance with this directive, a principal responsibility of a commander is to assure mission execution in a timely manner. The reliance of a Mission Essential Function (MEF) on cyberspace makes cyberspace a center of gravity an adversary may exploit and, in doing so, enable that adversary to directly engage the MEF without the employment of conventional forces or weapons.

Joint Publication 1-02, DoD Dictionary of Military and Associated Terms, defines cyberspace as "a global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers," and cyberspace operations as "the employment of cyber capabilities where the primary purpose is to achieve military objectives or effects in or through cyberspace. Such operations include computer network operations and activities to operate and defend the Global Information Grid."

The U.S. Department of Defense (DoD) depends increasingly on cyberspace to execute critical missions that are vital to maintaining American military superiority in the traditional domains of land, sea, air, and space. The U.S. is arguably more at risk to an asymmetric attack vector launched by an adversary that cannot, or chooses not to, confront the U.S. in a conventional conflict. In the end, the military advantages that net-centricity provides the U.S. military concomitantly offer an adversary affordable attack vectors through cyberspace against critical missions and advanced weapon systems.

The objective of this work is to explore novel, promising ideas and methodologies in nano-scale hardware devices that address the inherent insecurity in cyberspace and increase information assurance in DoD cyber systems. This work will first identify the most suitable material stack for pursuing future memristor device technologies. This will be accomplished by fabricating memristor devices with differing material stacks and comparing their memristance performance, both endurance and resistance drift, when subjected to variable temperature operational conditions. Additionally, room temperature comparisons of the effects of total ionizing dose on device performance will also be assessed. Second, a standardized method to define Physically Unclonable Function (PUF) quality and a broadly accepted computation standard for PUFs will be identified. The objective of this work is To explore

The security primitives, Physical Unclonable Functions (PUFs) and True Random Number Generators (TRNGs) are pieces of the overall security and trust puzzle. Physical Unclonable Functions have been used to mitigate a variety of potential threats and attacks including integrated circuit (IC) piracy, counterfeiting, malicious Trojan insertion and side-channel analysis and random number generators are needed for cryptographic applications. Memristors have security relevant characteristics that make them a logical future foundation to generate these primitives will be the next step. In addition, previous research has shown that memristors are more difficult to reverse engineer and is tamper evident.

The security characteristics that may be leveraged have been summarized by AFRL researchers in published work[3] and are listed below.

(1) Non-volatility: Memristors retain their memristance value even when the power is turned OFF.
(2) Bi-directionality: Some bipolar memristors exhibit similar current-voltage characteristics irrespective of the polarity of the applied voltage or current.
(3) Non-linearity: The I-V characteristics of memristors are highly non-linear due to their time-dependent behavior. Also, the High Resistance State (HRS) to Low Resistance State (LRS) ratio is typically on the order of 10^3-10^6.
(4) Formation process: For many memristors, a separate forming step (V_f) is required to initialize the memristor to the LRS. Prior to this point, the memristor behaves as a linear resistor.

(5) Memristance drift: On applying an input voltage (positive or negative) across certain metal-oxide memristors, the memristance changes because of the movement of dopants, a process called memristance drift. The amount of drift depends on the polarity, amplitude, and duration of the applied voltage.

(6) Process variations: The memristance of a memristor is affected by process-variation induced changes in its dimensions and dopant concentration. Furthermore, the effects of variation in the thickness of the memristor upon its memristance values are highly non-linear (more significantly for the LRS than the HRS).

(7) Radiation-hardness: Some memristor devices are inherently radiation-hard due to their material properties.

(8) Temperature stability: The LRS and HRS values are highly stable in the case of a TiO_2 memristor since the temperature coefficient of resistance for TiO_2 is very small (less than -3.82×10^{-3}/K). However, the switching speed of the memristor varies with temperature because of the change in dopant atom mobility.

All of these characteristics with the exception of non-volatility and radiation-hardness pose problems when designing memory and logic circuits using a metal-oxide memristor, but can be useful in the context of security. It is for these reasons this work will identify, from the myriad of choices in materials, the most suitable material stack for focusing future memristor device research and technologies. This will be accomplished by fabricating memristor devices with differing material stacks and comparing their memristance performance, both endurance and resistance drift, when subjected to variable temperature operational conditions. Additionally, room temperature comparisons of the effects of total ionizing dose on device performance will also be assessed.

A complementary research initiative that will leverage the results of the memristor research described above will result in a more fundamental understanding of the requirements for designing a robust PUF system.

A PUF can be described as a fingerprint that can be used to uniquely identify individual integrated circuits (ICs). PUFs are unique in that no two devices will have the same signature and are unclonable due to the inherent infeasibility required to create two devices with the same signature. In the literature, both uniqueness and unclonability have been attributed to intrinsic variations resulting from non-uniform manufacturing process. There is variability in the complex physical processes associated with IC design and manufacturing. This creates a natural defense to an attacker whom now must either control the noise, or selectively and predictively change manufacturing parameters without disrupting the functional correctness of the resulting ICs. This portion of research will seek to gain a better understanding of whether these fundamental assumptions are valid, and formalize standards that define what makes a suitable PUF.

In order to accomplish this, existing device types used for PUF designs will be reviewed and key criteria that are important in the design of a robust PUF such as variability, uniqueness, unclonability and stability with respect to aging over time will be identified. A formalized set of standards will then be generated from this research to identify and evaluate future PUF designs.

Cyberspace networks are increasingly vulnerable to a wide array of new threats, making it imperative that we equip our airmen with advanced and superior operational capabilities in cyberspace[4]. Identifying hardware level, physics based device processes that can be exploited to increase information assurance in DoD cyber systems is a necessary shift towards creating hardware security as a built-in, ground up security measure. This type of research, including security analysis, nanoelectronics, counterfeit device detection, cryptography, and cyberattack countermeasures will elevate security as a fundamental design parameter and transform the way new nanoscale devices are developed. The applications of the technologies discussed above are envisioned to be the cornerstones of a system that greatly inhibits the unwanted exfiltration of data and proliferation of malware through multiple systems.

References

[1] Department of Defense DoD Directive 3020.40, *DoD Policy and Responsibilities for Critical Infrastructure*, July 2010, available at: *http://www.dtic.mil/whs/directives/corres/pdf/302040p.pdf.*

[2] Department of Defense Instruction (DoDI) 8500.01E, *DoD Policy and Responsibilities for Critical Infrastructure*, July 2010, available at: *http://www.dtic.mil/whs/directives/corres/pdf/302040p.pdf.*

[3] Jeyavijayan Rajendran, Student Member, IEEE, Ramesh Karri, Member, IEEE, James B. Wendt, Member, IEEE, Miodrag Potkonjak, Member, IEEE, Nathan McDonald, Member, IEEE, Garrett S. Rose, Member, IEEE, and Bryant Wysocki, Member, IEEE ”Nanoelectronic Solutions for Hardware Security”.

[4] Air Force Basic Doctrine, Organization, and Command, 14 October 2011.

The Junior Force Council: Reaching Out to New Employees

By Victoria Horan

Introduction and Council Mission

At the Information Directorate, we have a unique group of individuals that serves to ensure that the newer military and civilian employees have a positive impact on the Air Force mission. Formerly known as the "New Employee Organization", this group is now recognized across the entire Air Force Materiel Command as the professional organization called the Junior Force Council, or JFC. Each directorate of the Air Force Research Lab hosts a local JFC, and these groups meet regularly as the AFRL JFC Corporate Board.

Officially the Junior Force of our organization is made up of all military and civilian employees with ten years of service or less at the Rome Research Site. The Junior Force elects their representative council annually to guide and motivate activities specific to the needs of the Junior Force. Whether these needs relate to mentoring or leadership or training, the JFC will work to find a solution. Additionally, the elected council voices the ideas and concerns of the Junior Force to leadership, both locally and at the higher levels.

Benefits of a Council

The Information Directorate benefits greatly from the creation of the Junior Force Council. Some of these benefits are clear, such as the benefits of offering training and professional development opportunities to junior employees. However the Council offers many other benefits as well. Serving on the council provides a junior employee an opportunity to exercise their leadership skills. Attending cross-directorate events allows for networking and collaboration that can be challenging for a new employee working in a lab. Additionally, planning and attending JFC events can allow junior employees to showcase their skills and abilities with senior leaders present. Finally, a core activity hosted by our JFC is providing opportunities for our employees to visit other Air Force installations to get a better understanding of our role as a research lab within the greater Air Force. All of these benefits combined provides for a stronger work force and improved employee retention.

Council Activities

Mentoring and meetings with senior leaders

Mentoring is a key component to the development of strong leaders in the workforce. The Junior Force can gain valuable insight into the roles and responsibilities of our senior members through both formal seminars and video teleconferences (VTCs) with leaders at all levels. We have been blessed with strong support from our leadership and have opportunities to network locally and via VTC with some of the top ranking officials within RI, AFRL, and AFMC. Some of our past speakers have been directors of other AFRL directorates and the director of staff at AFRL Headquarters. This unfiltered access to experienced leaders gives the Junior Force a unique and informative perspective on the workings of our organization.

On top of our group meetings with senior leaders, the JFC also offers a mentoring program at the local level. This program serves to provide not only traditional junior-to-senior mentoring partnerships, but also cross-division, civilian-military, and senior-to-junior mentoring pairings. By providing a wide variety of options for these relationships, we allow our workforce to fill in any gaps in their local networks and to learn about processes and procedures in a less intimidating and more personal setting.

Bluing Trips

The Air Force has adopted the term "bluing trip" to define a trip with the sole purpose learning about the Air Force as a whole and how a specific organization fits into the bigger picture. For research labs like the Information Directorate, this provides a key opportunity to learn about the operational aspects of our service and to understand what needs the warfighter may have that might be met by our local scientists and engineers. For employees of a geographically separated unit such as RI, this allows for insight into the workings of an Air Force base and the multitude of organizations housed in one location.

Each year, the JFC in Rome organizes a bluing trip focusing on one specific theme. While the trips are organized with this theme in mind, additional opportunities such as networking and openings for collaboration are always presented on these trips. In 2015, we were fortunate enough to have funds for two such trips.

Our most recent trip in August centered on mentoring for Junior Force employees. This trip to Wright-Patterson AFB brought 17 Junior Force employees and our two senior advisors to the annual JFC Symposium, hosted by the AFMC HQ JFC. This symposium featured a range of well-known speakers, ranging from the AFMC Commander, Gen Ellen Pawlikowski, speaking about how to approach new assignments, to Rep Niraj Antani, from the Ohio state House of Representatives, who motivated young professionals to make a difference. In addition to attending the symposium, attendees on this trip also toured and learned about research and ongoing collaborations with AFRL/RH, 711th HPW, and NASIC.

The first trip of 2015 centered on a different theme: engagement with the warfighter. This trip to Langley AFB centered on a visit to Air Combat Command to learn about our collaborations and connections. In addition to learning about the structure and mission of ACC, we also met with senior leaders from the Ryan Center, the Targeting Center, 480th ISR Wing, and NASA-Langley. This visit allowed Junior Force employees to see how our research and collaborations at AFRL/RI are providing tools for the warfighter in the field.

Training and Professional Development

The JFC also hosts regular training and professional development opportunities. For example, we regularly offer workshops on common processes and procedures

Figure 1: RI Junior Force employees meeting with Mr. Mike Gill, AFMC Executive Director

Photo by: Al Santacroce, AFRL

Figure 2: RI Junior Force employee 2Lt Val Red visiting NASA-Langley.

Photo by: Al Santacroce, AFRL

here at RI, such as writing our year-end review under the "Contribution-Based Compensation System". Additionally, we often host technical workshops and classes such as hosting an Android Boot Camp with Big Nerd Ranch.

In addition to training workshops and seminars presented to the Junior Force Council by senior leaders, our council members and Junior Force employees are always welcome to develop their own training opportunities as they see fit, such as a recent series of seminars titled "Path to Publication". These seminars were created and presented by Junior Force employees with doctorate degrees, and discussed topics such as how to write technical papers, what the journal paper submission process entails, and how to present technical work. Our junior workforce has a vast breadth of experience when considered cumulatively, and it is in our best interest to take advantage of it.

In conjunction with the Junior Force Council, the Information Directorate provides additional professional development for some of our brand new employees. In 2012, a new branch was established as the "New Employee Training and Evaluation Branch", under the supervision of Mr. Todd Humiston. This branch provides new employees an opportunity to work with up to three senior mentors in their first two years to ensure that the employee finds the right fit. By sampling a variety of project types, the new employees can learn about the Information Directorate while also finding a niche for their contributions. Since not all new hires are able to take advantage of this opportunity, the Junior Force Council stays in close contact with the New Employee Branch, and Mr. Humiston serves as one of the senior advisors to the council. Our other senior advisor, Ms. Linda Reed, serves as the Chief of Contracting at RI, and helps us to bridge the gap between mission and support employees.

Collaboration

One of the larger goals of the Junior Force Council is to arrange events aimed at increasing collaboration between employees at all levels in the Information Directorate. To support traditional scientific collaboration in-house, the JFC has hosted seminars and poster sessions to highlight the contributions of our researchers, both senior and junior.

Mission Statement
Lead research, development and integration of affordable information exploitation and cyber technologies for transition to our air, space and cyberspace forces.
Testing for Hardware Trojans: A Game-Theoretic Approach

Figure 3: Junior Force Employee Dr. Charles Kamhoua briefing his research to Ms. Linda Reed, Chief, Contracting Division, at a JFC Poster Session.

Photo by: Al Santacroce, AFRL

The JFC also works to provide openings for a variety of collaborations, such as building relationships between mission and support division employees. By including all junior employees locally, we are able to meet and understand the contribution of all employees.

Finally, we work to offer networking opportunities with members of JFC organizations at other Air Force sites. Through our network of councils within the directorates of AFRL and other installations, we are able to connect junior scientists and engineers locally with others working in similar fields.

Conclusion

To summarize, the Information Directorate provides a fantastic opportunity for junior employees to excel through the Junior Force Council. By creating a professional organization for the Junior Force and run by the Junior Force, new employees are able to find almost any resource necessary to succeed at AFRL. The key ingredient for a successful JFC is the support of supervisors at all levels. By maintaining senior leaders as advisors to the council and meeting regularly with upper management, the council is able to deliver the training, networking, and mentoring that is crucial to a junior employee's development.

About the Author

Victoria Horan is a research mathematician at AFRL/RI, the Information Directorate, working in the Networking Technologies Branch. She completed her B.S., M.A., and Ph.D. degrees in mathematics at Arizona State University, specializing in graph theory and combinatorics. She is currently the principal investigator for several efforts, both in-house and under the Complex Networks program at AFOSR. Dr. Horan also serves as one of the Junior Force Council co-chairs for the Information Directorate in Rome, NY.

RELEASE STATEMENT

Approved for public release; distribution unlimited: 88ABW-2015-4217

Application Specific Abstractions: A Research

By Lok Yan

The success of modern computing can be largely attributed to abstractions. By abstracting away the intricate details of how lower level components work, users at higher levels of abstraction are able to focus their efforts on content creation instead. While abstractions can drastically reduce complexities, they can also hide important details resulting in lower performance and even unintended software vulnerabilities.

In this paper, we argue that today's abstractions are overly generic resulting in the loss of important semantic and contextual information across abstraction boundaries. This loss of information leads to security vulnerabilities as well as inefficiencies. We suggest application specific abstractions as a potential solution and present a research agenda that builds upon recent results from the Electronic Design Automation and Program Analysis communities.

I. Introduction

In computing, abstraction is a technique used to hide certain complexities of machines. Users of an abstraction are presented with a simple and consistent model of the machine even though the underlying hardware and software could be extremely complex and ever changing. Abstractions effectively bisect developers into two groups (low and high) bounded together by an agreed upon interface specification. The lowlevel developers ensure all operations defined in the interface are implemented for the low-level hardware and software configuration while the high-level developers focus on creating new systems by using the agreed upon interface. In this way, developers can focus on authoring software for a specific version of Windows, Linux, OSX, Android, iOS, or others without having to worry about how much physical memory is available or how files are organized on a disk.

Abstractions reduce development costs by ensuring that common low level operations are only implemented once and complexity is reduced. Take file access for example. One can require each application to implement their own filesystem or one can create a filesystem abstraction layer that implements the filesystem once, and allows the applications to reuse a single implementation through an abstraction interface. Modern computing systems use the latter approach because it is extremely rare that an application needs to worry about the specifics of how files are organized on a physical disk.

While abstractions can be a boon to productivity by simplifying the machine model and reusing code, it can also be a detriment to security and performance if they are used in a *general purpose* fashion as most modern usages are. We argue in this paper that security vulnerabilities can arise when abstraction layers fail to understand the semantics of high level requests. Similarly, there is a performance overhead when the high level applications fail to understand or have access to resources that have been abstracted away by the layer. Both cases arise because the abstraction layer was designed to be generic to support as many applications as possible, thus ignoring application specific requirements or opportunities.

We argue that more malleable application specific abstraction (ASA) layers are necessary to balance the competing factors of developmental cost/convenience, security and performance. We also argue that formal methods and automation can be used to facilitate the transition to ASA; and that these techniques are necessary for a future with an overabundance of transistors due to continued process scaling without power scaling.

The rest of the paper is organized as follows. We first present some background information on abstractions in modern computing as well as examples of how they can impact the security and performance of high level applications in Section II. We present background on why there is an overabundance of transistors in modern and future integrated circuits designs in the same section. These serve as the foundations to our research agenda discussions in Section III. In that section, we will present a small sampling of related work in the Electronic Design Automation (EDA) and Program Analysis (PA) research fields and discuss how ASAs require a multidisciplinary approach. Finally, we summarize our arguments in Section IV.

II. Background

The relationship between abstractions, security and performance was introduced in the previous section. The opportunity to utilize the extra transistors in modern IC designs was introduced as well. We elaborate on those initial observations in this section.

A. Abstractions

Abstraction is a technique that 1). hides the complexities of a system and replaces it with a simplified model called an *abstraction interface*; and 2). partitions the developers into low, who implements the abstraction layer, and high, who uses the abstraction layer to build higher level systems. Simplification and division of labor help reduce the costs of development and is the source of modern computing success.

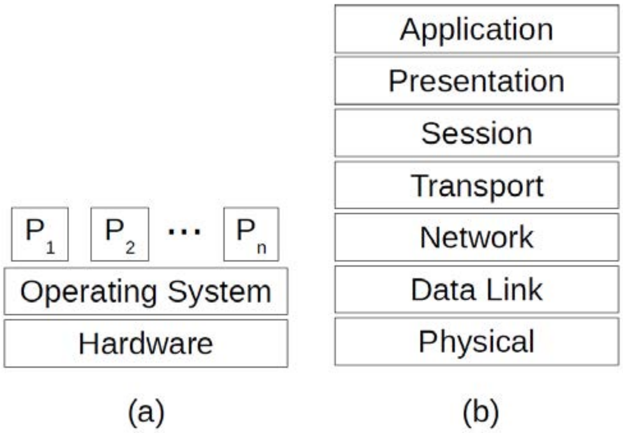


Fig. 1: Computing Stack (a) and Network Stack or OSI Model (b)

A well known computing abstraction is shown in Figure 1a. Here, the operating system abstracts away the complexities of lower level hardware and presents higher level processes and applications with a System Call Interface. A more complex abstraction stack, the Open Systems Interconnection (OSI) model, is depicted in Figure 1b. The OSI model shows multiple abstraction layers stacked on top of each other as seen from a communications perspective. This demonstrates that abstraction is used both vertically (e.g., the presentation layer further abstracts the session layer) and horizontally (e.g., there is more than one way to hide complexities).

Moreover, abstractions are not only used in software, but also in hardware as depicted in Figure 2. At the bottom of the figure are the fundamental building blocks of modern computers: transistors, resistors, capacitors, etc. The details of these analog devices such as the power and delay characteristics are abstracted away by the manufacturers who present digital logic gates in the form of standard cells that are specific to a manufacturing process. Chip designers can then use the cells to build progressively larger structures such as blocks and modules that are reused and combined to create sophisticated integrated circuits (ICs). These individual ICs are then combined into Systems-on-Chip (SoCs) which represent much of modern personal computing requirements dominated by phones, tablets and internet-of-things.

The rest of the stack is similar to that of Figure 1a, except for the two additional abstraction layers above the operating system in this notional Android stack. In Android, applications or Apps can either use the Android App Framework or native libraries to interact with the rest of the system. As an extra abstraction layer, the App Framework greatly simplifies most common tasks such as interacting with the user and inter process communication.

The framework and the Android Runtime layer that emulates the Dalvik Bytecode of Android Apps, being general abstraction layers, do incur additional overhead. For this reason, performance intensive applications such as games and multimedia Apps use native libraries to bypass the additional abstraction layers and interact with the system directly. This flexibility in how Apps interact with the rest of the system is the beginning of application specific abstractions.

Though the abstraction hierarchy in Figure 2 is fairly deep, it still does not represent all abstractions. In particular, programming language abstractions are not shown. Programming languages are examples of abstractions[1] that are implemented through *translation* whereas the layers in the Android stack implement abstraction through *extension and interpretation* [1].

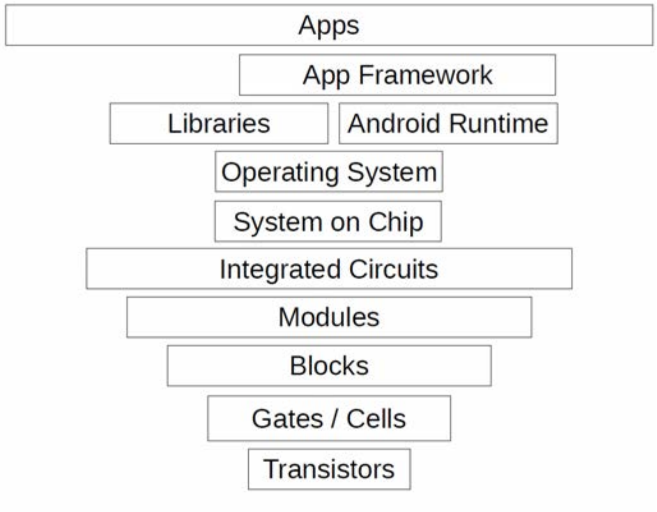


Fig. 2: Hardware and Software Stack for Android

In translation, the language specification is the interface specification, and the abstraction layer is the compiler. For example, Android Apps are written mainly in Java. The Java source is then compiled down to Dalvik Bytecode (the App's executable code).

Interpretation can be seen as a dynamic counterpart to translation. The compiler statically translates the behavior described in the higher level language into something that can be executed natively, but the interpreter must decode (interpret) the requests made by higher level users and execute code on their behalf. The Operating System or the Android Runtime are examples of interpretation.

Extensions are abstractions implemented in the same language as the intended users, which implies shared libraries. In this case, the functions, methods, classes etc. exported by a shared library constitute the abstraction interface and the library itself is the implementation.

1) Abstraction Interface Specifications: There are three entities in an abstraction: interface, low-level developer and high-level developer. Thus, we separate the sources of security vulnerabilities into security vulnerabilities due to ill-defined specifications, poor or incorrect low-level implementations and poor high-level implementations. Since vulnerabilities arising from implementation errors can result with or without abstraction, we will only focus on vulnerabilities due to ill-specified interfaces. We further limit our discussions to problems due to the nature of abstractions and how they are used only. Our discussion does not include cases where the interface specification was defined incorrectly. Thus, mistakes in the specifications, implementation and usage of abstractions are out of scope.

The abstraction interface specification is an agreement that binds the low-level and high-level developers. On one extreme, the rules within the specification are interpreted and defined by one party. In other words, the low-level developers define the interface on their own and the high-level user simply uses the interface as is. These are *generic abstractions* since there is a desire for the specification to be generic enough to cover many possible users. Generic abstractions exhibit a one to-many[2] relationship where one single low-level developer defines the interface for many users. We use the progressively wider blocks in Figure 2 to illustrate this concept.

On the other extreme, the rules are defined jointly and are *application specific*. These run the risk of being overly constrained though. Pictorially, if all abstractions layers are one-to-one then all blocks are of the same width.

2) Security: From a security perspective, application specific interfaces are desirable because the user's requirements can be incorporated directly into the interface specification. Unfortunately this is only applicable to highly sensitive applications that can absorb the high developmental costs, because it removes the multiplicative development factor of generic abstractions. One of the main weaknesses of generic abstractions is that by defining the specification without input from the user, some very important information can be lost leading to a semantic gap problem. Buffer overflows are a good example of this.

A buffer overflow occurs when the data is written beyond the confines of the intended buffer. While these can be attributed to programmer error, we argue that those that cross an abstraction boundary, e.g., resulting from a call to

[1] Dennis uses the term hierarchies

[2] These can also be many-to-many relationships where multiple groups implementthe same specification, e.g., emulators, simulators and real machines, but the users still greatly outnumber abstraction developers

a library function such as memcpy and strcpy, are the products of overly generic specifications. In these cases, the definition of the "intended buffer" is simply lost through abstraction. To demonstrate this we compare array copy as defined in C/C++ and Java.

In C++, arrays are copied either using memcpy, if the array elements are primitive types, or std::copy. Both are generic abstractions by extension. memcpy defines both the src and dst pointers as void* meaning all type information is lost and std::copy is a template function where all instantiations must support a set of common operations. Furthermore, both require the user to ensure that the destination buffer is large enough to hold all of the copied elements. If the user makes a mistake and tries to copy too much, then a buffer overflow vulnerability is created.

Java, on the other hand, automatically creates new Class definitions for every new array type [2]. That is when the user defines an int array (int[]) the Java virtual machine automatically creates an int array class. Copying arrays is achieved using the assignment operator which, for array classes, automatically performs bounds checking. This is an example of an application specific abstraction. One could argue that the C++ template function std::copy serves the same purpose if iterators also perform bounds checking by default. This is only true if the user defines iterators that does so. The built-in iterators do not perform bounds checking for performance reasons.

The same problem exists in abstractions through interpretation. Like memcpy, the read system call relies on the user to ensure that the buffer is large enough and removes the type information as well. The loss of type information can exacerbate security issues since interpretation uses different high and lowlevel languages. For example, when a Java program uses JNI (Java Native Interface) to call native code, all protections that

are afforded by the Java Virtual Machine and the language are lost. This means that writing to primitive Java arrays at the native level no longer generates ArrayOutOfBoundsExceptions. The implementer of the JNI function could perform bounds checking, but this is an example of application specialization and also this special treatment cannot be sustained across the multiple abstraction layers as exhibited in modern platforms.

A compiler is the abstraction layer in translation. Because compilers rely heavily on formally proven methods and algorithms, we believe the only source of security issues are due to under-specified or unspecified behaviors resulting in incomplete proofs and errors in the implementation - the latter of which are out of scope for our discussion. Undefined behaviors are interesting in the sense that the specification itself forces the low-level developer to make a design decision that defines the behavior.

A recent study showed that undefined behaviors led to unstable code (a superset of insecure code) in about 40% of all Debian Wheezy packages written in C/C++ [3]. The authors found that modern compilers are removing behaviors such as null-pointer checks, thereby violating the programmer's intent and introducing security vulnerabilities, for optimization purposes. The authors also note that the unstable behavior is only introduced at higher optimization levels for certain compilers. Since the optimization level is a configuration option, it is also an indication that application specific abstractions - instantiated through configuration - are desirable.

We have provided some background and examples into some security and reliability related issues due to the use of general abstractions in this section. We have also stated that modern computing stacks have many layers of abstraction even though there are overheads. The next subsection will present a brief description on how the community and industry has been able to hide the overhead through more sophisticated and powerful processors. We will also discuss how this dynamic is changing.

3) Overabundance of Transistors: Moore's law has been a mainstay of the semiconductor industry. Over the past few decades, chip manufacturers have continually made advances that effectively doubled the number of transistors in an integrated circuit every two years. While processors of the 1970's ranged in thousands of transistors per chip, today's state of the art processors range in the billions. The increase in transistor counts was mirrored by a similar increase in operating frequencies and performance until about a decade ago. These increases not only allowed for higher performance chips, but also higher sophistication in terms of predictive execution logic, longer pipelines, and specialized instruction sets (horizontal abstractions) well beyond the humble beginnings of the Intel 4004 general purpose processor of the 1971. The performance increases also allowed for far more sophisticated software built upon a multitude of abstraction layers.

The end of frequency scaling is attributable to end of Dennard scaling. In brief, the switching frequency of a transistor can increases as long as the threshold voltage of a transistor decreases in line with decreases in feature sizes; this condition is known as Dennard scaling. (A proper treatment of the topic can be found in Chapter 2 of Shacham's dissertation [4]). The end of frequency scaling gave way to multi-core scaling where more cores are added to processors to attain performance gains through parallelism and to use the available transistors.

However, these gains are unsustainable. The end to Dennard scaling also means as the number of transistors go up so does the power necessary to drive the overall chip. Thus increasing the number of cores also require higher capacity power sources and cooling solutions to dissipate the heat generated. The industry hit a "power wall" [4] that was exacerbated by the dramatic increase in mobile platforms where both power and cooling are limited. Given the limited power available in mobile platforms and even decreasing power envelopes to obtain better battery life, increasing the number of transistors meant there will come a time when not all transistors can be utilized and not all cores can be processing at once. This is known as the "dark silicon" problem and already exists today [5], [6].

Dark silicon is less of a problem and more of a design constraint though. There isn't a rule that states all transistors must be fully utilized; the transistors simply can't all be activated concurrently, they can be effectively used sequentially. Dark silicon does limit the ability to attain better performance through simple and general abstractions such as multi-core. As in the case for security, application specific abstractions and application specific integrated circuits seem to provide a way forward and these extra transistors can be used to serve that purpose. We discuss a research agenda that is based on and follows recent research in the next section.

III. Research Agenda

Our goal is to create a computing environment that supports application specific abstractions. This is only feasible if two conditions are satisfied: since each application specific processor is unique, there must be enough area on a die to support many different such processors; and since the non-recurring engineering costs for ASIC development are extremely high (at over $40Million per modern state of the art ASICs [4]) the development or non-recurring engineering (NRE) costs must be drastically reduced. We observe that the overabundance of transistors can help satisfy the first requirement and automation can help with the second. We present three major research directions that together can make application specific abstractions a reality throughout the modern computing stack.

Direction 1 - Informed Lower Level Stacks: Given the depth of modern computing stacks (Figure 2), having ASAs imply all abstraction layers are specially defined for the single application at the top of the stack.

Instead of designing processors that are specialized for individual applications, Turakhia et al. proposed an architectural synthesis framework named HaDeS that uses benchmarks to model the expected runtime behavior of various application types and then use the model to algorithmically determine the optimal allocation of cores in a heterogeneous chipmulti-processor [7]. The only requirement is that a library of general purpose cores that can process all of the benchmark applications, albeit with different performance characteristics, is already available. Thus, they can make use of the overabundance of transistors, but NRE costs are low as long as the library of cores already exists.

The library could be automatically generated using the Genesis 2 chip generator [4] where the benchmark program models are used as constraints to Genesis 2. Since Genesis 2 embodies the experience and expertise of chip designers during processor template creation, it is not only possible to generate a library of cores, but a library of cores that are optimized for each application in the benchmark suite or for each desired computational kernel feature. In Genesis 2, the only major NRE costs are in the development of the processor template - all processor instantiations and supporting software are automatically generated and therefore incur negligible additional costs - and verification costs.

Instead of designing processors for classes of applications as represented by the benchmarks, Goulding-Hotta et al. designed a processor, call GreenDroid, that is optimized for certain popular functions in the Android software stack [6]. They first analyzed Android framework software to identify heavily used components such as the Davlvik

Virtual Machine and web browsing related libraries. Once the hot code blocks are identified, they then automatically synthesized *conservation cores* or *c-cores* that efficiently implemented a portion of the hot components (i.e., a subgraph of the control flow graph) in hardware. Each c-core is then associated with a special instruction and the original Android software patched to take advantage of these application specific c-cores. The rest of the software executes normally on the general purpose core. GreenDroid is both automated and makes use of the extra transistors. The only drawback is that it requires the applications to have already been implemented.

Observation: Overall, researchers are making strides toward a future where the design of processors are informed directly by specific instances of expected applications, however there are still limitations. The processor designs are still only application class specific instead of application specific. Moreover, the work presented above is focused on performance which is much easier to model, quantify and compare than security. Thus, future research in this direction must not only seek to reduce NRE costs such as verification, it must also be capable of incorporating more complicated features, constraints and requirements such as security and reliability.

Direction 2 - Configurable Abstraction Layers: As in the case with compiler optimization levels affecting whether unstable code is generated, abstraction layers that are inherently generic could present knobs for individual applications to customize. Secure Computing (SECCOMP [8]) is a representative example. SECCOMP is a Linux kernel feature where applications can specify a set of system call filters. These filters can be used to prevent the invocation of system calls based on the system call number and parameters. This can be used to effectively remove unnecessary functionality and also reduce attack surfaces.

Instead of starting with a generic abstraction and then blacklisting unwanted functionality, it is also possible to create an abstraction template that is instantiated in accordance to the user's needs. Software Fault Isolation (SFI [9]) and Control Flow Integrity(CFI [10]) are examples of new abstraction concepts that uses program analysis techniques to implement protection mechanisms that are unique to each application.

SFI is a software only technique that is used to partition a single process's memory space into multiple ranges. Protection instructions are then automatically inserted around memory accesses so as to enforce the partitioning scheme. CFI is the control flow counterpart to SFI that ensures all control flow transfers (e.g., branches and function calls) are to expected and allowed locations determined through the automated analysis of the application itself. While there are some important technical limitations to the techniques, such as indirect branches, they demonstrate that templates can be used to synthesize abstraction layers as well as processor designs as in Genesis 2.

Observation: It can be seen that filtering can be applied to other abstractions and these software based techniques could be incorporated directly into future hardware designs. The question that remains is given a set of techniques, which ones should be implemented in hardware and which in software. In a world with an over-abundance of transistors, a related question is what roles can horizontal abstraction layers play?

Direction 3 - Hardware/Software Co-Design, Co-Synthesis and Co-Verification: Hardware/software co-design, synthesis and verification has been applied in embedded systems for decades and is a good embodiment of our overall goals. Recent work has demonstrated that it is feasible to specify an application's behavior in a single language and have tools automatically partition the behavioral description into hardware and software components as well as automatically synthesize them. The Bluespec Codesign Language (BCL) is such a language [11], [12].

In BCL, a programmer specifies an application's behavior in Bluespec System Verilog (BSV) extended with Guarded Atomic Actions that controls whether the state of the program is updated at runtime. These annotations help the BCL compiler determine which portions of the behavioral specifications must be sequenced and which can be executed in parallel.

A programmer also annotates the specification with knowledge on where the natural communications boundaries between the application's modules are. The BCL toolchain can then determine which modules are suitable for hardware versus software implementation based on predictive performance models and proceeds to automatically generate C++ code for the software modules, BSV for the hardware modules, and all the glue code necessary to schedule the components and tie them together.

In addition to generating the hardware and software artifacts, the BCL toolchain also generates simulators that can be used to debug the final design. This further helps reduce development costs.

Observation: BCL can be seen as the basis of future ASA toolchains as long as two problems are addressed. The BCL language is at the Register Transfer Level which will need to be abstracted again through translation lest it will not be suitable for developing high level applications. It has yet to be shown if the same techniques scale beyond embedded systems where the computing stacks are deeper and software applications are more numerous.

IV. Summary

In summary, we presented some background information on the problems of generic abstractions and argue that application specific abstractions are necessary to attain better better security, reliability and performance guarantees. We argue that since there is an over-abundance of transistors, it will be feasible to host many specialized processing cores (up to one per application) on a single die. We then presented related work in three research directions that contribute to the end goal of application specific abstractions: Informed Lower Level Stacks, Configurable Abstraction Layers, and Hardware/Software co-design, synthesis and verification.

References

[1] J. B. Dennis, "The design and construction of software systems," in *Software Engineering, An Advanced Course*, Reprint of the First Edition [February 21 - March 3, 1972]. London, UK, UK: Springer-Verlag, 1975.

[2] J. Gosling, B. Joy, G. Steele, G. Bracha, and A. Buckley, *The Java Language Specification, Java SE7 Edition*. Oracle, 2013, ch. 10: Arrays. [Online]. Available: https://docs.oracle.com/javase/specs/jls/se7/html/index.html

[3] X. Wang, N. Zeldovich, M. F. Kaashoek, and A. Solar-Lezama, "Towards optimization-safe systems: Analyzing the impact of undefined behavior," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, ser. SOSP '13. New York, NY, USA: ACM, 2013.

[4] O. Shacham, "Chip multiprocessor generator: Automatic generation of custom and heterogeneous compute platforms," Ph.D. dissertation, Standford University, 2011.

[5] B. Raghunathan and S. Garg, "Job arrival rate aware scheduling for asymmetric multi-core servers in the dark silicon era," in *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES '14. New York, NY, USA: ACM, 2014.

[6] N. Goulding-Hotta, J. Sampson, G. Venkatesh, S. Garcia, J. Auricchio, P.-C. Huang, M. Arora, S. Nath, V. Bhatt, J. Babb, S. Swanson, and M. Taylor, "The greendroid mobile application processor: An architecture for silicon's dark future," *IEEE Micro*, vol. 31, no. 2, Mar. 2011.

[7] Y. Turakhia, B. Raghunathan, S. Garg, and D. Marculescu, "Hades: Architectural synthesis for heterogeneous dark silicon chip multiprocessors," in *Design Automation Conference* (DAC), 2013 50th ACM/EDAC/IEEE, May 2013.

[8] "SECure COMPuting with filters." [Online]. Available: https://www.kernel.org/doc/Documentation/prctl/seccomp filter.txt

[9] R. Wahbe, S. Lucco, T. E. Anderson, and S. L. Graham, "Efficient software-based fault isolation," in *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, ser. SOSP '93. New York, NY, USA: ACM, 1993.

[10] M. Abadi, M. Budiu, U. Erlingsson, and J. Ligatti, "Control-flow integrity," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, ser. CCS '05. New York, NY, USA: ACM, 2005.

[11] N. Dave, "A unified model for hardware/software codesign," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[12] M. King, "A methodology for hardware-software codesign," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

About the Author

Lok Yan is a Computer Engineer at the Air Force Research Laboratory, Information Directorate, in Rome, NY. He is currently the Lead for the Foundations of Trusted Systems Sub-Core-Technical-Competency (CTC) under the Cyber Science and Technology CTC. He has a B.S. in Computer Engineering and M.S. in Electrical Engineering from Polytechnic University (now part of New York University) and a Ph.D. in Computer Information Science and Engineering from Syracuse University. e-mail: Lok.Yan@us.af.mil.

RELEASE STATEMENT

Approved for Public Release; Distribution Unlimited :
88ABW-2015-4644 20150929

Cross-domain Transfer: Information Support Server Environment (ISSE)

By Alex H. Gwin (Capt, USAF) and Richard C. Barrett

A. Cross-domain Transfer

The proper treatment of classified data has always been important throughout this nation's history. Classification of data was present even in the early period of the American Revolution when the Continental Congress passed a resolution in September 1774 to keep its proceedings secret [1]. It wasn't until March 1940, before World War II, when the formal classifications of secret, confidential, and restricted were established. Many executive orders since then have refined the treatment of classified information [2].

Over the past ten years, leaks (whether intentional or unintentional) have made major news headlines. Examples include the release of classified documents and emails by WikiLeaks since 2007 [3, 4] and the leakage of classified information by Edward Snowden in 2013 [5]. Data must be properly handled and protected in accordance with its classification level. It is widely regarded that the proper treatment of data commensurate with its classification level is important now more than ever. In this digital age, the access of information is lightning fast, and proper security protocols must be established and followed to prevent future leaks.

To ensure proper safeguarding of classified data, isolated domains/networks are used, such as the Non-secure Internet Protocol Router Network (NIPRNet), the Secret Internet Protocol Router Network (SIPRNet), and the Joint Worldwide Intelligence Communications System (JWICS), as well as other domains specific to missions and coalition partners. The domains are separate and isolated to protect their information. However, isolated domains create the problem of information isolation—the inability to share information. Classified information is useless unless it can be visible to the people that make decisions based on its facts. To transfer this information effectively and securely, an electronic capability with built-in security protocols is needed between the domains—that is, a cross-domain transfer solution.

B. ISSE Overview

ISSE (Information Support Server Environment) is a system with a long history that has evolved to become a premier cross-domain solution (CDS). It is a cross-domain transfer solution developed, maintained, and installed by the Information Handling Branch of the Air Force Research Laboratory (AFRL) Information Directorate in Rome, New York. It is also on the Unified Cross Domain Services Management Office (UCDSMO) baseline list, and it is fully accredited according to CNSSI 1253, NIST SP 800-53, and ICD 503 requirements.

ISSE was originally released as the USAFE (United States Air Forces in Europe) Guard in 1988 by the Rome Air Development Center. USAFE Guard's sole purpose was to disseminate threat update messages. It operated on a Harris Nighthawk computer with CX/SM MLS operating system. The system was officially re-branded and certified as ISSE in 1995. This work was done ahead of key government actions, such as the establishment of the multi-level security (MLS) working group by the Defense Information Systems Agency (DISA) in 1997. In 2001, Top Secret/Sensitive Compartmented Information (TS/SCI) and Below Interoperability (TSABI) and Secret and Below Interoperability (SABI) were coined in order to create categories of flow between domains with distinct security requirements.

ISSE provides the capability to transfer data bi-directionally between domains in either TSABI (commensurate with TS/SCI to/from Secret) or SABI (commensurate with Secret to/from Unclassified) cases. In either TSABI or SABI, the domain with the highest level of security is called the Controlling Security Domain (CSD) and the other domains are called Non-controlling Security Domains (NCSDs). At the time of publication of this paper, over 140 structured and unstructured files types can be transferred, including Microsoft Office files, images, video, databases, and chat.

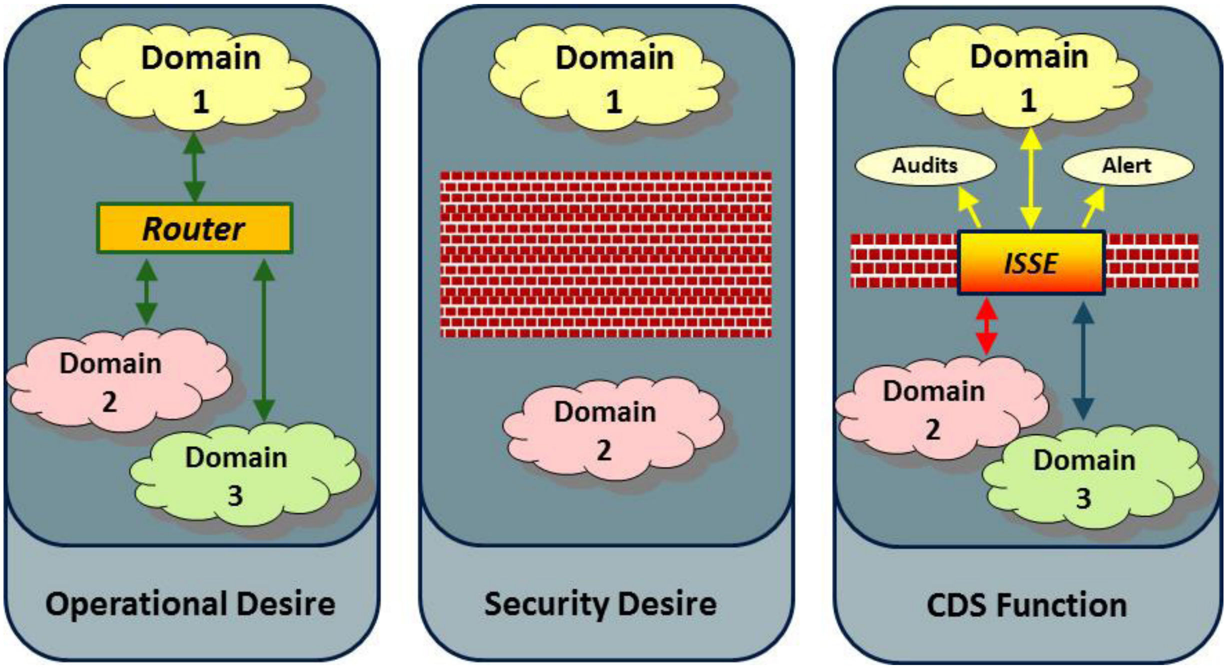


Figure 1. Cross-domain solutions (CDSs) balance operational and security desires. Data is securely transferred between separate security domains while maintaining high levels of security.

While transferring data is the main purpose of a CDS guard, security is equally (if not more) important. As seen in the publicized leak cases, the insecure transfer of data between domains can have adverse effects for national security. The security posture of ISSE is aggressive and well developed for preventing malicious activity. Additionally, ISSE enforces the security policies of the host unit. ISSE filtering criteria which are established by the host unit identify and flag issues when transferring files. When caught by the filters, the file is immediately pulled from the transfer queue and placed in a reviewer inbox. ISSE filters are highly configurable, based on the host unit's requirements. In addition to key word searches, ISSE parses, inspects, filters, and sanitizes. Each data path, i.e. thread, may be configured with different security policies. The thread filters check for viruses, malcode, file type, and digital signature. ISSE leverages commercial off-the-shelf software called Purifile© to inspect Microsoft Office file types, while the other filters are programmed by the ISSE software developers.

The ISSE architecture is fairly straightforward. The ISSE Secure Trusted Automated Routing (STAR) is the "guard" component at the domain boundaries that acts like a secure tunnel between security domains. Threads are established at the time of installation for data transfer in each direction. For instance, to conduct transfers between the CSD and NCSD bi-directionally, two threads are needed. The threads operate concurrently and independently from one another; that is, they operate in parallel and can be configured with different security policies. The STAR connects to the ISSE Proxy Server (IPS) of each domain. The IPS is composed of multiple Protocol Translators (PTs) and the ISSE Web Server (IWS). The PT acts to protect the STAR, compose and send email, relay COTS email, execute file transfers, and exchange data with the clients, IWS and STAR. The IWS can be configured for Reliable Human Review (RHR) and single/dual review for enhanced security. Additionally, an Application Programming Interface (API) can be configured in the STAR for mission applications that bypass the IPS. Examples of mission applications include Multi-level Database Replication (MLDBR), Full Motion Video (FMV), and Large File Slicer, which will be elaborated upon below. Two optional components for the ISSE system are Parallel Audit Review and Analysis Toolkit (PARAT) and Security and Workflow Enforcement Services (SAWES). PARAT provides near-real time audit collection and analysis. It collects, organizes, and presents the audits collected by ISSE to the administrator. It may be used to monitor the file transfers, users' activity, and send alerts to the administrator. SAWES is an upstream review and orchestration engine which allows the user to self-review work, receive feedback from the automated filters, and make adjustments as needed.

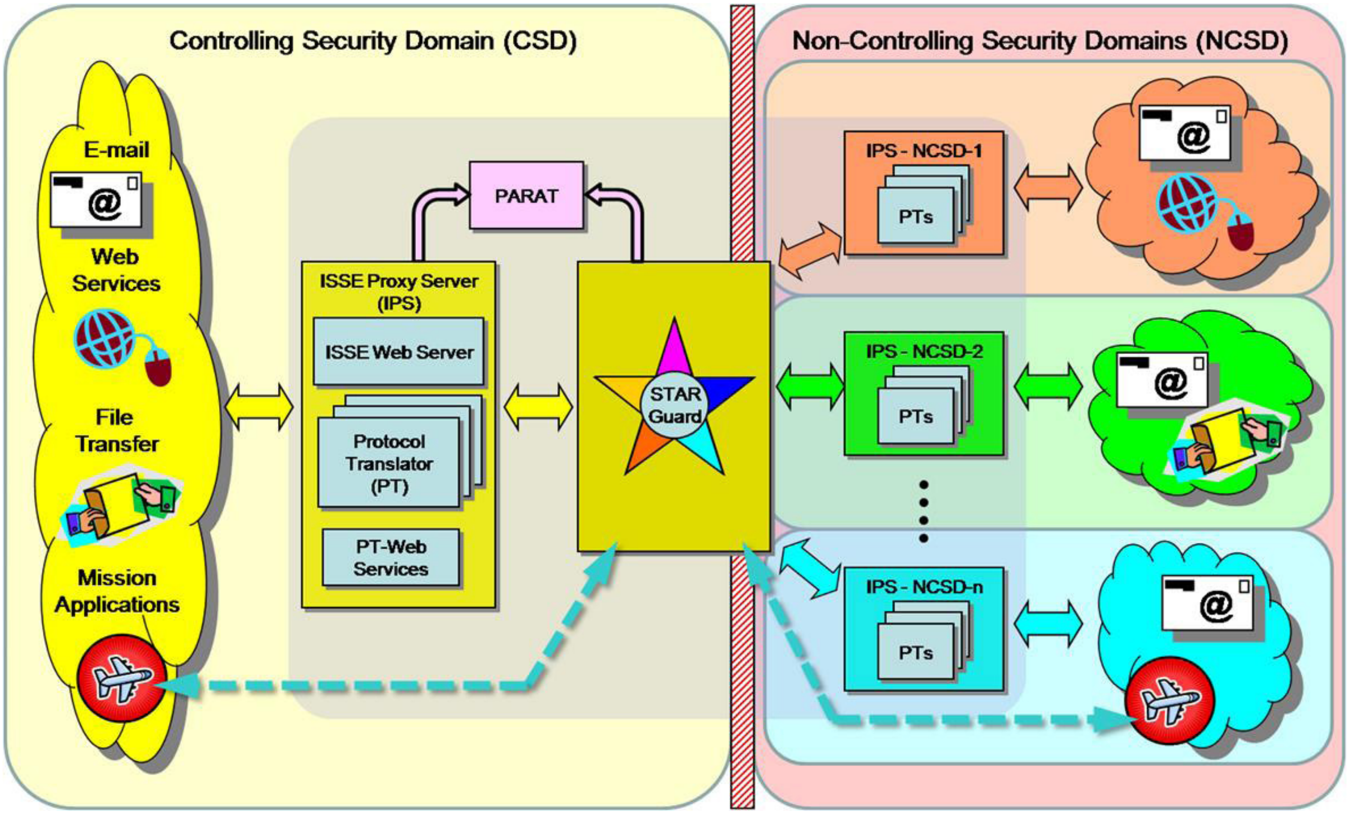


Figure 2. ISSE architecture of STAR interconnecting a CSD and multiple NCSDs. The STAR is central to ISSE's function which transfers the data and performs security checks. It connects the ISSE Proxy Servers of each network. Various capabilities are shown, such as web services, file transfer, and email.

The ISSE Program Management Office (PMO) oversees the entirety of the system acquisition. These activities include site survey, installation, training, and support from the Core Configuration Management (CCM) help desk. In order to acquire ISSE, customers in the Intelligence Community (IC) typically contact the DoDIIS Cross-domain Management Office (DCDMO). DCDMO and DISA will discuss requirements to arrive at the best CDS for the organization's needs. Other U.S. government agencies may reference the UCDSMO baseline list or contact the ISSE PMO directly. When ISSE is selected as the best solution, the ISSE PMO conducts a site survey to determine the details of the site's cross-domain requirements. Subsequently, the system is installed by the ISSE installers. On-site training is conducted for site administrators and trainers, and an out-brief is completed. At this point, the ISSE system is ready to use. Should any questions or concerns arise, agencies can call the 24/7 help desk. Most of the questions can be adjudicated immediately. If it is a more serious problem, the PMO engineers work with the site to resolve the problem. Additionally, training is offered at the PMO site in Rome, New York, several times per year. The annual support fee also covers one site visit per year.

There are several mission applications and capabilities that have been added to ISSE as user requirements have arisen. Three that will be discussed here are Multi-level Database Replication (MLDBR), Full Motion Video (FMV) adapters, and the Large File Slicer. MLDBR provides real-time, automated database replication between security domains for Oracle, DB2, MS SQL, and Sybase formats. MLDBR uses ISSE for cross-domain replication of database information; one MLDBR host can even interface with multiple ISSE systems. It leverages XML formatted messages to replicate databases between the CSD and NCSD(s). Database replication is a common user requirement leveraged by numerous organizations.

The FMV v1.0 capability was a special user request and was tested in the Unified Vision 2014 exercise [6]. At the exercise's central location in Ørland Air Station, Norway, ISSE provided 12 channels for video transfer to participants in Norway and Germany. ISSE was connected to an unclassified network and a coalition partner network. FMV performed well in this real world exercise by exhibiting exceptional video quality and less than 0.5-second latency.

Finally, ISSE is programmed to accept files less than 2 GB in size. Should the need arise to transfer files larger than this requirement, i.e. very high resolution photos, the Large File Slicer can be used. This application uses the ISSE API to communicate with the STAR. First fielded in January 2015 and demonstrated continually since then, it has human and machine interfaces that display the progress of the transfer. It operates by creating small ISSE packages from the larger file for a nearly infinite transfer capability. It sends the packages in parallel through the STAR while checking for security and malware, and compiles the pieces into the original file on the receiving end.

C. Evolving Systems

When it was first accredited in 1995, ISSE was purely point-to-point and served one data transfer method, e.g. email or file drop, per installation. Version 3.4 enabled multiple organizations to transfer between two domains, and version 3.6.1 enabled multiple organizations to transfer between two or more domains. It is this v3.6.1 which is considered to be "enterprise" in today's terms. At its highest point, ISSE was fielded in 160 operational locations. Since the advent of the enterprise construct, this number has been reduced, as expected. By counting the total number of threads, we can arrive at a realistic estimate of the capabilities delivered by ISSE systems. An inventory in September 2015 placed ISSE operating on an impressive 734 threads in 73 systems worldwide. This represents 46 percent fewer systems while supporting the flow of 298 percent more data.

ISSE has evolved from its first use as a point-to-point solution to be compatible with the enterprise construct which is prevalent today. This approach to the cross-domain business makes sense for financial reasons. For an organization with cross-domain needs, being incorporated into an enterprise system saves money by reducing the installation costs and manpower costs associated with system administration. Organizations housing the enterprise systems can charge user fees to the tenant organizations and staff one or more full-time administrators who oversee the operations of the system. The major disadvantage of the enterprise construct is that many organizations are tied into one system; if that system fails, the operational consequences are farther reaching than if the organization hosted its own CDS. Despite this concern, albeit a minor one, the enterprise construct is expected to become even more prevalent as new customers come online and some existing customers transfer to enterprise.

As an example of the conversion to enterprise, one such organization migrated from 18 point-to-point systems among seven sites to three enterprise systems among three sites. This major effort resulted in real cost savings in engineering support, licensing costs, power, administrative overhead, and 50 percent less hardware, while increasing the availability of mission critical data. The organization also upgraded their ISSE systems, and the improved transfer rates from the synergistic effects of combining upgrades and enterprise consolidation resulted in more than one billion files annually, not to mention the added security and connectivity to additional security domains.

An unremitting problem for the ISSE PMO is hardware obsolescence. From inception to fielding, a major version of ISSE is several years in the making. By the time a version is fully developed, tested by the engineers, tested by the government, final configurations are made, and certification is completed, several years have passed. (Minor versions can be fielded in several months—if enough manpower is applied to the effort.) Because the new version's operating system is only compatible with certain hardware, the problem then arises that when hardware is no longer supported, there is a hardware obsolescence problem looming in the horizon. ISSE uses Oracle's Solaris operating system (OS) which has excellent security attributes. Solaris is used heavily by the bank, stock market, and insurance industries [7]. Despite this solid user base, there exists some concern about Solaris' diminishing user base and future supportability, a concern that is not necessarily shared by the PMO. A third-party study was completed to investigate whether ISSE should move to another operating system. In order to transfer ("port") to another operating system, significant funds and manpower would be required to accomplish this effort in parallel with other development and maintenance schedules. Additionally, there was no significantly compelling reason to port to another OS, since hardware obsolescence is persistent for all OSs. The ISSE PMO determined that the best alternative was to stay with Solaris and integrate new OS versions and test with beta versions whenever possible.

There is one other approach to mitigating hardware obsolescence the PMO is currently investigating. This involves placing ISSE on a cross-domain access solution. These are secure systems with virtualized security domains. Each domain is separate and therefore very secure within a small amount of hardware. The advantage of this approach is that x86 hardware can be used for the access solution, which will be supported for the foreseeable future. The Solaris OS is interfaced with a virtual machine of the access solution. As a corollary, if successful, the resultant hybrid ISSE will require fewer pieces of hardware and less power to operate. This effort is currently being completed for three domains and several mission applications on SecureView, which is a program also overseen by the Information Handling Branch of AFRL in Rome, New York.

D. Conclusion

The Information Support Server Environment (ISSE) is a cross-domain transfer solution that is used by numerous U.S. government organizations and coalition partners. It is an electronic capability which securely transfers data between separate networks. Since its initial fielding in 1995, it has become a premier cross-domain solution that has continued to meet users' needs by evolving to the enterprise construct and providing advanced mission applications. It continues to stay relevant by anticipating the changing cross-domain landscape. For more information about ISSE, please contact the ISSE PMO at rrs.isse.pmo@us.af.mil or 315-330-7838.

About the Authors

Alex Gwin is currently the deputy program manager of ISSE at Air Force Research Laboratory - Information Directorate (AFRL/RI) in Rome, New York. He is currently a captain in the United States Air Force. He has previously served as an engineer and executive officer at Kirtland AFB in Albuquerque, New Mexico, and as a master's student at Wright-Patterson AFB in Dayton, Ohio. Captain Gwin has a Master of Science in Electrical Engineering from the Air Force Institute of Technology and a Bachelor of Science from Texas Christian University.

Richard Barrett is currently the Systems Engineer for ISSE at Air Force Research Laboratory - Information Directorate (AFRL/RI) in Rome, New York. He has over 26 years of experience leading systems engineering and acquisition activities, plus four years leading U.S. Air Force ICBM operations. His systems engineering experience includes laser weapons, avionics, manufacturing equipment, fiber-optics, network communications, and cross-domain solutions. He is a retired officer from the Air Force Reserves. Mr. Barrett has a Master's of Administrative Sciences (MBA-equivalent with Information Management Systems emphasis) from University of Montana, and a Bachelor of Science in Electrical Engineering (Quantum devices and Communication Systems emphasis) from Polytechnic University (now, New York University Polytechnic School of Engineering).

References

[1] Maus, Cathy N. (July 1996). U.S. Department of Energy OpenNet. "Office of Classification: History of Classification and Declassification" [online]. https://www.osti.gov/opennet/forms.jsp?formurl=od/history.html

[2] Peters, Gerhard and John T. Woolley (2015). The American Presidency Project. "Executive Order 8381 – Defining Certain Vital Military and Naval Installations and Equipment [online]". http://www.presidency.ucsb.edu/ws/?pid=78426

[3] Joseph, Channing (Sep 2007). New York The Sun. "WikiLeaks Releases Secret Report on Military Equipment [online]". http://www.nysun.com/foreign/wikileaks-releases-secret-report-on-military/62236

[4] Khatchadourian, Raffi (April 2010). The New Yorker. "The Use of Force [online]." http://www.newyorker.com/news/news-desk/the-use-of-force

[5] Finn, Peter and Sari Horwitz (June 2013). The Washington Post. "U.S. charges Snowden with espionage [online]". https://www.washingtonpost.com/world/national-security/us-charges-snowden-with-espionage/2013/06/21/507497d8-dab1-11e2-a016-92547bf094cc_story.html

[6] North Atlantic Treaty Organization (May 2014). "More than just information gathering: Giving commanders the edge [online]". http://www.nato.int/cps/en/natolive/news_110351.htm

[7] Oracle. "Financial Services: Overview [online]". http://www.oracle.com/za/industries/financial-services/overview/index.html

RELEASE STATEMENT

Distribution A. Approved for Public Release: [88ABW-2015-4666] Distribution Unlimited.

WE WORK FOR YOUR BUSINESS.

Is your organization currently facing a challenging Information Technology oriented research and development problem that you need to have addressed in a timely, efficient and cost effective manner?

HOW CAN THE CSIAC HELP?

In a time of shrinking budgets and increasing responsibility, IACs are a valuable resource for accessing evaluated Scientific and Technical Information (STI) culled from efforts to solve new and historic challenges. In addition, users can also leverage the CSIAC's experienced technical scientists, engineers, and information specialists to answer their technical questions.

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.

Technical Inquiry Services

The CSIAC provides up to 4 hours of Free Technical Inquiry research to answer users' most pressing technical questions. Our subject matter experts can help find answers to even your most difficult questions.

Core Analysis Tasks (CATs)

Challenging technical problems that are beyond the scope of a basic inquiry can be solved by initiating a Core Analysis Task (CAT). Through the CAT program, the CSIAC can be utilized as a contracting vehicle, enabling the DoD to obtain specialized support for specific projects within the CSIACs technical domains (Cybersecurity, Information Assurance, Software Engineering, Modeling & Simulation, and Knowledge Management/Information Sharing).

CSIAC

CALL NOW! 800-214-7921

EMAIL at: info@csiac.org

FOR MORE INFO, GO TO: https://www.csiac.org/about/technical-inquiries-and-cats

NOTES:

Article Submission Policy

The CSIAC Journal is a quarterly journal focusing on scientific and technical research & development, methods and processes, policies and standards, security, reliability, quality, and lessons learned case histories. CSIAC accepts articles submitted by the professional community for consideration. CSIAC will review articles and assist candidate authors in creating the final draft if the article is selected for publication. However, we cannot guarantee publication within a fixed time frame.

Note that CSIAC does not pay for articles published.

AUTHOR BIOSAND CONTACT INFORMATION

When you submit your article to CSIAC, you also need to submit a brief bio, which is printed at the end of your article. Additionally, CSIAC requests that you provide contact information (email and/or phone and/or web address), which is also published with your article so that readers may follow up with you. You also need to send CSIAC your preferred mailing address for receipt of the Journal in printed format. All authors receive 5 complementary copies of the Journal issue in which their article appears and are automatically registered to receive future issues of the Journal.

COPYRIGHT:

Submittal of an original and previously unpublished article constitutes a transfer of ownership for First Publication Rights for a period of ninety days following publication. After this ninety day period full copyright ownership returns to the author. CSIAC always grants permission to reprint or distribute the article once published, as long as attribution is provided for CSIAC as the publisher and the Journal issue in which the article appeared is cited. The primary reason for CSIAC holding the copyright is to insure that the same article is not published simultaneously in other trade journals. The Journal enjoys a reputation of outstanding quality and value. We distribute the Journal to more than 30,000 registered CSIAC patrons free of charge and we publish it on our website where thousands of viewers read the articles each week.

FOR INVITED AUTHORS:

CSIAC typically allocates the author one month to prepare an initial draft. Then, upon receipt of an initial draft, CSIAC reviews the article and works with the author to create a final draft; we allow 2 to 3 weeks for this process. CSIAC expects to have a final draft of the article ready for publication no later than 2 months after the author accepts our initial invitation.

PREFERRED FORMATS:

- Articles must be submitted electronically.
- MS-Word, or Open Office equivalent (something that can be edited by CSIAC)

SIZE GUIDELINES:

- Minimum of 1,500 – 2,000 words (3-4 typed pages using Times New Roman 12 pt font) Maximum of 12 pages
- Authors have latitude to adjust the size as necessary to communicate their message

IMAGES:

- Graphics and Images are encouraged.
- Print quality, 300 or better DPI. JPG or PNG format preferred

Note: Please embed the graphic images into your article to clarify where they should go but send the graphics as separate files when you submit the final draft of the article. This makes it easier should the graphics need to be changed or resized.

CONTACT INFORMATION:

CSIAC
100 Seymour Road Suite C102
Utica, NY 13502
Phone: (800) 214-7921
Fax: 315-351-4209

Michael Weir, CSIAC Director
John Dingman, Managing Editor
Email: info@csiac.org

CSIAC is a DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC), technically managed by the Air Force Research Laboratory (AFRL) in Rome, NY and operated by Quanterion Solutions Incorporated, Utica, NY.

ABOUT THE JOURNAL OF CYBER SECURITY AND INFORMATION SYSTEMS

CSIAC JOURNAL EDITORIAL BOARD

Distribution Statement
Unclassified and Unlimited

CSIAC
100 Seymour Road
Utica, NY 13502-1348
Phone: 800-214-7921 • **Fax:** 315-732-3261
E-mail: info@csiac.org
URL: https://www.csiac.org/

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 DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC). The CSIAC is technically managed by Air Force Research Laboratory in Rome, NY 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.

Funding provided by a grant from Empire State Development

COVER ILLUSTRATION

Keri Burkhart, AFRL

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 Journal of Cyber Security and Information Systems Vol.IV, No I"

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
100 Seymour Road
Utica, NY 13502-1348

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

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

Cyber Security and Information Systems
Information Analysis Center
100 Seymour Road
Suite C-102
Utica, NY 13502

Journal of Cyber Security and Information Systems

Focus on Air Force Research Laboratory's Information Directorate

— IN THIS ISSUE —

Air Force Research Laboratory / Information Directorate (Rome NY)
By Charles Messenger, Quanterion Solutions Incorporated 4

Command and Control of Proactive Defense
By David Last, David Myers, Matthew Heffernan, Meghan Caiazzo, and Capt. Nicholas Paltzer 8

Cyber Deception
By Dave Climek, Anthony Macera, and Walt Tirenin 14

A Science of Network Configuration
By Sanjai Narain, Dana CheeBrian Coan, Ben Falchuk, Samuel Gordon, Jaewon Kang, Jonathan Kirsch, Aditya Naidu, Kaustubh Sinkar, Simon Tsang, Sharad Malik, Shuyuan Zhang, Vahid Rajabian-Schwart, and Walt Tirenin 18

Networking Modeling and Simulation: Bridging the Gap from Theory to Field Tests
By Elizabeth Serena Bentley, Sunil Kumar, Joel Dallaire, and Jerry Reaper 32

Bridging Fault Tolerance and Game Theory for Assuring Cyberspace
By Kevin Kwiat & Charles Kamhoua, AFRL/RIGA 38

Physics of Information Assurance
By Donald Telesca, Ph.D., AFRL/RITB 42

The Junior Force Council: Reaching Out to New Employees
By Victoria Horan 44

Application Specific Abstractions: A Research
By Lok Yan 48

Cross-domain Transfer: Information Support Server Environment (ISSE)
By Alex H. Gwin (Capt, USAF) and Richard C. Barrett 55