NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SECURITY CONSIDERATIONS FOR NETWORK-CENTRIC WEAPON SYSTEMS

by

Erik A. Nesteruk

September 2009

Thesis Advisor:	Rachel Goshorn
Co-Advisor:	Ted Huffmire

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Security Considerations for Network-Centric Weapon Systems		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Erik A. Nesteruk			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

This thesis describes the security risks for network-centric weapon systems as a combination of different aspects of security, each with its own threats and mitigation strategies. Computer and network security deals with cryptography, authentication, and attacks on software. Information security deals with the ability of the system to process information of different classifications but prevent disclosure to unauthorized users. Physical security ranges from hardware destruction to reverse engineering of captured hardware. Operational security covers the inability of covert units to transmit to the network without compromising their positions. Personnel security discusses the ways that people can intentionally or accidentally weaken the system during development or operations.

Security of network-centric weapon systems is now a System of Systems (SoS) engineering problem and system developers must therefore embrace a systems engineering approach to security and consider all the threats and vulnerabilities facing the system. This examination must include not only the technical characteristics of the components but also the people who operate and maintain the system and the requirements of the mission. Only by mitigating the most efficient attacks on the system, regardless of the type of attack, can developers maximize the overall security of the system.

14. SUBJECT TERMS Network-Centric, Weapons Systems, Systems Engineering, Security, Cryptography, Authentication, Espionage, Sabotage, Confidentiality, Integrity, Availability			**15. NUMBER OF PAGES** 93
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SECURITY CONSIDERATIONS FOR NETWORK-CENTRIC WEAPON SYSTEMS

Erik A. Nesteruk
Lieutenant Commander, United States Navy
B.E., Vanderbilt University, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: Erik A. Nesteruk

Approved by: Rachel Goshorn
Thesis Advisor

Ted Huffmire
Co-Advisor

David Olwell
Chairman, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis describes the security risks for network-centric weapon systems as a combination of different aspects of security, each with its own threats and mitigation strategies. Computer and network security deals with cryptography, authentication, and attacks on software. Information security deals with the ability of the system to process information of different classifications but prevent disclosure to unauthorized users. Physical security ranges from hardware destruction to reverse engineering of captured hardware. Operational security covers the inability of covert units to transmit to the network without compromising their positions. Personnel security discusses the ways that people can intentionally or accidentally weaken the system during development or operations.

Security of network-centric weapon systems is now a System of Systems (SoS) engineering problem and system developers must therefore embrace a systems engineering approach to security and consider all the threats and vulnerabilities facing the system. This examination must include not only the technical characteristics of the components but also the people who operate and maintain the system and the requirements of the mission. Only by mitigating the most efficient attacks on the system, regardless of the type of attack, can developers maximize the overall security of the system.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author wishes to thank Professors Goshorn and Huffmire for their guidance during the writing of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The need for military commanders and units to communicate with each other has existed since antiquity. The creation of hierarchically organized armies led commanders to realize that their forces fought more efficiently when concentrated against a smaller force. These commanders developed tactics to flank enemy positions and deliver a concentrated attack. Execution of these tactics required the commanders to maintain an operational picture and communicate orders to their units, who executed the orders and reported their situations back to the commanders. Communication started with runners and flag signals, evolved into radio voice messages, grew to include written messages delivered by ground or satellite radio, and finally made use of tools like the U.S. Navy's Link-16, which allows real-time sharing of both friendly and enemy position data. Network-centric weapon systems seek to provide a technological and tactical leap in the efficiency of such communication.

A. BENEFITS AND CHALLENGES OF NETWORK-CENTRIC WEAPON SYSTEMS

Network-centric weapon systems seek to elevate the status of the latest communication tools from their current position as add-on peripheral gear to a prominent role as a critical element of every platform. They capitalize upon the greater communication capability provided by computers and allow every unit to use an operational display that shows the sum of friendly and hostile unit positions known across the entire force. The availability of this information drives a paradigm shift where units are considered nodes in a network rather than stand-alone actors. Most units serve as both users and providers of information, while stand-alone sensors serve as providers only and command centers serve as users only.

This paradigm shift from stand-alone actor to network member has its greatest impact upon human behavior. Equipped with the communications, processing, and interface equipment needed to provide a force-wide situational awareness display, the people in the system adapt their tactics, behavior, and organization to take full advantage

of their new awareness. Commanders use a more complete picture of the action to provide subordinate commanders and individual units with general combat goals instead of specific movement orders; operational units receive these goals and decide for themselves how best to maneuver and strike in a rapidly changing environment. Units also observe the status of both friendly and enemy forces and independently adapt their tactics to best collaborate with friendly forces. This self-synchronization and collaboration is one of the goals of network-centric warfare (Office of Force Transformation 2005).

Another goal of network-centric warfare is to make the command process faster and more robust. The powerful communications tools inherent in a network-centric system allow commanders to rapidly send situation updates, goals, and orders to the fighting forces. Provided with a complete operational picture and the set of combat goals, subordinate commanders gain the ability to immediately and seamlessly assume command duties when the original commander is disabled. The automatic sharing of information eliminates the need for lengthy turnover processes between the departing and relieving commanders and greatly reduces the chance that information could be dropped during the transition (Office of Force Transformation 2005).

Overall, network-centric systems seek to lift the "fog of war" and provide more efficient use of forces. When every ship, plane, vehicle, and infantry or special forces group knows the complete picture (the location and tasking of all friendly forces, the estimated position and strength of enemy forces, and the set of combat goals and progress made toward them), well-trained forces can apply themselves to maximize overall combat effectiveness. Implemented properly, the use of network-centric systems promises greater combat power provided by a smaller force with fewer friendly losses.

Such a comprehensive system faces serious implementation issues. Shifting to network-centric warfare requires a change not only in hardware and software but also in doctrine, organization, training, and logistics. Such a dramatic shift requires time, probably several years at least, to accomplish. The problem, then, is how to conduct operations during the transition period. How will the network-centric units coordinate their movements with units not yet converted? How will commanders lead their forces

when the network-centric portion expects to receive goals and information while the legacy platform-centric portion still requires specific movement orders and manual relaying of situational awareness information? The entire premise of network-centric warfare fails when major parts of the force are not connected to the network, so the upgraded units will likely be forced to operate as legacy units until the entire force is converted.

Even after the total conversion of U.S. forces, allied and coalition partners will remain outside the network, greatly complicating multi-national operations. Other nations will likely have an incompatible network architecture, if they have one at all, so U.S. forces will again have to revert back to legacy practices. Given that the transition to network-centric warfare includes sweeping changes in doctrine, operations, and training, this devolution back to legacy-style operations would force U.S. units to fight in an unfamiliar manner and without the benefit of their training, ensuring lower efficiency and greater friendly losses.

B. SECURITY RISKS

In addition to the logistical and operational challenges of transitioning to network-centric warfare, network-centric weapon systems also carry major security risks. The shift to a network paradigm, where each unit relies upon a valid connection to a functional network, means that a successful attack upon the network itself can degrade the entire force. These attacks can threaten any of the three critical features of an information system: confidentiality, integrity, and availability (Harris 2008, 59–61).

Attacks on the information system feature of confidentiality threaten to expose the entire friendly order of battle, tactical plans and goals, and knowledge of the opposing force. The last item provides perhaps the greatest benefit to the enemy because he now knows which of his units remain undetected and available for a surprise assault. An enemy equipped with this information has a powerful tactical advantage that can provide much greater effectiveness for his forces, which are now able to attack the most vulnerable parts of the other force while avoiding detection and prosecution of their own units.

Attacks on the information system feature of integrity threaten to sow chaos into the friendly force, even if few modifications actually get made to its database of information on unit positions, mission status, and sensor data.. The mere perception of tampering would destroy confidence in all network-provided information, causing hesitation in tactical movement and a flood of communications seeking to confirm position reports and identity. This apprehension among the friendly force would eliminate the very benefits of networked systems, self-synchronization and a fast and robust command element, and leave paralyzed units vulnerable to hostile maneuver. Even worse is the case of undetected tampering, where the enemy can trick friendly forces into firing on themselves while chasing false contacts and operating blind to real threats.

Attacks on the information system feature of availability represent the risk with the least severity but greatest likelihood. A friendly force trained and organized to operate as a cohesive whole could suffer major problems if suddenly deprived of the network and forced to operate independently. Enemy attacks can disrupt individual unit connections to the network or degrade the entire network itself using a wide array of computer, physical, and electromagnetic tools. Friendly forces could slide into hesitation and paralysis if suddenly forced to operate independently, providing an assertive enemy the opportunity to bypass friendly defenses and strike at their most important targets.

Protecting the security of network-centric weapon systems requires consideration of several interrelated aspects: computer and network security, information security, physical security, operational security, and personnel security. Attackers have at their disposal a broad array of techniques to disrupt the network, so developers, maintainers, and users must take a systems engineering approach to the problem of defending the network. This systems engineering approach considers all the vulnerabilities and the possible attack vectors that could exploit them in the context of the system's operational environment. It then considers the relative efficiency of these attacks and applies defensive resources to protect the most important and vulnerable parts of the network. Given the limits of funding and time during development, developers must use this

systems engineering approach to counter the most efficient attacks. The efficiency of an attack includes its ease of execution, the number of attack opportunities, and the amount of damage it can deliver.

System developers have two primary tools to ensure the security of the network. First, the hardware, software, and operating procedures must be developed with security as a primary requirement. Designing the network to meet functionality goals and then later trying to build in security, a common approach in the commercial world, is likely to leave serious vulnerabilities behind (Schneier 2000, 395). Flaws will remain because it is impossible to envision all possible attack techniques and mitigate them and because some security attacks rely on weaknesses in the system architecture itself. The design must include defense-in-depth features that minimize single points of failure. For example, a network could encrypt its communications and require authentication before accepting any messages, so an attacker who manages to crack the encryption scheme and send messages on the network would also have to figure out a way to impersonate a legitimate network user and spoof the authentication process. Second, each network component and the overall system itself must be subjected to rigorous testing and certification. This testing includes analysis of the software structure and coding, hardware examination, and a wide range of scenarios that target the users, interfaces, and other aspects of the system. Testing alone cannot guarantee the security of any complex system, but it can at least harden the system against the easiest and most damaging attacks (Schneier 2000).

The need to include security in the design phase of the system extends to the entire system and should not be limited to the hardware and software design. Network-centric weapon systems are systems-of-systems (SoS), so the security architecture must extend to each individual member system and to the interface to the overall SoS. Using a rigorous systems engineering approach for SoS will consider all the threats and vulnerabilities and reduce the risk that the entire system could be easily defeated through a previously un-considered attack on some peripheral part of the system (Office of the Deputy Under Secretary of Defense for Acquisition and Technology 2008). This effort must include not only the technical characteristics of the components but also the people who operate and maintain the system, the support elements that keep the system running,

and the environment in which the system will operate. Once designed, the tactics, doctrine, logistics, and personnel elements of the system must continue the systems engineering approach and consider security from the larger system perspective.

The systems engineering approach must also extend beyond the design phase into the entire system life-cycle. The earliest evaluation of technologies and doctrine must include the need for an inherently secure foundation (Higginbotham et al. 1998). For example, the Internet and its supporting protocols were designed for an environment where all users were inherently trusted, so use of these protocols may inflict military systems with the same security problems now faced by commercial and industry users (Harris 2008, 19–20). During the operation and support phase of the system, developers and commanders must consider the security risks that can come from operators, maintainers, and administrators and the consequences of deviating from security rules. This thesis will guide those developers and commanders, as well as operators and other stakeholders, in the acquisition and design of secure network-centric weapon systems.

This thesis makes a distinction between the computerized systems comprising a network-centric weapon system and the computer networks already in widespread use for office automation, collaboration, and communication. Most scholarship on computer security focuses on the latter type of system. These systems—the ones using x86-based processors from Intel or AMD, Windows or Unix-based operating systems, and TCP/IP connections to the global internet—form the foundation for commerce, education, and entertainment (Comer 1999). They experience regular security attacks from people seeking to commit fraud, steal proprietary information, or knock a website out of service (Schneier 2000, 1–5). Their components, being designed for widespread and general-purpose use, are inherently open to anonymous members of the public and must be hardened to prevent unauthorized access or malicious operations. Service-oriented architectures used for military purposes fall into this category and, although they share many of the security risks with network-centric weapon systems, they inherit the additional concerns faced by commercial networks.

Network-centric weapon systems, the focus of this thesis, may evolve from a collection of special-purpose hardware and software, such as the LINK-16 tactical data link, that inherently rejects unauthorized access and malicious operations (Camana 2009). Its operators will be trained in proper and secure operation and will be accountable members of the military. Even though the stakes may be higher with a weapon system than an office network, the developers may find it easier to secure the weapon system because of its limited scope and rigorous configuration management. Many of the same security principles still apply, however, and much of the work done to secure office systems can inform the development of weapon systems' security. These commercial practices become especially relevant if the network-centric weapon systems are constructed from the same commercial hardware and software components as the office networks, which may be necessary in order to keep the cost down and the schedule short enough to be deployable (Commission on Physical Sciences, Mathematics, and Applications 2000, 176).

C. EXAMPLE SYSTEMS

Network-centric weapon systems recently in development include the U.S. Army's Future Combat System (FCS) and the Navy's Cooperative Engagement Capability (CEC). Both systems illustrate the emergence of a new class of networked system. This new class is a hybrid of the office automation network and the data-linked combat system, and it requires a consideration of security principles tailored to its combined lineage. Developers must avoid both the blunt misapplication of inappropriate office-network security controls and the assumption that the system has no security risks due to its combat system characteristics.

The FCS as originally envisioned in Figure 1, combined ground vehicles, unmanned aerial vehicles (UAVs), and remote ground sensors into a single network. This network was to provide mission planning, situational awareness, and tactical coordination to operators and commanders. The vehicles included tanks, personnel carriers, mortars, cannons, command units, medical platforms, and unmanned units optimized for attack, reconnaissance, and logistics. Four types of UAVs and two types of

unattended ground sensors were to provide sensor data without risking harm to personnel (Globalsecurity.org). Due to cost and schedule problems, the Army has significantly revised and reduced the scope of the FCS program; however, some elements may yet emerge as deployable systems. Future weapon systems will probably inherit the network-centric philosophy of FCS, so the original program serves as a useful example of a generic network-centric weapon system of the future. This shift toward a network-centric weapon system will likely continue even with the shift in emphasis to counter-insurgency operations. Army Colonel John Buckley, a strategic planner for the Headquarters of the Department of the Army, said, "Indeed, there is a strikingly high correlation between the types of capabilities that commanders in Iraq and Afghanistan are requesting and the types of capabilities that are being developed through FCS" (Osborne 2009).

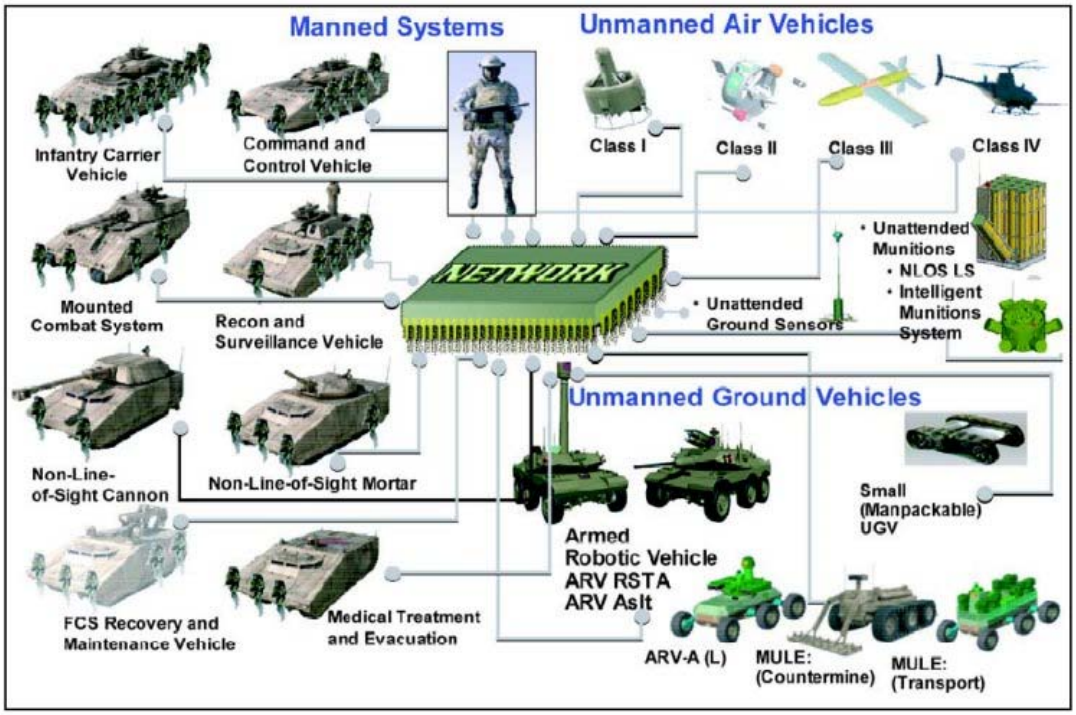

Figure 1. Future Combat Systems components (From *Defense Industry Daily* 2008)

The CEC provides a smaller example of a network-centric weapon system. CEC gives ships, planes, and air defense units the ability to share their radar sensor information and create a single, fused radar air track for each target. This sharing and fusing of sensor data allows all units to observe and react to any air target spotted by any

friendly unit that is connected to the network. It also improves track accuracy by giving increased weight to the data reported from more accurate radars (O'Neil 2007, 14). These functions of CEC are shown in Figures 2, 3, and 4. This application of network-centric philosophy provides only the situational awareness function for air contacts and includes no functions for status-sharing or command functions; however, it does provide a low-cost way to explore new tactics and find the best new capabilities to include in future systems.

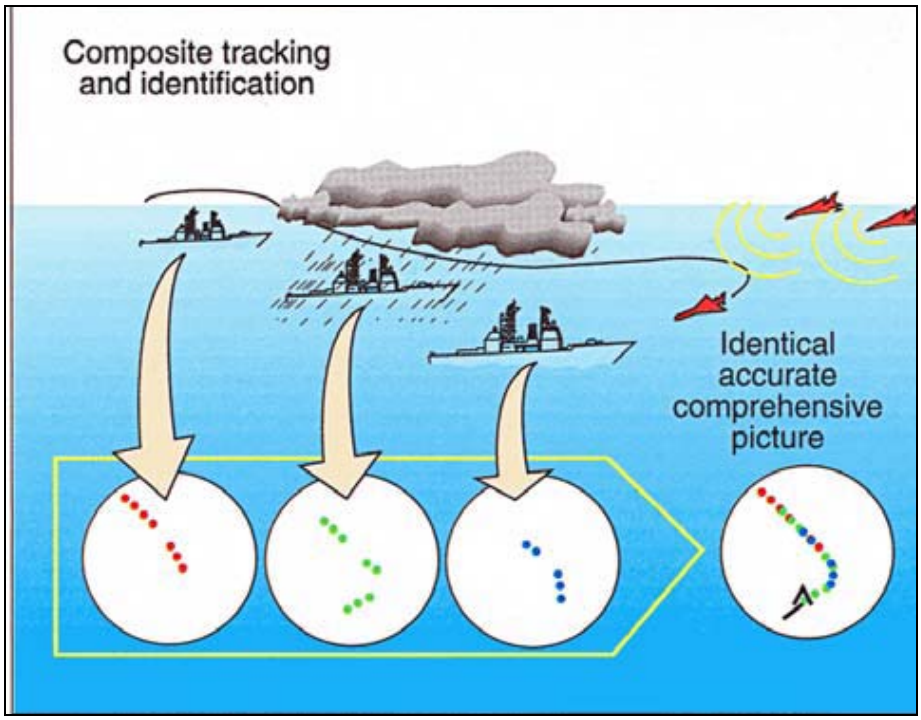

Figure 2. CEC composite tracking and identification
(From Johns Hopkins APL 1995, 379)

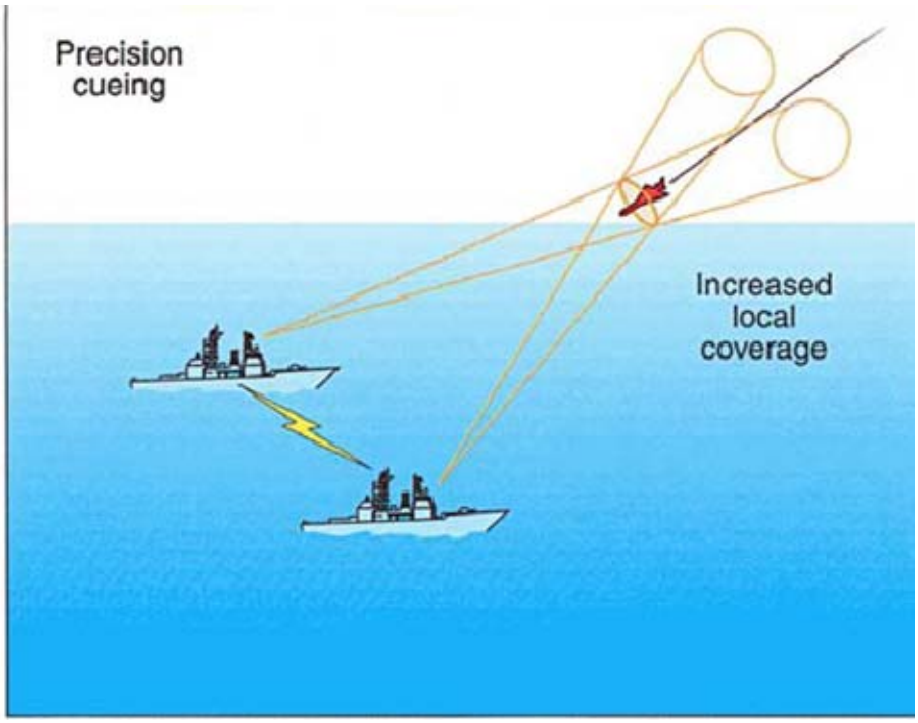

Figure 3. CEC precision cueing (From Johns Hopkins APL 1995, 379)

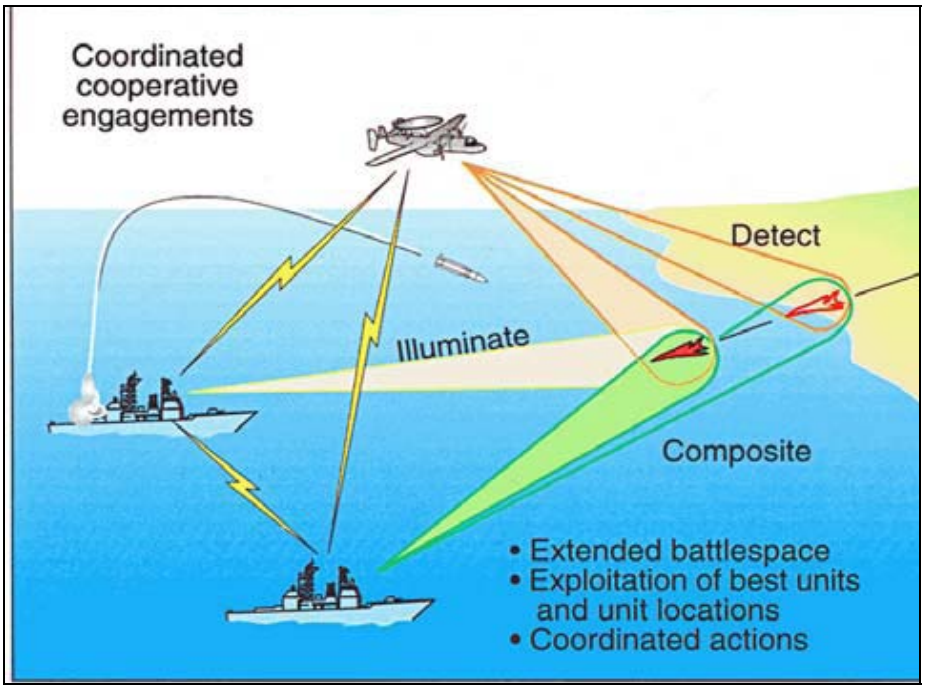

Figure 4. CEC coordinated cooperative engagements
(From Johns Hopkins APL 1995, 379)

The enormous cost and scope of the Army's FCS illustrate the complexity of network-centric weapon systems and the difficulty of implementing them properly. Security is costly, and as Department of Defense (DoD) policy urges developers to pursue the "good enough" solution and to save money by stopping short of the perfect solution (Dudney 2009, 2), developers risk catastrophic failure of the fielded systems if they avoid the rigorous design and analysis needed to provide that security.

D. OTHER STUDIES ON THIS TOPIC

Network-centric weapons systems are System of Systems (SoS) that require a systems engineering approach to their security considerations. Therefore, few studies exist on security as specifically applied to network-centric weapon systems. Discussions on computer system security for home, commercial, and industrial, and non-tactical military purposes dominate the literature. Many of the principles of these works on commercial-type security do apply to weapon systems, particularly those built from commercial hardware, software, and protocols, but tactical use brings some unique security risks that this thesis will enumerate. Three studies of note include a 2004 report titled "Issues and Requirements for Cybersecurity in Network-centric Warfare" by Drs. Stytz and Banks of the Air Force Research Laboratory; a book published in 2000 by the Commission on Physical Sciences, Mathematics, and Applications titled *Network-Centric Naval Forces: A Transition Strategy for Enhancing Operational Capabilities*; and a study underway by the Naval Studies Board titled "Information Assurance for Network-Centric Naval Forces."

The Air Force Research Laboratory report focused on computer and network security, describing both as attacks on software. The report provided technical details on these software attacks but did not consider the other sources of security risks covered in this thesis. The authors describe their vision as follows:

> Our vision for network centric warfare cyber security can briefly be described as calling for a seamless web of protection technologies for all levels and all portions of the network and software. The protection capabilities (and needs) range from data to network to software with all components being imbued with inherent capabilities to verify their own

> correct and secure operation as well as the correct and secure operation of the other interacting components of the cyber battlespace. (Stytz and Banks 2004, 4)

The book from the Commission on Physical Sciences, Mathematics, and Applications (part of the National Research Council) provides a detailed and comprehensive examination of the entire field of network-centric warfare. The chapter on Information Assurance includes such non-traditional risks as the espionage and sabotage threat during development, the vulnerabilities inherited from commercial hardware and software, and the danger of enemy eavesdropping and spoofing using captured hardware. The book embraces the systems approach to implementation, operations, and security, and provides an excellent companion to this thesis despite its focus on the Navy and Marine Corps. This thesis provides more of a systems engineering perspective as well as consideration of aspects such as operations security not covered in the book.

The Naval Studies Board project commenced in January 2008 and was planned to be a twelve-month study. The project's goals include an analysis of studies on information assurance, an evaluation of the acquisition process with respect to information assurance, an assessment of security vulnerabilities, and development of best practices. The project duration was extended to June 30, 2009; however, as of this writing, no reports have yet been released and the website for the National Academies still lists it as a current project. These reports are likely to provide useful information on the topic, though it is not evident whether the Board will have taken the systems approach or will have focused on technical aspects only.

E. PURPOSE OF THIS THESIS

This thesis will evaluate the most important security risks and requirements in the design of network-centric systems and services. To accomplish this, it will describe the difference between network-centric and stand-alone systems and consider how security for a network-centric system differs from commercial computer security. It will summarize the attack methods of concern, describe how these attacks can gain access to

systems, and explain the consequences of successful attacks. It will recommend features and practices to reduce the security risk and discuss the DoD requirements for testing and certification that enforce the use of security features.

This examination of security considerations for network-centric weapon systems will help guide the acquisition process for network-centric and Service Oriented Architecture systems. It will evaluate the current DoD security directives and their applicability to network-centric weapon systems. It will also provide source material for use in the development of Naval Postgraduate School resident and certificate courses on network-centric system development.

In this thesis, the following Chapters II–VI describe the various aspects of security most likely to factor into the design of network-centric weapon systems: computer and network security, information security, physical security, operational security, and personnel security. Chapter VII provides information on combining these disparate elements into an optimized system solution that provides the greatest security for the lowest cost. Chapter VIII provides an overall conclusion.

THIS PAGE INTENTIONALLY LEFT BLANK

II. COMPUTER AND NETWORK SECURITY

This chapter on computer and network security deals with technological attacks against the execution of code and the transmission and storage of data. Since network-centric systems will be highly computerized, this part of system security captures the greatest share of attention and resources both by industry, as suggested by the preponderance of literature on the subject, and by this thesis. It also presents the greatest engineering challenge since the problems and solutions focus on technical aspects of system operation rather than the operational and procedural aspects dealt with in other parts of system security.

The fundamental security challenge at the heart of any computerized system is one of abstraction. Levels of abstraction include the network level, component level, and others depending upon the specific design. Computers do exactly as commanded by their programming and processor design, but users have almost no insight about the actual operations taking place within the microprocessors, memory chips, and other hardware. Users typically only see the user interface, which represents only a vague abstraction of the billions of operations per second acting upon the billions of bytes of data and instructions in a typical computer. As noted security expert Bruce Schneier stated,

> People don't understand computers. Computers are magical boxes that do things. People believe what computers tell them. People just want to get their jobs done. (Schneier 2000, 255)

The security risk here is that the user interface reports only what it is commanded to report by the underlying application, not the actual series of events. Computers can lie just as easily as they can tell the truth, and users have few ways to find the lies.

Attackers capitalize upon the enormous complexity of computer operations that requires such a profound abstraction in order to communicate with users. Simple programs in simple systems have few opportunities for attackers to interfere with this trusted user-computer relationship; unfortunately, any system that meets the extensive requirements of a network-centric military force would be complex and therefore vulnerable. System designers can establish a security architecture that reduces

complexity by defining discrete process and data elements and their interfaces, reducing the number of entry points for an attack. Such a design follows the software engineering principles of loose coupling and high cohesion (Larman 2006).

Attacks on the computer network can threaten that system's confidentiality, integrity, and availability. Confidentiality attacks involve reading secret information by intercepting network communications or by implanting code that grabs secret data and leaks it to the enemy. Integrity attacks involve the injection of false information into a trusted network link or the direct alteration of data stored within the system. Availability attacks involve the interruption of the system's operations or its ability to communicate. Some attack techniques combine these three elements and can be used to produce a wide variety of effects and extensive damage (Committee on National Security Systems 2006).

A. ATTACKS ON THE NETWORK

Outsiders wishing to attack a network-centric system generally start by attacking the communications between network nodes. The simplest type of network attack, jamming, relies upon the fact that a tactical military network would use electromagnetic transmissions to send most of its information. Jamming has been practiced for decades, and it simply involves the radiation of electromagnetic noise on the same carrier frequency in use for communication. Sophisticated, modern jammers can silently listen for transmissions and quickly send bursts of energy at a matching frequency, cutting off whatever message the target system intended. Jamming intends to disrupt the availability of the system by making it impossible for network nodes to communicate with each other.

Anti-jam techniques have also evolved over the years, and they often involve spread-spectrum transmissions with error detection and correction (Anderson 2008, 567–72). Spread-spectrum transmissions involve rapid changes in the carrier frequency (about 77 thousand times per second in the case of Link-16), forcing the attacker to spread the jamming energy over a broad range of frequencies and therefore reducing the jamming effect on any given frequency (Carmana 2009, 1-AM-62). The frequency selection for the next hop relies upon some secret data shared among the network nodes, denying the

enemy the ability to predict the next frequency. Interleaving, the re-arranging of data bits within a message, can be used with error detection and correction algorithms such as Hamming codes or Reed-Solomon codes to provide redundancy in the data stream that allows the receiver to reconstruct a message containing a small number of missing or flipped bits (Carmana 2009, 2-AM-80). In addition, several adaptive filtering techniques, such as notch filtering, adaptive beamforming, nonlinear adaptive filters, etc. have been developed for anti-jam applications (Haykin 2002).

Outsiders can also attack the confidentiality of the target system by eavesdropping on network communications. Assuming no security, electromagnetic transmissions can be received by friend and foe alike, so attackers need only to have a receiver tuned to the correct frequency and placed in the same geographic area. The needed size and design of this receiver and the proximity to the transmitter depend upon the transmission frequency and power as well as terrain characteristics. Landline communications can be observed through the use of a cable tap. These taps can physically penetrate the cable, for electrical or optical cables, or can measure electromagnetic induction for electrical cables only (Anderson 2008, 530). Protecting against eavesdropping, therefore, cannot rely upon denying the enemy the ability to receive and copy transmissions.

Protection against eavesdropping must use encryption to disguise the true content of friendly transmissions. Encryption has been in use for thousands of years and has exploded in complexity since the appearance of the computer, but the two fundamental elements have remained the same: the key and the algorithm. The key is simply a string of data shared in secret among the network nodes. The algorithm, or cipher, is the set of procedures for altering the message, or plaintext, and turning it into ciphertext that can only be decoded by someone who also possesses the same key and algorithm (Harris 2008, 666).

Keys come in two forms, symmetric and asymmetric. Symmetric keys, together with an appropriate cipher, perform both encryption and decryption. They operate simply and quickly but require the secure distribution of keys in advance. They also require separate keys for every needed combination of participants since anyone with the key can read the conversation. Asymmetric keys exist as matched pairs of keys, with one key

performing encryption and the other key of the pair performing decryption. Users keep one key private and publish the other one for other network members to use. Asymmetric encryption is much slower and more complex, but it requires no advance key distribution and requires only one key pair for each node (Harris 2008, 679–83).

Ciphers use a combination of substitution, transposition, and other mathematical manipulation to achieve the desired degree of confusion and diffusion in the ciphertext. Substitution involves replacing one character for another and is popular in one-time-pad ciphers, where the key is the same length as the message and messages are often encoded and decoded by hand. Transposition involves shuffling the order of the characters, similar to a word-scramble puzzle (Harris 2008, 679). Confusion refers to making the relationship between the key and the ciphertext as complex as possible, and diffusion refers to making the relationship between the plaintext and the ciphertext as complex as possible. Together, they ensure that small changes to the key or plaintext result in large changes to the ciphertext, making it difficult for an attacker to crack the code (Shannon, 1946, 708–709).

Computer-based ciphers tend to operate at the bit level instead of the character level of ancient ciphers, so more complex mathematical operations become available. Many ciphers use the "exclusive or" as a fast and reversible way to encrypt a data string according to the key. Complex ciphers borrow the concepts of substitution and transposition by applying them to blocks of data and then sometimes chaining these blocks together and using the ciphertext from one block to form a new key for the next block (Harris 2008, 685–7). Ciphers for asymmetric keys can rely upon such irreversible operations as modular arithmetic (taking the remainder of a division operation) to ensure that an enemy cannot decrypt the ciphertext using the public half of the key pair (i.e., the key used for encryption) (Anderson 2008, 171).

The challenge for the attacker and an encrypted message becomes the decryption of an intercepted message. The attacker needs the same resources needed by the legitimate recipient: the key and the cipher. Users always keep their keys secret, and they often change them on a regular basis to limit the damage in case a particular key gets compromised. Keys can be stored in the memory of the computer, on a storage device

such as a floppy disc or USB drive, or on an external processor such as a smart card. Smart cards such as the military Common Access Card have the advantage that they never transmit the key onto the computer being used for communication, where an attacker who gained access to the memory or file system could steal it. Rather, they receive the plaintext from the computer, perform the encryption on the smart card's processor, and send the ciphertext back to the computer for transmission. Smart cards lack the standard file system of a general-purpose computer, making it much more difficult for an attacker to copy the key from the card's memory (Harris 2008, 191).

While keys are always kept secret, the cipher may not. The security community has two distinct schools of thought regarding the secrecy of ciphers and computer code in general. One side supports open-source, or public release, because scrutiny by the world community can reveal flaws that escaped the attention of the designers. Ciphers are difficult to keep secret, and their compromise may allow adversaries to discover new flaws and execute attacks. The other side supports secrecy of ciphers because an attacker must know the cipher in order to perform a cryptologic attack. Even a weak and flawed cipher can provide adequate protection if the adversaries do not understand how it works (Harris 2008, 668).

The decision on whether to release ciphers or make them secret depends in part upon which of the following scenarios seems more likely. On one hand, an adversary might examine a public cipher and find a flaw that system designers and the worldwide cryptographic community all overlooked. On the other hand, an adversary might steal a secret cipher and discover a vulnerability that system designers failed to catch. This decision depends in part upon the development methods used for the cipher. Ciphers produced by a contractor, even one operating in a classified environment, may fall victim to the widespread threat of industrial espionage. Ciphers produced completely within the National Security Agency, on the other hand, may better resist the industrial espionage threat.

Other security risks against secret ciphers can also influence the public/secret decision. Secret ciphers may be discoverable by reverse-engineering attacks upon hardware that has fallen into enemy control. Once compromised, a cipher (unlike the

encryption key) cannot easily be changed; it may take several years to develop a suitable replacement. Finally, an adversary that discovers the cipher may share it with any number of parties hostile to the United States. Secrecy of ciphers, and of code in general, may help to confound an attack, but the overall security of the system and its messages must not rely primarily upon that secrecy. This idea was first proposed in 1883 by Auguste Kerckhoffs, who said that the only secrecy should be in the key and that the algorithm should be publicly known (Harris, 2008, 668).

Even a mathematically perfect cipher may fail to protect the confidentiality of a message if an attacker can exploit a vulnerability in the way the cipher is implemented in hardware or software. In the early 1960s, the National Security Agency discovered that attackers could read encrypted messages by measuring the electromagnetic emanations coming from communications hardware. Code-named "Tempest," this program also led to the realization that emanations depended on the combination of components and how they were connected—a far more complex challenge than studying individual components in isolation (Boak 1973, 98). Attackers can also use timing analysis or power analysis to figure out the cryptographic key and other protected information by carefully measuring the time or electric current used by the computer performing the encryption/decryption (Anderson 2008, 533).

A perfectly designed and implemented cipher may protect the contents of network messages, but adversaries may still deduce some important information by conducting traffic analysis. Traffic analysis is the study of the identity of the sender and receivers of messages as well as the timing and number of messages being sent. For example, a spike in traffic from a particular unit may imply that it has located enemy units or is starting an attack. Friendly forces can defeat traffic analysis by sending decoy traffic to disguise patterns in genuine messages (Harris 2008, 1087).

Network-based attacks on integrity extend the techniques used when compromising confidentiality. Where an attack on confidentiality copies a message and decodes it to learn its secrets, attacks on integrity seek to use the target's network to inject messages with false information. This false information could be used to hide the

attacker's forces, create diversions, or even cause fratricide by tricking the opposing force to fire on itself. The adversary must understand the target's communications infrastructure, protocols, ciphers, and encryption keys in both situations.

Attackers use a wide variety of approaches to inject false information. At the simplest level, the attacker mimics a legitimate network node and sends properly-formatted but false data. This impersonation attack can create false target data, orders, or status reports. More complex attacks use session hijacking or "man-in-the-middle" techniques to exploit a legitimate communication session between two network nodes. Session hijacking requires the attacker to observe a session in progress and then conduct an impersonation-type attack by mimicking one of the participants, assuming its identity in the conversation. If the node whose identity is being hijacked can be silenced, the attacker can continue the session while masquerading as that node. Man-in-the-middle attacks work only when the two network nodes under attack cannot directly communicate with each other, as is the case over a wired connection where the wires pass through a node under enemy control. In this case, the enemy hijacks the session in both directions, capturing all traffic and making changes as desired (Harris 2008, 1084–6).

Authentication tools can foil attacks on integrity by forcing network nodes to prove their identity before participating in a communications session. These tools typically use the cryptographic key as a "shared secret," where possession of the key proves the person's identity. Both symmetric and asymmetric keys can serve this function, with the same logistical considerations as when they are used to protect confidentiality. In the case of asymmetric encryption, the message sender uses his own private key to encrypt some portion of the message. Properly decrypting that portion with the sender's public key verifies the message sender's identity because only the sender possesses the private key needed for encryption (Harris 2008, 682).

Message integrity tools can also foil attacks on integrity by making it impossible for the attacker to alter the message content without those changes being detected. This task generally employs a cryptographic hash function, which uses a complex algorithm to distill any size document down to a hash value. These hash values are typically on the order of 128 bits long, and they are sent with the message to prove that the message has

not been altered. Well-designed hash functions cause the hash value to change dramatically at the slightest change of the original file, and a large hash value makes it infeasible for the attacker to find some modification that would produce exactly the same hash value (Harris 2008, 714).

To check message integrity, the message recipient simply runs the hash function on the received message and compares this generated hash value with the hash value sent with the message. A match proves that the message has not been altered. Of course, a wily attacker could simply change the message, generate a new hash value, and send the new hash value along with the altered message; however, encrypting the hash value prevents that from happening since the attacker cannot properly encrypt the bogus hash. In addition to protecting a message from undetected changes, an encrypted hash value links the message to its sender and therefore serves as a digital "signature" used for authentication of the sender (Harris 2008, 714–7).

Hash functions go beyond simple error detection and correction techniques like cyclic redundancy checks, checksums, or Hamming codes. These techniques protect messages from accidental modification during transit caused by electromagnetic interference or processing errors, but an attacker could easily create checksum values that match his altered message. Network nodes must not rely on error detection and correction techniques to protect messages from malicious modification.

A public key infrastructure (PKI) formalizes the process of sharing and validating public and private keys so that message recipients can trust that messages report their true originators and contain unaltered content. Under a PKI, each user has a digital certificate issued by a trusted Certificate Authority (CA). These certificates contain the user's identity, his public key, the CA signature, and other protocol information. The CA signature consists of a hash of the certificate data encrypted by the CA's private key, and it proves that the certificate is legitimate. So long as all participants trust the CA, any certificate signed by that CA can be trusted and used to validate the digital signatures of received messages. Larger PKIs may have multiple layers of CAs to issue large numbers of certificates, but each subordinate CA in this situation has its own certificate signed by the one inherently trusted "root" CA (Harris 2008, 725–32).

Commercial computer networks, including home users accessing merchant websites on the Internet, use a PKI to establish encrypted connections and verify the identity of participants. This commercial application of PKI has a significant weakness because compromised certificates can still appear to be valid. The original intent for PKIs was to keep certificate revocation lists (CRLs) that list any digital certificates that should no longer be trusted. As the Internet grew, however, CRLs grew too cumbersome to update and check (Harris 2008, 728–9). Military networks have an advantage compared to the Internet because the closed and limited nature of military networks permits an effective use of CRLs and a more reliable defense against compromised certificates.

The final type of attack upon integrity involves recording a legitimate communication session and replaying it later. Since the recorded information contains all the correct cryptographic keys, formats, and hash values, it would appear to the receiver to be a legitimate message from the original sender. The attacker likely would not even know the contents of the message he captured, but injecting "stale" information could cause confusion for the receiver. Network nodes, therefore, need some method to ensure that replayed messages are rejected.

A combination of time stamps and nonces guarantee the originality of friendly messages and defeat the replay attack. A time stamp shows the age of the original message, and receivers can automatically reject messages older than a chosen age. The nonce, which is simply a randomly selected number, proves the uniqueness of each legitimate message. When all network nodes agree on a certain age threshold beyond which they will reject incoming messages, then they simply need to read the nonce of each message they receive and compare it to a list of received nonces within the age threshold. Since each legitimate message will have a unique nonce, any message that carries the same nonce is rejected as a replay (Anderson 2008, 66–7; Harris 2008, 756). This assumes that the attacker cannot decrypt the message, replace the nonce with a new value, and re-encrypt it; however, any attacker with the ability to perform those steps could launch much more damaging attacks than a simple replay.

B. ATTACKS ON THE COMPUTER

Once an attacker gains access to a computer within a network, a vast array of attacks become possible. Where a network attacker operates as an outsider interfering with message flow and possibly reading secret messages and injecting false information, a computer attacker assumes the much more dangerous position of an insider capable of altering both data and programming code. Any attack technique can potentially interfere with availability, confidentiality, and integrity in any combination.

Gaining direct access to one of the computers within the network can be achieved through network or physical access. Network-based access starts with one of the integrity-based network attacks discussed above and involves the attacker injecting some sequence of commands that exploits a remote-access feature of that host (Anderson 2008, 636). These remote-access features may give network administrators the ability to control or configure network nodes from a central location or to update computer code following the discovery of a security vulnerability or some other kind of bug. Physical access involves actual possession of the target computer and manipulation of memory, cache, or processor registers to install malicious code or false data. Further information about physical access will be presented in Chapter IV.

The number of vulnerabilities available on a computerized system depends largely upon how much of the system uses general-purpose hardware and software. In order to save money, develop systems faster, and take advantage of the latest technology offered by industry, program managers frequently use commercial hardware and software as the foundation of their developing systems. Developers take these general-purpose processors and software and program them to perform the functions needed by the new system. Unfortunately, these items retain the inherent ability to perform any function if so programmed, so attackers can potentially alter a device's programming to change its behavior. Adding to this vulnerability is the fact that many commercial applications are not designed with rigorous security as a top requirement and therefore can contain vulnerabilities that ease an attacker's access to the code. Commenting on the general lack of security in commercial software products, Bruce Schneier wrote,

> The costs of adding good security to software products are essentially the same ones incurred in increasing network security—large expenses, reduced functionality, delayed product releases, annoyed users—while the costs of ignoring security are minor: occasional bad press, and maybe some users switching to competitors' products. The financial losses to industry worldwide due to vulnerabilities in the Microsoft Windows operating system are not borne by Microsoft, so Microsoft doesn't have the financial incentive to fix them. If the CEO of a major software company told his board of directors that he would be cutting the company's earnings per share by a third because he was going to really—no more pretending—take security seriously, the board would fire him. If I were on the board, I would fire him. Any smart software vendor will talk big about security, but do as little as possible, because that's what makes the most economic sense. (Schneier 2004)

In contrast, systems built with single-purpose hardware such as application-specific integrated circuits lack the ability to deviate from their programming (Null and Lobur 2006, 716). To use this option, developers must capture the functionality of the system and build those functions into the hardware itself. While appealing from a security perspective, this option presents major problems with development. Development of complex computerized systems uses an iterative approach, where functions are built, tested, integrated, and tested again (Larman 2006). Each round of testing can uncover flaws in the programming that must be fixed for the next version. Furthermore, users often change their requirements during and after development as they gain proficiency with the new system and conceive of new functions. Making all those changes in software alone is a reasonable task—one with which most developers are familiar since the commercial and educational environments use those skills and techniques. Trying to revise the design of an integrated circuit and produce a new sample for each round of testing, however, would be prohibitively expensive and time-consuming and is not a practical approach.

Given the need to build military systems upon general-purpose commercial hardware and software, developers must devote their attention to those remote-access features that can be exploited by an adversary. These features include remote-update

capabilities that allow software updates in the field, remote-configuration tools that allow control and configuration of deployed elements, and communication links used to exchange sensor data and command and control information.

Remote-update capabilities, where system administrators push software updates to network hosts, present a major security challenge to the integrity of executable code. These features rely upon authentication and integrity techniques to ensure that only appropriate connections are allowed and that the received data has not been altered. An adversary who exploits some weakness in these protective measures and impersonates a trusted network member can gain the ability to reprogram the victim node to perform any function desired by that adversary. These programming attacks can shut down the node, corrupt data, or open a communications channel back to the adversary to relay network information (Harris 2006, 1044).

Remote-configuration tools present less of a threat since they lack the infrastructure to directly alter the executable code. These functions would be especially relevant to unmanned vehicles and sensors that cannot be commanded by a physically-present operator. An adversary impersonating an authorized network member could shut down or otherwise mis-configure these vehicles and sensors.

Absent any direct access to the executable code or configuration functions, as discussed in the previous chapter, attackers could still exploit weaknesses in the communications links that form a fundamental part of any network-centric system. These weaknesses can include unusual or boundary conditions, buffer overflows, and injection attacks. Each of these scenarios involves some malicious string of data being passed to the victim node and having some unexpected and detrimental effect.

Attacks involving unusual or boundary conditions rely on the fact that complex systems have so many possible permutations of incoming data that developers cannot test them all. An algorithm written with the assumption that incoming data will conform to certain ranges and formats, may produce an unstable condition when the incoming data fails to obey those assumptions. For example, the F-22 fighter's avionics system failed

when the plane crossed the international dateline because of a programming error (CNN 2007). While most examples of this type of problem involve functionality failures, an attacker could inject data strings to produce similar results.

Additionally, buffer overflows have been used for decades and still present security risks to computers. Buffers are the memory storage areas reserved by the program to receive incoming data. If the incoming data transmission is larger than the buffer space, that data could possibly overwrite some of the executable code. A well-crafted buffer overflow attack can cause the program to execute part of that transmitted data as program code, permitting the attacker to conduct a small-scale reprogramming and insert malicious code such as viruses or worms (McClure, Scambray, and Kurtz 2005, 218–20).

Even when systems prevent buffer overflows, injection attacks can take advantage of systems where commands are transmitted together with the data. These systems use specific alphanumeric strings, special characters, or some other mechanism to differentiate command signals from data (McClure, Scambray, and Kurtz 2005, 561–2). An attacker with network access and knowledge of these command signals could perform the same kind of remote-configuration attacks discussed earlier even if the system lacks a dedicated control mode.

Defense against attacks that use unexpected inputs, boundary conditions, buffer overflows, or command injections requires developers to consider all possible combinations of input data in code analysis and testing. Developers should also use bounds-checking functions that filter out ranges of data that could never be received in an operational environment. The field of software safety provides useful ideas for protecting security by suggesting the operation of a separate, parallel function that monitors system behavior and takes action when the system does something unexpected or unsafe. Many networks already use intrusion detection or prevention systems to perform this independent monitoring function. Developers must also scrutinize commercial code intended for use in the system, remove any unnecessary functions and interfaces, and insert bounds-checking functions where needed (Joint Software System Safety Committee, E-10).

In addition to the scenarios above where attackers try to inject bogus data or malicious commands into the system, many other avenues of technological attack exist. The specifics of these techniques are beyond the scope of this thesis and depend upon the hardware, software, and protocol configuration in use. Some of the risk comes from the same viruses, worms, and other malicious code that regularly threatens commercial computer networks. If the network-centric weapon system includes commercial hardware and software, it must include the same defensive measures used by commercial networks to prevent malicious code from executing. These measures include strict configuration management that requires new code to be tested prior to installation, the use of access controls so operational users cannot install software, and anti-virus software running on developmental machines to prevent viruses from attaching themselves during system development.

Developers must decide whether their executable code and algorithms should be kept secret or made available to the public. This decision mirrors the same issue with cryptographic algorithms and has the same risks and benefits to both options. The DoD has begun to recognize the value of open-source software, where the source code is published for everyone to see and members of the public can contribute code via the project team, and has established the "Forge.mil" Web site. This website builds on the open-source model of the SourceForge website that provides the infrastructure for community collaboration on software projects. Using the "Forge.mil" website, developers of military systems can subject their code to public scrutiny and benefit from their observations and contributions.

Security for executable code (i.e., the ability of software to resist attacks) depends, just as in the cryptography scenario, upon the secrecy of configuration settings and cryptographic keys. Developers who choose to keep their algorithms secret may achieve some additional "security through obscurity," but they must not rely upon the algorithm or code secrecy to provide security. Any elements that provide security must be easily and quickly replaceable in the event these elements are compromised, and the source code would require a great deal of time and effort to replace.

Much of the security effort for computerized systems focuses on managing complexity. Developers use tools like security policies and security architectures to understand the structure of the system and control the ways that different components can interact. Since attackers can break into the system through poorly designed or mis-configured interfaces, these tools help minimize the complexity and remove unnecessary interconnects.

The security policy defines the system's requirements with respect to security. It identifies and prioritizes assets necessary to carry out the military missions, establishes the goals regarding protection of information, and defines organizational duties and authorities. The security policy emerges from an examination of the threats against the system and serves to defend against those threats—but only where the risk exceeds the cost of the defense. Broad policy statements act as the basis for security standards and guidelines, which create rules governing the legal behavior of the system. The standards and guidelines provide the basis for detailed operating and administrative procedures that direct the actions of operators and maintainers (Harris 2008, 110–5).

The security architecture provides an abstract representation of the components and interconnections of a system. It captures the functions performed by each component and the data flow throughout the network. Developers use a security architecture to illustrate which components need to communicate and which ones must be isolated. Designing the components around the architecture helps to ensure that modules communicate efficiently without violating the security policy. Defining and using the security architecture is a key effort in systems engineering that helps developers reduce system complexity and better understand the system's design (Anderson 2008, 657–9).

One option for a security architecture is the Trusted Computing Base (TCB). The TCB represents the combination of all security protection mechanisms within the system, including hardware, software, and firmware. By using a TCB, developers can create a small, simple, and reliable construct that manages system processes in a way that enforces the security policy. Capturing the protection mechanisms in the TCB allows developers of the rest of the system to devote their effort to the features, reliability, and

usability aspects of operation without having to worry as much about security. It also allows functionality changes to be installed without changing the security posture of the system (Harris 2008, 321–6).

A TCB could include a reference monitor that mediates access between users and data files. It could also provide for a security kernel as the hardware, software, and firmware within the TCB that implements the concept of the reference monitor. The reference monitor concept plays a major role in protecting the confidentiality of systems that process information of different classification. Chapter III describes the reference monitor in greater detail.

C. TESTING AND CERTIFICATION

Testing and certification for security differs from testing and certification carried out to prove the functionality of software. In the functionality case, tests capture the most likely scenarios, configurations, and data communications. Failures of functionality tests can be easily detected through improper system behavior or incorrect data processing. Because security problems often emerge from unexpected situations and since testing alone cannot capture every possible permutation of inputs and system conditions, security certification must also rely upon an examination of the program design. This examination looks for both the presence of security protective features and the proper design and implementation of the code. Most importantly, it verifies that the security policy, which specifies the legal behavior for the system, is properly implemented and will be enforced.

To test and certify the computer and network security of a system under development, industry and the DoD often rely upon an international program called the Common Criteria for Information Technology Security Evaluation (CNSS 2003, 2). This program, sponsored by the U.S. National Institute of Standards and Technology and National Security Agency, in addition to similar organizations in 25 other countries, provides a framework for evaluating the security of both hardware and software. The Common Criteria evaluation studies the degree of assurance that the product under consideration properly implements its security functions. Each evaluation tests a product,

called the Target of Evaluation, (TOE) in one specific configuration and operational environment. A document called a Security Target defines the security properties of the TOE and may include one or more Protection Profiles, which capture the implementation-independent security requirements for a given class of device. These security properties in the Security Target document include Security Functional Requirements, which specify functions that the TOE must perform, and Security Assurance Requirements, which describe the measures taken during development to ensure that the security functions will operate properly (Signatories of the Common Criteria Recognition Agreement 2009).

The evaluation verifies that the Security Functional Requirements are satisfied and rates the Security Assurance Requirements by providing an Evaluation Assurance Level (EAL) of one to seven, with seven being the highest assurance. Higher levels require increasingly more rigorous and formal design and testing and an accordingly higher cost for development. The formal mathematical methods used to prove the security of systems seeking a high EAL drive developers to design those products in accordance with the evaluation criteria (Signatories of the Common Criteria Recognition Agreement 2009).

The evaluation also verifies that the security policy has internal consistency, the security architecture faithfully matches the policy, and the design correctly implements the architecture. These processes take a great deal of time and money, delaying product deployment and driving costs beyond the ability of some programs to handle. The time and cost of an evaluation grows with both the EAL desired and the size of the TOE, so schedule and budget pressures may drive program managers to settle for a lower EAL than they might otherwise desire. Evaluations oriented toward a lower EAL skip the formal mathematical verification of the security model and present a risk that the model may contain vulnerabilities (Signatories of the Common Criteria Recognition Agreement 2009).

The evaluation process also complicates the configuration management of the deployed system. Since the evaluation examines the TOE in one specific configuration and environment, changes to the code may invalidate the evaluation and require an

update. A program using incremental and evolutionary development methods may not be able to afford the time or money required to re-evaluate each version of the system. Use of a TCB with a security kernel and reference monitor may avoid these problems by grouping functionality changes outside the part of the system that requires security evaluation. The conflict between the desire for agile development and the need for evaluated security requires further research in the field of network-centric system acquisition.

Testing the hardware and software for security flaws provides some assurance about system security; however, it cannot guarantee the security of the system against all technical attacks. Attackers have an effectively infinite variety of methods at their disposal, making it impossible to test for all possible attacks. The limited value of testing underscores the value of a secure design, making essential for developers to include security requirements in all stages of design rather than simply trying to add security features to an already-built product. As Bruce Schneier noted,

> The real lesson is that the patch treadmill doesn't work, and it hasn't for years. This cycle of finding security holes and rushing to patch them before the bad guys exploit those vulnerabilities is expensive, inefficient, and incomplete. We need to design security into our systems right from the beginning. We need assurance. We need security engineers involved in system design. This process won't prevent every vulnerability, but it's much more secure — and cheaper — than the patch treadmill we're all on now. What a security engineer brings to the problem is a particular mindset. He thinks about systems from a security perspective. It's not that he discovers all possible attacks before the bad guys do; it's more that he anticipates potential types of attacks, and defends against them even if he doesn't know their details. I see this all the time in good cryptographic designs. It's over-engineering based on intuition, but if the security engineer has good intuition, it generally works. Kaminsky's vulnerability is a perfect example of this. Years ago, cryptographer Daniel J. Bernstein looked at DNS security and decided that Source Port Randomization was a smart design choice. That's exactly the work-around being rolled out now following Kaminsky's discovery. Bernstein didn't discover Kaminsky's attack; instead, he saw a general class of attacks and realized that this enhancement could protect against them. Consequently, the DNS program he wrote in 2000, djbdns, doesn't need to be patched; it's already immune to Kaminsky's attack. That's what a good design looks like. It's not just secure against known attacks; it's also secure against unknown attacks. We need more of this, not just on the internet but in voting machines, ID

> cards, transportation payment cards ... everywhere. Stop assuming that systems are secure unless demonstrated insecure; start assuming that systems are insecure unless designed securely. (Schneier 2008a)

Computer products used for the DoD also require certification and accreditation in accordance with DoD Directive 8500.01E and Intelligence Community Directive 503. While most of the technical work in this process examines the computer and network security of the system, the certification and accreditation process is designed to evaluate the security of the entire system. Details on this process will be provided in the Chapter VII discussion on whole-system security.

THIS PAGE INTENTIONALLY LEFT BLANK

III. INFORMATION SECURITY

A shift to network-centric warfare would require the interconnection of combat units, sensors, logistics elements, intelligence analysts, and commanders. Each of these units provides its own type of data to the network, ranging from unclassified data on spare parts supplies, to secret data on unit positions and status, to top-secret "sensitive compartmented information" (SCI) data from surveillance satellites and covert sensors. However, not all users of the network have the security clearance and need-to-know required to gain access to all of that information. This discrepancy between the sensitivity of information on the network and the authorization of network users to read, create, and modify that information means that the system must use rigorous controls to limit information access to authorized users. Failure to enforce these controls can lead to information "spillage" outside of its approved domain and compromise of sensitive information to hostile forces.

Office automation computer systems used by the military usually perform this access limitation function through the use of Multiple Independent Levels of Security (MILS). This practice requires the creation of a separate network for each classification level. Users wishing to exchange unclassified logistics or business information use an unclassified computer. Those same users move to a secret-level computer to work on secret operational information and then move to a top-secret-level computer to process top-secret intelligence information. The MILS architecture assumes that any user of a given system has the proper security clearance and formal access approval for all of the information on that system. This security mode of operation is called "dedicated mode" when all users also have a need-to-know for all information and "system high mode" otherwise. Systems containing SCI or special-access program information may operate in "compartmented security mode" when users have different formal access approvals and need-to-know (Harris 2008, 354).

The MILS architecture works well for office workers who have room for multiple computers on their desks or in their workspaces. It also works well where each type of information exists entirely on only one of the networks and does not span multiple

systems. MILS presents major logistical problems, however, when users lack the ability to use multiple computers at one time or when information flows across different security domains. Operators of airplanes, tanks, and infantry units lack the space for multiple computer systems, and their operations naturally combine data of many classifications and compartments onto a single operational picture.

Network-centric weapon systems could solve the logistical and operational problems of the MILS architecture by using a Multi-Level Security (MLS) architecture. MLS allows data of different security classifications to be combined onto a single computer system and network. To protect data confidentiality, MLS systems assign classification labels to each subject (users and user processes) and object (files and data elements), and ensures that subjects only access objects for which they possess an appropriate label. For example, a user with a secret clearance would be granted access to secret, confidential, and unclassified information but not to top-secret information (Anderson 2008, 239–73).

To enforce this label-based access control, MLS systems use a reference monitor and the Bell-LaPadula model. A reference monitor is a small process operating in the security kernel of the operating system through which all information flow between users and data files must pass. When implemented properly, the reference monitor cannot be modified or bypassed and is simple enough to be rigorously evaluated and proven secure. As a central part of the security architecture, it enforces the security policy and allows components to communicate with each other only when the messages conform to that policy (Harris 2008, 327–8).

In the case of the Bell-LaPadula model of access control, the reference monitor prohibits reading information above the object's clearance and also prevents writing information below the object's clearance. Both rules protect confidentiality, the first by prohibiting users from accessing information beyond their clearance and the second by preventing users from "spilling" information of one level onto a file with a lower classification level. Depending on the application, developers may use other models where integrity takes priority over confidentiality. For example, the Biba model defines

levels of integrity such that information with a high integrity label can be trusted more than information with a lower label. The mechanisms of the Biba model work in the opposite manner as the Bell-LaPadula, so users cannot write up to higher integrity levels or read down to lower integrity levels (Harris 2008, 327–38).

To access a MLS computer system, users log on and select a security level. For example, a user cleared for Secret information can select a Secret, Confidential, or Unclassified level. Selection of the correct security level depends upon the task at hand and can be changed if the user wants to start working on a different task. It is essential to select the appropriate level because choosing a level too low for the task will prevent the user from being able to read the desired files; for example, a user logged on at an Unclassified level will be unable to access files labeled Secret. Choosing a level too high for the task is also a problem because the classification label of any modified files will be changed to match the active security level. For example, if a user logged on at a Secret level makes a change to an Unclassified file, that file's security label will change to Secret and it will no longer be accessible at the Unclassified level.

This need to operate at the correct security level for the task at hand can force users to change their security level frequently during the course of their work, depending upon how often these users need to access files of different classifications. Desk-bound knowledge workers such as intelligence analysts or mission planners may find this security level switching an inconvenience, but a warfighter operating a network-centric weapon system would find it impossible to change logon contexts without causing a major operational disruption.

Furthermore, MLS systems assume that each system will be accessed by only one person and that the user's personal identification and authentication credentials correspond to that one person. Again, this model works well for the knowledge workers who have their own computer terminals within their own individual workspaces, but warfighters typically do not enjoy these conditions. Operators of network-centric weapon systems such as ship or tank crews often operate their terminals as a team. Even if a terminal is primarily operated by one watchstander, its contents may be displayed to

several other people who are not identified and authenticated by the system. Physical security controls can compensate for this problem by ensuring that everyone physically capable of observing or operating the terminal has a clearance and access equal to the primary operator who is logged in, but this workaround defeats the underlying assumption that all users are identified and authenticated before being granted access.

A more appropriate architecture for connecting network nodes with different security levels can be built by using Cross-Domain Solutions (CDS). A CDS describes the use of high-assurance guards and filters to allow the interconnection of network segments and elements that operate at different security levels without allowing information transfer into unauthorized areas. These guards and filters examine the data trying to flow from a higher-security domain to a lower-security one and prevent any high-security data elements from passing through. Configuration of a CDS can be complex depending upon the type of information exchanged, and any mis-configuration or failure to anticipate data types or formats could lead to the compromise of sensitive information.

A CDS architecture provides a way for sensors, command centers, and weapon systems to share information without allowing complete visibility into all the information contained within each node. For example, a remote sensor that provides data at the Top Secret SCI level can feed its data to a command center approved to handle that level of information. The command center can then strip away data elements that identify the information's source and method of collection, thus reducing its classification level to Secret, and send the contact report to a contact database for sharing with warfighters. The guard connecting the command center and the contact database uses a customized algorithm to examine the command center's transmission for any information above the Secret level and blocks any such data.

Chairman of the Joint Chiefs of Staff Instruction 6211.02C governs the process for developing and getting approval to operate a CDS. This process ties into the overall certification and accreditation process that will be described in Chapter VII. A CDS does not provide the same rigorous security as a MLS system, but it can allow greater

operational flexibility with an acceptable level of risk, especially for systems that communicate with fixed message formats. The use of fixed message formats makes it difficult for buggy or malicious processes to send sensitive information into lower-security domains.

Even a well-designed CDS or MLS system can still fall victim to an attack on confidentiality through the use of covert channels and traffic analysis. A covert channel is any unusual or unexpected method of bypassing the security mechanism and sending highly sensitive information to a lower-security domain. For example, if some network resource is visible to both the high and low sides of a network, the high side could send information to the low side by changing some characteristic of the shared resource. This example would require both sides of the network to have some covert program installed that can translate the changes in the network resource into a digital data stream. Since covert channels can use a vast array of methods, implementing a system that provably does not leak information is extremely challenging (Anderson 2008, 263–5).

The concept of distributed information sharing resembles the current discussion in industry about "Cloud Computing" and Service Oriented Architectures. Cloud Computing refers to the infrastructure that allows information and services to be hosted in a central location and provided to users upon demand. A related concept, Service Oriented Architecture refers to the standardization and centralization of computing services so they can be used by multiple units of a larger organization (Hurwitz et al. 2007). Information stored within the network, rather than locked up inside one isolated unit, can improve the situational awareness of the friendly forces but also be at greater risk of compromise. Developers must decide carefully the maximum sensitivity of information allowed to be stored within the larger network. This decision requires a risk analysis of the methods of compromise and the tactical advantages of having the information in a distributed form (Zittrain 2009).

In addition to information, security classifications can also be applied to hardware. Assigning a classification level places requirements for storage and handling that seek to prevent theft or sabotage. In practice, however, it is difficult to classify

hardware because the security controls may place a significant burden on operators and maintainers. Developers must seek a balance between the need for efficient handling and the need to prevent theft and loss (Boak 1973, 22).

IV. PHYSICAL SECURITY

Physical security for a network-centric weapon system goes beyond the "gates, guards, and guns" that form the core of physical security in a generic military context. Since the components of this system—the tanks, planes, ships, and troops—will be involved in combat and combat-support operations, they have different physical security needs than buildings and bases. Enemy forces can still threaten the system, however, in three ways that involve physical contact with the hardware. First, they can salvage functioning network hardware from captured friendly units and use that hardware to monitor communications and transmit false information. Second, they can corrupt the manufacturing process of the system hardware and implant their own hardware devices that perform some malicious function. Third, they can simply destroy the network nodes and remove the ability of system units to communicate with each other.

A. EAVESDROPPING AND SPOOFING USING CAPTURED HARDWARE

Most computer security scenarios involve an attacker intercepting network communications or, in the worst case, remotely accessing a network host and planting a malicious program. With network-centric weapon systems, however, comes the additional problem of physical attacks on the computer hardware itself. Since the networking hardware will be installed on ships, tanks, planes, and even infantry units, it must be assumed that the enemy will at some point gain physical possession of that hardware. Once this happens, the hardware can be used to eavesdrop on friendly communications or insert false information. If the system is still functioning, the enemy may have access to the entire situational awareness picture, knowing everything the friendly forces know and, most importantly, which enemy forces are still undetected (Commission on Physical Sciences, Mathematics, and Applications 2000, 180).

Operators may have the opportunity to perform an emergency destruction when capture becomes likely. Such procedures usually focus on cryptologic keys hardware but can extend to all sensitive equipment. These procedures, however, assume that operators will have the time and ability to conduct a thorough destruction and that the enemy will

be unable to reconstruct the destroyed equipment. Designing for emergency destruction can be a challenge since operators need robust equipment that survives the rigors of combat. This very quality of robust design makes hardware difficult to destroy intentionally and more likely to fall into enemy hands (Boak 1973, 47).

An attacker who gains possession of a functioning network hardware unit can try to access any information stored within that unit's memory. This information can include a recent situational awareness picture—a summary of the positions and status of all friendly units and the estimated position and status of any detected enemy units. The attacker might also try to recover any cryptographic keys stored in the unit and used for authentication and message encryption. All this information would be stored in volatile memory (e.g., memory chips) and non-volatile memory (e.g., flash memory or disk drives), and since this hardware will likely be built to commercial standards it may not be too difficult to copy and read this information. Law enforcement units commonly do this as a forensic analysis of computers seized from criminal suspects (Harris 2008, 876–9).

System developers can use tamper resistance to make it difficult for attackers to extract information from captured hardware. For example, the IBM 4758 processor

> responds to tamper attempts in hardware by quickly zeroizing its secrets and executing a coprocessor state change, without requiring software intervention. Temper conditions include penetration attempts, temperature extremes, voltage variation, and radiation. (Dyer et al. 2001 60)

However, perfect tamper resistance is impossible to achieve, and designers cannot objectively determine how tamper-proof their hardware is (Schneier 2000, 215). The tamper-resistance challenge is complicated by attacks that use non-invasive methods that would not trigger a reaction from the target hardware. For example, Paul Kocher's differential power analysis technique, created in 1998 to extract the private key from smart cards, measures the power used during several hundred known transactions with the card to deduce the key value (Anderson 2008, 533).

Tamper-resistant designs must compete against an ever-improving array of forensic tools and techniques that can extract data from seemingly secure products. Recent demonstrations have shown that data in volatile memory chips can be extracted

by cooling them down before removing them from the target computer (Jackson 2008). Hard drives or flash memory chips can be copied and subjected to brute-force attacks on any encrypted contents. Attackers can re-construct deleted data from any magnetic medium by looking for the minute traces of residual magnetism left behind after files are deleted or overwritten (Anderson 2008, 490). Overall, it is exceptionally difficult to protect information when the medium that holds it is in enemy hands.

The use of asymmetric encryption can reduce the risk that the enemy will recover a cryptographic key from a captured hardware device and use that key to intercept friendly communications. By recovering a unit's secret key, the attacker can only decrypt messages sent to that particular device. Likewise, with the effort to use the captured unit to inject false information into the friendly network, recovering a secret key would allow the attacker to impersonate only that specific device. If the friendly network uses a robust PKI and the loss of the friendly unit is noted, network administrators can place that unit's PKI certificate onto a certificate revocation list and prevent friendly units from trusting any further messages from that unit.

In addition to using captured hardware to steal secrets and send bogus information, attackers can also conduct reverse-engineering activities to learn the details of the hardware designs. Once the designs are well understood, an industrially sophisticated enemy can then manufacture their own versions of the units and use these in the field to disrupt the friendly network or to impersonate friendly units and inject false information. Careful use of cryptographic authentication, such as that provided by a PKI, would limit the effectiveness of these attacks, and make imposter units capable of little more than low-level jamming and noisemaking.

The reverse-engineering threat also extends to the software and firmware resident on any captured hardware. Even though any code would exist only in the form of binary executables (instead of source code), adversaries could use reverse code engineering tools like disassemblers and debuggers to reconstruct the source code and learn the details of the algorithms in use. They could use this knowledge to discover weaknesses in the processing algorithms and develop new code-based attacks (Peikari and Chuvakin 2004, 9).

B. MALICIOUS HARDWARE

The hardware supporting network-centric operations may fall into enemy hands during its construction as well. The use of commercial hardware designs and fabrication means that the physical production of processor chips will take place outside of direct military observation and control. This is especially true for hardware produced overseas, where factory managers may face pressure from intelligence services that may be hostile to the United States (King et al. 2008).

Just as malicious code can be introduced into software or firmware, it can also be embedded into the design of microprocessor hardware. A hostile production facility can create back-door access points, logic bombs designed to cause failures when certain conditions are met, or covert transmission functions that send data to hostile collection points or introduce flaws in the cryptographic functions. Detecting these functions can prove extremely difficult as they consist of minor variations in the arrangement of the millions of transistors and connections found on modern processors (King et al. 2008).

To mitigate the risk of malicious hardware, program managers must provide oversight and security for their entire supply chain. They should inspect the final design of custom-made hardware and ensure that the fabrication process faithfully reproduces that design. Many hardware designs will use existing commercial processors, meaning that the chips were designed before they were selected for military use and therefore it is unlikely that an adversary would have implanted a military-specific malicious process. In this case, the program manager must ensure that the production process does not get altered and that the units supplied to the military project are identical to those provided to the commercial market. Finally, programs may decide to avoid certain vendors when the risk is too high, such as when the DoD canceled its decision to buy SIPRNet computers from the Chinese-owned company Lenovo in 2006 (U.S. Congress 2008, 73).

The risks from malicious hardware may not exceed the risks created from the lack of security effort put into commercial software, but both parts of the supply chain require careful attention. Dr. James Mulvenon, testifying before Congress in 2008, said,

> I would obviously agree that the supply chain is a big problem, particularly given the increasing percentage of these products that are being manufactured in China, the pressure that's being put on some of these companies to include Chinese standards, which involves giving up source code for Chinese-designated companies to then be able to build the APIs to make them compatible with those Chinese standards. But we should also look closer to home as well as in the sense that, as a Mac user since '87, I can tell you that Microsoft and its buggy code probably represents a far graver information warfare threat to the United States than a lot of backdoor Chinese equipment. But as long as we have a low bid acquisition strategy in that area, we're going to go down that road, and it requires much more attention to code auditing and hardware auditing than we do right now. (U.S. Congress 2008, 74)

C. HARDWARE DESTRUCTION

Much of the discussion about computer network security centers around the complex and obscure ways their programming can be altered or configurations changed. In a military situation, however, the quickest way to attack the availability of a network is to simply destroy some of the hardware. As Bruce Schneier pointed out, "One of the characteristics of denial-of-service attacks is that low-tech is often better than high tech: Blowing up a computer center works much better than exploiting a Windows 2000 vulnerability" (Schneier 2000, 39). A dispersed network that includes warfighters, commanders, and sensors would require a great deal of hardware to propagate the signals throughout the network, and this hardware would include relay stations on satellites, planes, ships, and ground vehicles and facilities. These relay stations would serve to collect messages from nearby units and re-transmit them to other units and relay stations, eventually delivering the message to the entire network. Destroying some of these stations could effectively remove some of the nodes off the network and therefore make them unable to contribute to the tactical operations.

Destroying these network relay stations could prove especially easy, especially when compared to the effort needed to destroy actual warfighting units. In order to reach a large number of network nodes, the relay stations would have to be positioned in an easily observed location. They would also need to transmit more frequently and with higher power then the warfighting units or sensors, further raising their visibility. Unfortunately, a location visible to friendly forces would likely also be visible to enemy

ones. The enemy may make his first strike focus on networking hardware, causing a long-term denial-of-service on the friendly network. Robbed of their ability to communicate, friendly forces would have to revert to less-effective legacy radio links or just give up on communications and perform independent operations. These independent operations are exactly the opposite of the integrated, network-centric operation for which the units were trained, so friendly forces would suffer a dramatic loss of effectiveness.

The problem of vulnerability to physical attack is especially acute for satellites. The U.S. military has designed much of its communications infrastructure to rely on satellites, and a future network-centric architecture would likely follow this philosophy. Historically, orbital space has been a bastion for the U.S., but recent advances in anti-satellite weapons have shown that hostile nations can knock them out. China, in particular, has demonstrated the ability to destroy or disable satellites, and recent writings of the Chinese army suggest that satellites would be the first targets in a conflict with the U.S. (U.S. Congress 2008, 10).

Given the difficulty of hardening or hiding network relay stations on the battlefield, the only remaining method of assuring availability may be the deployment of a large number of smaller and highly mobile stations. Unmanned Arial Vehicles (UAVs) could serve this function well if they deployed in large numbers and formed a mobile ad-hoc network to pass information. Given a robust networking protocol that can quickly adapt to the loss of network members but not overwhelm the system with repetitive or circular traffic, friendly forces can replenish the UAVs as they are disabled and maintain network connectivity. A large number of small satellites could also perform this function, replacing the small number of large and vulnerable satellites currently in use.

V. OPERATIONAL SECURITY

The operational philosophy behind network-centric warfare assumes that warfighters, sensors, and commanders will have continuous data links available to share their contact information, status, and orders. Unfortunately, radio transmissions can be easily detected by enemy forces, revealing the position and movement of friendly units and even providing some clues to the status of those units. Since military tactics often rely on stealth and surprise, the emanations needed to participate in the network can leave friendly forces at a severe disadvantage. Developers of network-centric systems, therefore, must consider the operational security needs of friendly forces to operate without detection from the enemy.

The need to operate without detection greatly affects covert units such as submarines, stealth aircraft, and Special Forces units. These units, termed disadvantaged users, typically operate without making any detectable emanations, allowing them to receive information from the network but not to transmit anything (Goshorn 2008, 3). They rely upon their covert status for mission effectiveness and even their survival since they lack the ability to withstand or repel a significant attack.

The inability to transmit to the network presents both tactical and technical problems. Tactically, the network-centric military of the future will adopt a doctrine that assumes connectivity among all warfighters, sensors, and commanders. Units will train to use the common situational awareness picture created by all network nodes, and their tactics will be based on this information sharing. The presence of friendly units operating off the network could complicate this tactical assumption and increase the risk of friendly fire.

The technical problem comes from the possible use of a connection-oriented protocol for network communication. Connection-oriented protocols establish a communications session between the sender and receiver as the first step of any information exchange, and they can authenticate each other and create a cryptographic

key just for that session. Most importantly, they require both parties to acknowledge any data packets sent by the other party. This acknowledgement ensures that all the data arrives intact (Comer 1999, 331).

Any network member restricted from transmitting but still wishing to receive network information would be unable to use a connection-oriented protocol since they could not establish a session or acknowledge packets. They would typically have to use a connectionless protocol to carry one-way data transfers. Connectionless protocols are commonly used on the Internet to carry streaming audio and video feeds, and they provide no protection against data loss. Corrupted packets may be recognized as such by the receiver and disregarded, but since the receiver has no way to ask for the packet to be re-sent, that data is lost (Harris 2008, 498–9).

Reliance upon a connectionless protocol and its risk of data loss might be tactically acceptable for covert units if the network pushes frequent updates to situational awareness data and tactical orders. In this situation, the information lost due to dropped packets can be revived in the next update, and the receiver can maintain an adequate tactical picture. This solution, however, would increase the burden on tactical networks and place them at greater risk of overload.

Network administrators would also face the difficult task of deciding where to provide broadcast coverage intended for these covert units. Transmitting throughout the entire network would consume the available bandwidth and restrict the ability of the rest of the force to communicate. Transmitting only in the geographic area where the covert units operate risks alerting enemy forces to the covert units' presence, and it might not even be possible to arrange this type of coverage since the most covert operations are generally kept secret from the mainline forces. Administrators must carefully balance the need for network capacity with the need to protect the secrecy of covert units.

Covert units may be able to use Low Probability of Intercept (LPI) communications methods to achieve some degree of network participation. LPI communications are wireless data transmissions that are difficult for anyone but the intended receiver to detect and are also difficult to jam (Boak 1973, 11). Typical LPI

technologies make use of high-frequency radio transmissions using a high-gain, directional antenna (Carmana 2009, 2-AM-82). Optical signals can be highly secure, though they can easily be blocked by clouds or smoke.

Another operational security concern comes from an adversary's ability to deduce important tactical information from reading patterns in friendly communications. The adversary may not be able to decode and eavesdrop on these transmissions, but through analyzing the amount and location of transmissions they may be able to determine that some significant action is taking place. This aspect of operational security is especially important before hostile action has actually started, since a spike in communications activity can tip off the enemy that a surprise attack is coming. Friendly units can avoid giving away this type of information by maintaining a constant level of communications activity even when the messages contain no tactical information, filling their messages instead with "placeholder" data that is ignored by the recipient (Commission on Physical Sciences, Mathematics, and Applications 2000, 194).

THIS PAGE INTENTIONALLY LEFT BLANK

VI. PERSONNEL SECURITY

Human developers, users, and maintainers form a critical part of any military system, and a proper systems engineering effort must account for their contributions and limitations. People can threaten the security of a system intentionally during the developmental or operational phase by conducting espionage or sabotage. Even loyal and well-intentioned people can cause security problems by failing to follow security-related procedures or by falling victim to social engineering attacks. Developers must consider every aspect of human interaction and protect against the security risks that people bring to the system.

A. THE HUMAN COMPONENT

Any network-centric military system will include many people to design, build, maintain, and operate the hardware and software. The fields of systems engineering and human factors engineering have advanced the concept that these people are integral parts of the system and that any design decisions must account for their actions and mistakes. As the DoD Office of Force Transformation stated,

> The implementation of NCW is first of all about human behavior as opposed to information technology. Our focus should be on human behavior in the networked environment. How do military forces behave, perform, and organize themselves when they are networked? (Office of Force Transformation 2005, 3)

Accordingly, many security experts believe that people are the greatest security weakness in any computerized system. As Bruce Schneier stated, "People often represent the weakest link in the security chain and are chronically responsible for the failure of security systems" (Schneier 2000, 255).

The system life cycle includes human activities at every stage. At system inception, people execute the three elements of the military's procurement and sustainment system: the Planning, Programming, Budgeting, and Execution process, the requirements development process, and the acquisition management process (DoD 2003). Once this system identifies needs, allocates funding, and assigns a program manager,

other people work to design and build the system. The fielded system then requires other people to operate and maintain the hardware and software elements throughout the operational phase of the system life cycle. Even at the end of system life, people are needed to oversee the decommissioning and disposal of hardware and software. Every stage of the system life cycle requires people who must be trusted to perform their role properly, and the security risk to the system comes from people who intentionally or accidentally violate that trust.

In any other aspect of security, technological methods can solve security vulnerabilities and reduce risks. Rigorous design practices and thorough testing combined with a process of detecting new risks and quickly issuing patches or updates can eliminate most risks to computer and network security. Meticulous labeling of information and design of information flows can eliminate most risks to information security. Tamper-proofing, authentication techniques, and hardware redundancy can boost physical security. LPI communication methods and connectionless protocols can provide operational security. However, because human interaction involves judgment, no amount of technology-based techniques and tools can, by themselves, bring the personnel security risks to an acceptable level (Harris 2008, 135). These tools can reduce the personnel-related security risks, but designers must consider human fallibility and incorporate procedures that anticipate the most likely human failures and minimize their likelihood and impact. As Bruce Schneier stated, “Mathematics is logical; people are erratic, capricious, and barely comprehensible” (Schneier 2000, xii).

Given the need for trusted human interactions in any complex system, designers must consider the ways that people can introduce security problems. People can cause security violations intentionally or accidentally. The intentional violations, espionage and sabotage, are the easiest to understand. Accidental violations, however, can come from a wide variety of behaviors and situations and often involve a trade-off between security and functionality. Designers must weigh the need for security with the efficiency, reliability, and ease of use of systems and provide both technological features and operational procedures to achieve an acceptable level of risk.

B. INTENTIONAL SECURITY VIOLATIONS

The need to include fallible and unpredictable humans as critical components of complex systems, introduces the risk that some of those people may be working for the enemy as spies or saboteurs. These people, whether they sympathize with the enemy or simply want fast cash and an exciting line of work, can give sensitive information to the enemy, inject false information into the network, and cause equipment failures and communication outages. The immense complexity of a network-centric system can help these malicious insiders disguise their actions and avoid detection. This malicious action can take place during system development and during the operation and support phase of the system life cycle.

1. Developmental Stage Security

During the developmental stage, enemy agents can collect system design and configuration information and leak it to hostile forces. The improvements in digital information storage technology have made espionage easier and more efficient than it was in the days of the cold war, and vast quantities of information can be copied onto flash-memory devices and smuggled out of the building. These devices, including Universal Serial Bus (USB) thumb drives, memory cards, portable hard drives, cell phones, and digital media players, will successfully interface with the computer hardware used in many development environments since the devices and hardware both conform to the same commercial open-architecture interface standards. Armed with detailed design and configuration information, potential adversaries can search for system weaknesses and develop customized attacks for use if a conflict emerges. States and trans-national organizations not directly involved in a conflict with the U.S. can provide these attack methods to those groups actually doing the fighting as “off-the-shelf” tools ready for immediate use (Commission on Physical Sciences, Mathematics, and Applications 2000, 177).

Thefts of technical data may result from industrial espionage in addition to the nationally sponsored spying conducted for military purposes. American technology companies face a significant threat of espionage from foreign companies, some of whom

are assisted by their governments (Epstein 2008). If advanced technology information is stolen by a foreign entity, it may be sold to other companies and governments and used to create competitive products and to design the same customized attacks made possible from government-sponsored espionage. Secretary of Defense Robert Gates, describing the threat from industrial sabotage, said,

> In a world that increasingly measures power in economic as well as military terms, many foreign intelligence services are turning their sights to stealing American technology and trade secrets. Some countries with whom we have had good relations may adopt a two-track approach, cooperating with us at the level of diplomacy while engaging in adversarial intelligence collection. (Nolan 2000, 1)

China presents a particular threat of industrial espionage. They have a long history of reverse engineering and commonly use it to match competitor's products and build new military hardware. While military-developed products would require some degree of espionage, the commercial products that often form the foundation of military systems can be acquired simply by purchasing them (Fritz 2008, 33).

Enemy agents, commercial spies, and disgruntled employees can also pose a sabotage threat. Major software projects can be so complex that a small but malicious change in a code library may go unnoticed. These changes can include back-door access points that can allow system access without proper authentication, logic bombs that degrade system performance or cause data errors when the system is used operationally, or covert channels that leak sensitive information without proper encryption or classification-checking controls. Both espionage and sabotage by insiders share features that cause the event to take place and that help it happen successfully (Band et al. 2006, vii).

Defending security during system development requires the standard tools used in government and commercial office settings. People must pass background checks and hold an appropriate security clearance before being granted access to sensitive material. Administrators of the development system should remove any access points for digital media not needed for development work, including USB ports, CD and DVD recorders, and card readers. Systems can also use software applications at the workstation and

network level to monitor information flow over the network and detect suspicious actions from users. Network operations should be thoroughly logged, where the system records each operation in a log file, so security personnel can recognize security violations and identify the personnel responsible. Code submitted to production libraries must be inspected and understood by more than its author. All these security measures will add to the cost of development, but managers must resist the temptation to raid the security budget to make up for development cost overruns.

The background checks and security clearances, while helpful for screening out people with known suspicious connections and activities, cannot predict anyone's future activities or accurately judge a person's true intentions and loyalties. Program managers must, therefore, resist the temptation to downplay the likelihood of insider attacks simply because all the insiders have survived some vetting process. Bruce Schneier, discussing the futility of using terrorist watch lists to prevent terrorist attacks, points out how identity cannot be correlated with intent:

> But even if these [terrorist watch] lists were complete and accurate, they wouldn't work. Timothy McVeigh, the Unabomber, the D.C. snipers, the London subway bombers and most of the 9/11 terrorists weren't on any list before they committed their terrorist acts. In the end, the photo ID requirement is based on the myth that we can somehow correlate identity with intent. We can't. And instead of wasting money trying, we would be far safer as a nation if we invested in intelligence, investigation, and emergency response — security measures that aren't based on a guess about a terrorist target or tactic. (Schneier 2008b)

2. Operational Stage Security

During system operations, administrators have the power to attack the confidentiality, integrity, and availability of the network and its data. The specific role and duties of these administrators depends upon the system design, but some degree of human intervention and supervision will be required for any type of system. Administrators could possibly link a covert enemy unit to the network, providing that unit with the locations, status, and orders of the entire friendly force as well as the complete description of sensor contacts and knowledge about the opposing force. They could mis-configure encryption settings to allow some part of the network to transmit its

information in the clear where it could be intercepted by hostile units. They could attack the integrity of the network by directly editing data, changing friendly units to appear hostile, deleting enemy units from sensor reports, or creating bogus hostile units to serve as a diversion. They could also damage the system's availability by removing friendly units from the network or shutting down the network itself.

The use of contractors to perform administrative and technical duties presents a special risk because military supervisors have less ability to screen and manage contractor personnel. Unable to build technical expertise in-house, some companies, and government and military organizations rely on contractors and run the risk of becoming dependent upon them. As an example of the risk that comes with contracted information technology (IT) administration, the World Bank recently experienced several major penetrations in their highly sensitive treasury unit. The FBI discovered that spyware had been installed on servers and suspected that contractor personnel, who performed nearly all the bank's IT services, were the culprits. The bank terminated the contract with the company but found itself unable to operate without the contractor personnel because bank personnel lacked enough knowledge about the system to be able to operate it themselves. As a result, many of the contractors continued to work, either as employees of the replacement company or as newly hired bank staff. The perpetrators were never discovered and the bank cannot be certain that they are not still working there in sensitive positions (Behar 2008).

To protect their security from personnel-related risks, operational network-centric systems can borrow some of the practices, such as two-person control and controlled access, that are used by high-security programs. The two-person control concept aims to make it impossible for one person to sabotage the system without their actions being discovered, reported, and corrected (Commission on Physical Sciences, Mathematics, and Applications 2000, 178). Administrative-level access to the system, at least in areas that can damage confidentiality, integrity, and availability, should require the direct attention of at least two trained administrators. This requirement must be more than a procedural requirement; the administrative software function should control access by requiring valid access credentials from two or more qualified personnel. Once the participants are

identified and authenticated, the second person must observe the actions of the person actually operating the system and have the ability to stop the changes before they are promulgated to the system. Defeating this security measure would require collusion between the two administrators, a highly unlikely situation that would be considered an acceptable risk (Harris 2008, 136).

Logging of administrative actions is just as important to the operational system as it is in the developmental environment. Administrators must access the system with unique user identification and authentication, preferably using two-factor authentication including a physical token such as a smart card. Given this strict authentication, every administrative action can be recorded along with the identity of the person who took the action. Careful log reviews may then detect suspicious actions or outright security violations and identify the responsible party. To ensure that the logging system remains effective, it must be designed such that the log file cannot be altered and the logging function cannot be switched off or bypassed (Harris 2008, 245–6).

C. ACCIDENTAL SECURITY VIOLATIONS

Security violations caused by loyal and well-intentioned operators can cause damage similar to that done by spies and saboteurs. Some people may be tricked by social engineering attacks into revealing secret information or granting system access to enemy agents. Others may ignore security rules in order to boost productivity. Unlike the case with intentional violations, these accidental events reveal systemic weaknesses that often require broad countermeasures to combat and may never be fully preventable.

1. Social Engineering

Some of the most notorious computer hackers, like Kevin Mitnick, found social engineering to be a more efficient method of penetrating computer system security than technological attacks (Schneier 2000, 267). He, and many others, used fraud and deception to trick trusted insiders into revealing secret information or allowing him to access restricted parts of the system. This type of attack can be used against any type of organization or system that includes people and is not restricted to computer-based or network-centric systems; however, they can prove especially effective against computer-

based systems because of the inherent value of information like passwords and configuration settings and the damage that an attacker can cause when sensitive information is leaked.

Social engineering uses the ancient arts of fraud and deception to prey upon the fundamental characteristics of human nature. Most people want to be appreciated and will be tempted to bend the rules to help someone they believe deserves assistance and will show them appreciation (Schneier 2000, 268). People also tend to view authority figures with a mixture of fear and respect and may yield to the demands of someone impersonating an authority figure. Others simply wish to avoid conflict and may provide information or access to someone demanding it in order to get the person to leave. Some people may fall victim to their own greed and break security rules in return for some tangible benefit, though this act would generally propel the "victim" into the category of a spy or saboteur and not a well-intentioned dupe.

Compounding the vulnerability to social engineering attacks is the fact that people are often very good at self-deception. They sometimes substitute wishful thinking and naïve assumptions for a sober analysis of the situation and the risks of breaking the rules (Paul, Niewoehner, and Elder 2007, 49). People may fail to consider the risks of divulging sensitive information or to rigorously authenticate the people requesting this information because of the subconscious assumption that everything will end well because it has in the past.

Defense against the social engineering attack relies on security training, the creation of a security-centered culture, and the enforcement of the principle of least privilege. The training exposes personnel to the concept of the social engineering attack and to the techniques that attackers use to get information and access. It establishes authentication requirements that personnel must enforce on anyone seeking information or access and it shows the consequences of failing to properly authenticate someone. Training provides little benefit by itself, but it creates the intellectual foundation for the security culture needed to ensure that the secure procedures and practices are actually followed.

A robust culture of security within an organization ensures that the information learned in training is consistently and correctly put to use. Supervisors must demonstrate through word and action the importance of security. They must conspicuously enforce security procedures and make security practices an important part of personnel evaluations, including awards and disciplinary actions. They must also accept that security practices necessarily reduce productivity and ensure that personnel are never pressured to take shortcuts in order to meet an urgent deadline or production goal. Finally, senior personnel must set the example and scrupulously follow the security rules themselves. This last point may be the most difficult from a cultural perspective, since senior personnel typically enjoy exemptions from policies that apply to rank-and-file personnel (e.g., reserved parking space, administrative support); however, it is vital that security practices and rules are perceived as being a legitimate precaution that applies across the board, and not simply an indication of rank and status.

System developers and administrators can also use technological means to help reduce the risk of social engineering attacks (Schneier 2000, 268). Although hardware and software tools cannot prevent personnel from divulging information or granting access to an attacker, they can reduce the damage inflicted by such an error. System administrators must vigorously apply the principle of least privilege and only allow personnel to access information and systems they need for their duties. Compartmentalizing information in this manner ensures that few personnel have the ability to give away "the keys to the kingdom" and that compromises of data and access damage only the area under the victim's immediate control. Other technological tools that can help organizations resist social engineering attacks are those that allow personnel to reliably authenticate anyone asking for information or access. For example, use of digital signatures on emails reduce the risk that an attacker can impersonate someone inside the organization and extract information from an unwitting co-worker.

2. Security Procedures

Any system with human operators will likely require some kind of operator action to maintain the security of the system. Such actions may include activating an encrypted

channel or using a low-probability-of-intercept method of communications. The system and component design will determine what operator actions are available and necessary for security. This design may force operators to operate in a secure mode by disallowing any other option, or it may provide greater operational flexibility but rely on operators to make the most prudent choice.

A well-designed network-centric weapon system would naturally rely far less upon operator actions to maintain security than a typical computer network used for office automation or electronic commerce. Commercial systems, in particular, must remain open to a wide variety of un-authenticated users with hardware and software operating according to open architecture standards. Even if the sensitive parts of the system use authentication and other security measures, the open and less-protected interface to the general public allows attackers an avenue to gain access and work their way into the protected areas. In contrast, a network-centric military system has no need to allow access to anyone but an authorized user who can prove his identity and will use specific hardware, software, and protocols to gain access. Therefore, the military system can often use technological countermeasures to provide security and prohibit operator actions that introduce security risks.

The system design may provide some method of bypassing or overriding security procedures for operation during an emergency when the normal procedures are infeasible. For example, a radio may allow operators to establish an unencrypted connection if the authentication/encryption process fails and the operator has an urgent need to communicate. Such flexibility could improve the reliability of the networking equipment and provide a backup method of operation, preventing a failure of a security-related function from automatically becoming a failure of the entire network. The degree of freedom given to operators to perform this security override will depend upon the system design, but any design will rely upon the operator's security-related actions to some degree.

While providing the ability to operate in a less-secure configuration may improve reliability, it also counts on the operator's judgment to only operate in this manner when necessary. Unfortunately, operators may lack a sufficient understanding of and

appreciation for security and may start bypassing security-related procedures to improve productivity and ease of use. Pressured to "Just get it done," some operators may view security rules as unnecessary obstacles and avoid them, opening a vulnerability that can be exploited by a hostile force. The high turnover in the military and the life-and-death tactical concerns mean that security knowledge may be a low priority for operators.

Protecting system security from this abuse of operator discretion requires actions similar to those needed to protect against the social engineering attack. Operators must be trained on the importance of secure operation and the risks of taking shortcuts. Supervisors must set clear standards on secure operating practices and consistently enforce them while setting the example through their own behavior. The system should log all non-standard operations to help supervisors enforce the standards and understand the real-world use of the system.

Developers must also understand what kinds of personnel-related attacks on the system work and why. Users sometimes disregard security indicators and focus on content, particularly when warnings are over-used due to system mis-configurations or software errors. Therefore, the system must not rely upon users to seek out security indicators and scrupulously obey security warnings. Even security-conscious users can be fooled by visual deception attacks that create false security indicators to mask malicious activity (Dhamija, Tygar, and Hearst 2006, 3).

THIS PAGE INTENTIONALLY LEFT BLANK

VII. WHOLE SYSTEM SECURITY

As the previous chapters have explained, attacks on the security of network-centric military systems can take a wide variety of forms. Focusing security efforts only on one aspect—computer and network security, for example—would leave the system vulnerable to unexpected attacks from other avenues. According to Arthur Cebrowski,

> Implementation of NCW must look beyond the acquisition of the technical enablers to individual and organizational behavior, e.g., organizational structure, processes, tactics, and the way choices are made. In other words, all elements of the enterprise are in play. (Office of Force Transformation 2000, i)

System developers must embrace a systems engineering approach to security and consider all the threats and vulnerabilities facing the system in the larger context of its operational environment. This examination must include not only the technical characteristics of the components but also the people who operate and maintain the system, the support elements that keep the system running, and the environment in which the system will operate. As Bruce Schneier noted when discussing cryptography,

> The weak points had nothing to with mathematics. They were in the hardware, the software, the networks, and the people. Beautiful pieces of mathematics were made irrelevant through bad programming, a lousy operating system, or someone's bad password choice. (Schneier 2000, xii)

Once they identify the system-level risks and create security requirements to address those risks, the engineers can allocate those requirements into detailed features and characteristics that meet the system's needs.

The systems-based approach applies even within security domains such as computer and network security. Adversaries may avoid the highly protected tactical network and focus their efforts on less-protected areas such as logistics systems. These lower-profile systems can provide both an entry point into tactical systems and a source of information that can infer the movements, status, and intentions of friendly forces. For example, seemingly innocent information on logistics preparations and reconnaissance

operations can signal the timing of an attack (Boak 1973, 4). Furthermore, it may be possible to defeat the entire system by crippling its supporting infrastructure. According to John Tkacik,

> The Chinese military doctrine stresses the importance of penetrating an adversary's military logistics and personnel networks. Furthermore, the multiple intrusions into what nuisance and criminal hackers would regard as boring, mundane networks — networks that do not offer the treasure trove of credit card numbers, bank accounts, and identity data that criminal hackers typically seek — suggest a military purpose. (Tkacik 2008, 3)

Given that the number and types of risks are virtually unlimited and that the resources to mitigate those risks are highly constrained, the other task for systems engineers is to evaluate the relative efficiency of security-related attacks and apply the development effort to combat the most likely and effective attacks. This task requires a great deal of judgment since the risks tend to be subjective and speculative, but a detailed failure analysis study combined with close attention to real-world security violations in computer-intensive systems should help developers quantify the risks and assign a relative priority.

Developers must also indirectly boost the security of the system by making the design as simple as possible. According to Bruce Schneier,

> Complexity is the worst enemy of security. As systems get more complex, they necessarily get less secure. (Schneier 2000, 1)

Simpler systems enjoy greater security because of the reduced number of opportunities to attack the system through unexpected inputs or configurations. They present an easier management task for administrators and operators, reducing the likelihood of inadvertent security violations. They also make easier subjects for technical evaluation, allowing a more thorough analysis and a more confident declaration of the security of the design.

The security problems brought about by complexity may be especially difficult for a network-centric weapon system. Such a system will likely be formed from a wide variety of components, each optimized for the special requirements of the forces and commanders using them. Current methods may provide an assurance that each

component can operate securely, but this assurance does not extend to each combination of components operating together in a tactical network. The problem compounds even more when systems of allied nations join the network. Managing this complexity requires the creation of a security architecture that specifies the components of the system and their interactions. This architecture simplifies the task of security analysis by allowing developers to operate at a high level of abstraction with regard to system elements.

A. TEST AND EVALUATION

Once the developers complete a preliminary design of the system, they can use penetration testing to discover overlooked vulnerabilities. This kind of testing uses a "red team" of creative people who have significant training and experience in attacking networked systems to attack the system and record the outcome of their efforts. To be effective, this testing should impose as few constraints as possible on the team. The team should be given general information about the system and its configuration, especially if the system uses commercial components, in an effort to best simulate an attack from a sophisticated foreign power. They should work not only to attack the system but also to find novel ways of circumventing the security-related countermeasures. Even the most rigorous penetration testing cannot detect all possible vulnerabilities, however, since the number of possible attacks is essentially infinite and since a real attacker may conceive of an attack method unimagined and untested by the red team (Harris 2008, 1090–8).

This penetration testing effort can be part of the larger developmental testing effort. Developmental testing, as described in DoD Instruction 5000.02, subjects the system to a set of planned, controlled, and complete scenarios to verify that the security functions operate properly. Since this testing takes place in a controlled environment and is conducted by developmental personnel, it can methodically exercise a wide range of inputs and configurations to discover vulnerabilities not adequately mitigated by the security functions. This testing starts at the component level and works its way outward

to larger portions of the system, eventually including scenarios that capture the unique needs of information, physical, operational, and personnel security as well as interactions among them.

Developers must also conduct operational testing to measure the demand placed upon users by the security functions and requirements for computer and network, information, physical, operational, and personnel security. This testing uses real operators and operationally representative environments to simulate the function of the system in a combat situation. Part of this testing should measure the time, effort, and skill needed to operate securely, comparing those metrics with the system configured to operate without the security features and calculating the difference. Testing, including comments from operators, will reveal those specific functions or procedures that impose a significant operational penalty. Developers can then evaluate the relative cost and benefit of each security function and procedure and either improve or remove the ones that impose a cost far greater than their value to security. Operational testing should also include a red team effort to compromise the system in a less predictable and controlled environment, providing one more opportunity to discover novel and unexpected vulnerabilities and attack techniques.

There is an important distinction between functional testing and security testing (as introduced in Chapter II), though both are required to provide a reasonable assurance that the system will function properly and securely. Functional testing steps through all the usage scenarios, simulating the users, interfacing systems, and inputs for each scenario. It verifies that the system will correctly process inputs from users and other systems and react as intended without crashing or providing erroneous data. It also ensures that the user interface is adequate to allow users to operate the system without confusion, delay, or error. The Joint Interoperability Test Command, part of the Defense Information Systems Agency, plays a major role in functional testing of networked systems.

In contrast, the security testing considers the ways that an attacker might attempt to compromise the system. It considers the wide variety of unusual or unexpected inputs that might place the system in an insecure state. Since there exists a nearly infinite

combination of unexpected inputs, configurations, and conditions, security testing cannot guarantee secure operation the way that functional testing can guarantee proper operation. This is why security testing must consider the most efficient attack methods, whether from the fields of computer, network, information, physical, operational, or personnel security, and apply testing effort toward vulnerabilities judged most likely to be exploited in an attack.

B. CERTIFICATION AND ACCREDITATION

While testing provides a degree of technical assurance that the system will function as designed and will be both effective and suitable for combat use, the certification and accreditation process provides a parallel effort to judge the security of the system and ensure that systems are only operated if their security risk is reduced to an acceptable level. The DoD currently uses the Defense Information Assurance Certification and Accreditation Program (DIACAP), as defined in DoD Instruction 8510.01, to perform this function and to establish configuration management processes for security functions. In this context, certification refers to technical evaluation of an information system that determines how well the system complies with its required Information Assurance (IA) controls. Accreditation refers to the granting of permission to operate and acceptance of risk by a designated accrediting authority (DAA).

The DIACAP applies to DoD information systems and is most applicable to office-automation systems using standard commercial hardware, software, and protocols. According to DoD Directive 8500.02E, DIACAP applicability extends to platform IT interconnections. Platform IT refers to the hardware and software in special-purpose systems such as weapons, simulators, test and maintenance equipment, and research and development equipment. The DIACAP does not apply to the platform IT itself, but only to its network connection. This boundary in the DIACAP between the weapon system and its network connection works well for legacy weapon systems where the network connections are limited in scope; however, network-centric weapon systems will feature a highly-integrated network connection and will be designed with network functionality as a top priority. The security features of a network-centric weapon system will span every

aspect of its operation, not just the network interface. Therefore, the DIACAP will require either significant additional guidance or a revision before it can be applied to network-centric systems.

DIACAP applicability, along with that of the entire DoD IA program described in DODD 8500.02E, does not extend to intelligence systems. The intelligence community has its own certification and accreditation program, described in ICD 503. Significant elements of a network-centric system, including unmanned sensors, satellites, and reconnaissance platforms, as well as the analysts and commanders who use the information provided by these elements, may fall under ICD 503 for their certification and accreditation. The fact that two separate programs will be responsible for different parts of the overall system strains the cohesion of the system and threatens to prolong the legacy philosophy of independent systems. If intelligence-gathering platforms are to be included in the network-centric system of the future, this split responsibility could greatly complicate the system's development.

The DIACAP assigns IA controls to information systems according to the level of confidentiality, integrity, and availability needed by the particular system. These controls include elements of computer and network security, information security, physical security, and personnel security. Operational security concerns are handled separately in publications about tactical doctrine and do not contribute any IA controls. DoD Instruction 8500.2, "Information Assurance Implementation," lists controls grouped into the following subject areas: continuity, enclave boundary defense, enclave computing environment, identification and authentication, personnel, physical and environmental, security design and configuration, and vulnerability and incident management.

Early in their development, information systems receive two designations that determine which IA controls will apply. The Confidentiality Level (CL) of classified, sensitive, or public describes the sensitivity level of the information processed by the system and the degree of protection that must be applied to prevent attackers from learning this information. The Mission Assurance Category (MAC) combines the integrity and availability needs of the system and describes the degree of protection needed to ensure that the system's information is correct and accessible. Systems

handling vital information for operational forces receive MAC I, systems handling important information for supporting missions receive MAC II, and systems handling day-to-day business information receive MAC III. Each CL and each MAC has a separate attachment to enclosure 4 of DODI 8500.2 containing the required security controls, so all systems will use two of the six attachments to get the complete list of required controls.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSION

This thesis has explored the security of network-centric weapon systems by describing the different sources of security risks and by explaining the importance of a systems engineering approach. Developers and policy makers must keep security concerns in mind as they bring new network-centric systems to the warfighters. Part of this consideration includes weighing the applicability of network-centric systems to current and future operations and building the communications tools to provide the foundation for the future network.

A. APPLICABILITY TO CURRENT AND FUTURE OPERATIONS

A military force using network-centric weapon systems would see the greatest advantage against another large, conventional military force. The benefits of a networked force allow for improvements in maneuver and concentration of fire against ground units, ships, and aircraft. It provides the ultimate expression of conventional, state-on-state, third-generation warfare.

The military conflicts currently facing the United States and its allies do not conform to the type of combat that benefits from network-centric weapon systems. Rather, the missions in Iraq and Afghanistan resemble the counterinsurgency and fourth-generation models of asymmetric warfare and political stabilization (Lind et al. 1989). These conflicts feature non-state actors, widely dispersed forces, and a blurred line between combatants and non-combatants. The social, cultural, political, and economic factors that dominate these situations derive little benefit from an increasingly effective conventional military force—a force that lacks a proper enemy to engage.

If the United States will primarily experience this unconventional type of warfare for the near future, an aggressive investment in network-centric weapon systems may be misplaced. On the other hand, a robust network could serve as a force multiplier and allow a smaller conventional force to provide continuing deterrence against a large, conventional attack while the remainder of the force concentrates on unconventional missions. The natural life-cycle progression of weapon systems provides opportunities to

add network-centric characteristics and capabilities in an incremental manner, providing opportunities for user feedback and for learning from early experimentation.

The nascent network-centric capabilities used during Operation Enduring Freedom and Operation Iraqi Freedom have provided an opportunity for experimentation in an operational environment as well as a glimpse of their potential. Commanders praised the ability to track friendly forces and the greater efficiency of air operations due to the situational awareness provided by data links (Office of Force Transformation 2005, 17). In these situations, however, the enemy lacked the ability to threaten the security of the network. Security, therefore, has not been tested. Conflict with a major power capable of sophisticated information warfare would probably include a major attack on all aspects of system security. Developers, therefore, must not conclude that the lack of security problems to date indicates the lack of threats or vulnerabilities that may manifest themselves in the future.

The move away from large-scale, conventional warfare has already manifested itself in recent changes to the FCS. First, the Army canceled the manned ground vehicle portion of FCS. Then, in May 2009, it changed the program to a successor program called Army Brigade Combat Team Modernization. This new effort will continue many of the concepts from FCS, but will deliver them over a longer time period as a gradual force modernization. As Ben Friedman, a research fellow at the CATO Institute commented,

> It will be better for the ground forces to have FCS broken up. Conventional insurgencies turn out to be situations where you want heavy vehicles. We don't need to get there that fast; you can get to theater on sealift. It is probably good that the Pentagon is adjusting to realities that turn out to be different. (Osborne 2009)

Despite these recent cuts, however, the Army still plans to create the tactical network envisioned for FCS and use it with the upgraded forces as they are deployed (Osborne 2009). The pace may have slowed, but the march toward a network-centric force continues as the services plan to gradually add more networking capabilities and network-centric tactics.

The future development and deployment of network-centric weapon systems depends largely on the enabling communications technology that gives disparate units the ability to share information. One example of this enabling technology is the Joint Tactical Radio System (JTRS). JTRS promises to provide this capability, featuring nine separate waveforms optimized for different users with their unique needs and constraints. JTRS radios feature a modular and open-architecture design with most functionality provided by software. This design allows services to buy compatible radios and use software configuration to optimize the functions to the particular mission needs (Carmana 2009).

B. SUMMARY OF SECURITY CONSIDERATIONS FOR NETWORK-CENTRIC WEAPON SYSTEMS

Network-centric weapon systems offer the prospect of greater efficiency and effectiveness of a military force. The combination of situational awareness, self-synchronization, and robustness of command promises to help a smaller force operate with the power of a much larger force without the investment and risk of adding more vehicles and personnel. As shown in this thesis, network-centric weapon systems also introduce the critical, strategic-level risk that the network could fall victim to an asymmetric attack. Given the expertise and opportunity, a much smaller opposing force might discover critical information, plant false data into friendly networks, and shut down the network entirely.

Industry experience with large-scale computer networks shows their lack of security and the consequences of security-related attacks. Given the schedule and budget pressures upon the defense industry, network-centric systems will likely use commercial hardware, software, and protocols for some of their construction (Camana 2009, 2-PM-28). Therefore, many of the security practices seen on commercial office-automation and electronic commerce systems carry through to these future weapon systems and their supporting networks. Unfortunately, much of the scholarship on hacking and computer security focuses on technical attacks on hardware and software and ignores the multi-faceted systems engineering approach necessary to protect a weapon system.

The overall system-level security of a network-centric weapon system can be described as a combination of network and computer security, information security, physical security, operational security, and personnel security. Each aspect brings unique dangers and protection methods that must be considered in the larger context of the military operation. A military adversary will seek the most efficient means to weaken the advantage provided by the network, so developers must seek the systems perspective and guard against the most efficient attacks. No computer system, and certainly not one as complex as a tactical military network, can be completely safe against all attacks on confidentiality, integrity, and availability, but the efficient and comprehensive protection of the network and all its components may stave off a compromise long enough for friendly forces to achieve their objectives.

The systems engineering approach to security applies throughout the system life cycle to program managers, developers, commanders, operators, and maintainers. At each stage, all these stakeholders should recall the importance of security and the consequences of attacks on confidentiality, integrity, and availability. They must maintain the systems engineering approach and consider the possible consequences to the larger system rather than focusing exclusively on their own area of specialization.

LIST OF REFERENCES

Anderson, Ross. 2008. *Security engineering*. Indianapolis, IN: Wiley.

Band, Steven, Dawn Cappelli, Lynn Fischer, Andrew Moore, Eric Shaw, and Randall Trzeciak. 2006. *Comparing insider IT sabotage and espionage: A model-based analysis*. Carnegie Mellon University. http://www.sei.cmu.edu/reports/06tr026.pdf (accessed August 30, 2009).

Behar, Richard. 2008. World bank under cyber siege in 'unprecedented crisis.' Fox News. October 10. http://www.foxnews.com/story/0,2933,435681,00.html (accessed August 30, 2009).

Boak, David. 1973. A history of U.S. communications security. Lectures for the National Security Agency in 1966.

Camana, Peter. 2009. Current and future military data links. Lecture, Technology Training Corporation seminar, San Diego, CA. March 12.

CNN. 2007. This week at war. February 24. http://transcripts.cnn.com/TRANSCRIPTS/0702/24/tww.01.html (accessed August 30, 2009).

Comer, Douglas. 1999. *Computer networks and internets.* Upper Saddle River, NJ: Prentice-Hall.

Commission on Physical Sciences, Mathematics, and Applications. 2000. *Network-centric naval forces: A transition strategy for enhancing operational capabilities.* Washington DC: National Academies Press.

Committee on National Security Systems. 2003. NSTISSP no. 11, revised fact sheet: National information assurance acquisition policy.

———. 2006. Instruction no. 4009: National information assurance glossary.

Defense Industry Daily. 2008. GAO Future Combat Systems Report. March 23. http://www.defenseindustrydaily.com/GAO-Future-Combat-Systems-Report-2008-04809/ (accessed August 30, 2009).

Department of Defense. 2003. Instruction 5000.2: Operation of the defense acquisition system.

Dhamija, Rachna, J.D. Tygar, and Marti Hearst. 2006. *Why phishing works.* Paper presented at the Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada. April 22.

Dudney, Robert. 2009. The 75 percent force. *Air Force Magazine*, March.

Dyer, Joan, Mark Lindemann, Ronald Perez, Reiner Sailer, Leendert van Doorn, Sean W. Smith, and NS Steve Weingart. 2001. Building the IBM 4758 Secure Coprocessor. *Computer*, 34, no. 10 (October): 57–66.

Epstein, Keith. 2008. U.S. is losing global cyberwar, commission says. *BusinessWeek*. December 7.

Fritz, Jason. 2008. How China will use cyber warfare to leapfrog in military competitiveness. *Culture Mandala* 8, no. 1 (October): 28–80.

Globalsecurity.org. Future Combat Systems. http://www.globalsecurity.org/military/systems/ground/fcs.htm (accessed July 17, 2009).

Goshorn, Rachel. 2008. Findings for network-centric systems engineering education. Paper presented at MILCOM 08, San Diego, CA. November 17.

Harris, Shon. 2008. *CISSP all-in-one exam guide.* New York: McGraw-Hill.

Haykin, Simon. 2002. *Adaptive filter theory.* Upper Saddle River, NJ: Prentice Hall.

Higginbotham, M. Douglas, Albert Milheizler, Joseph Maley, and Bernard Suskie. 1998. Integrating information system security engineering with system engineering with system engineering tools. *Proceedings of WETICE '98.*

Hurwitz, Judith, Robin Bloor, Carol Baroudi, and Marcia Kaufman. 2007. *Service oriented architecture for dummies.* Indianapolis, IN: Wiley.

Jackson, William. 2008. Protect your crypto key from the cold. *Government Computer News*. August 8.

Johns Hopkins Applied Physics Lab. 1995. The cooperative engagement capability. *Johns Hopkins APL Technical Digest* 16, no. 4: 377–96.

Joint Software System Safety Committee. 1999. *Software system safety handbook: A technical and managerial team approach.* http://www.egginc.com/dahlgren/files/ssshandbook.pdf (accessed August 30, 2009).

King, Samuel, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. 2008. Designing and implementing malicious hardware. Paper presented at LEET 08, San Francisco, CA. April 15.

Larman, Craig. 2006. *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development.* Upper Saddle River, NJ: Prentice Hall.

Lind, William, Keith Nightengale, John Schmitt, Joseph Sutton, and Gary Wilson. 1989. The changing face of war: Into the fourth generation. *Marine Corps Gazette.* October.

McClure, Stuart, Joel Scambray, and George Kurtz. 2005. *Hacking exposed: Network security secrets & solutions*. New York: McGraw-Hill.

Naval Studies Board. 2008. Information assurance for network-centric naval forces. http://www8.nationalacademies.org/cp/projectview.aspx?key=48902 (accessed August 30, 2009).

Nolan, John. 2000. *A case study in French espionage: Renaissance software*. Department of Energy Office of Counterintelligence. http://www.hanford.gov/oci/maindocs/ci_r_docs/frenchesp.pdf (accessed August 30, 2009).

Null, Linda, and Julia Lobur. 2006. *Computer organization and architecture.* Sudbury, MA: Jones & Bartlett.

Paul, Richard, Robert Niewoehner, and Linder Elder. 2007. *The thinker's guide to engineering reasoning.* Dillon Beach, CA: Foundation for Critical Thinking Press.

Peikari, Cyrus, and Anton Chuvakin. 2004. *Security Warrior*. Sebastopol, CA: O'Reilly.

Office of Force Transformation. 2005. *The implementation of network-centric warfare.* Washington, DC: Department of Defense.

Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. 2008. *Systems Engineering Guide for Systems of Systems, Version 1.0*. Washington, DC: ODUSD(A&T)SSE.

O'Neil, William. 2007. *The cooperative engagement capability (CEC): Transforming naval anti-air warfare.* Washington DC: Center for Technology and National Security Policy.

Osborne, Kris. 2009. FCS is dead; programs live on. *Defense News,* May 18.

Schneier, Bruce. 2000. *Secrets and lies.* New York: John Wiley & Sons.

———. 2004. Introduction to the second edition of *secrets and lies*. http://www.schneier.com/book-sandl-intro2.html (accessed on August 30, 2009).

———. 2008a. The DNS vulnerability. *Cryptogram*, August 15. http://www.schneier.com/crypto-gram-0808.html (accessed on August 30, 2009).

———. 2008b. Photo ID checks at airport. *Cryptogram*, September 15. http://www.schneier.com/crypto-gram-0809.html (accessed on August 30, 2009).

Signatories of the Common Criteria Recognition Agreement. 2009. Common criteria for information technology security evaluation, version 3.1 revision 3.

Stytz, Martin, and Sheila Banks. 2004. *Issues and requirements for cybersecurity in network centric warfare.* Wright Patterson AFB, OH: Air Force Research Laboratory.

Tkacik, John. 2008. Trojan dragon: China's cyber threat. *Backgrounder*, no. 2106 (February 8).

U.S. Congress. U.S.–China Economic and Security Review Commission. 2008. *China's proliferation practices, and the development of its cyber and space warfare capabilities*. 110th Cong., 2nd sess., May 20.

Zittrain, Johnathan. 2009. Lost in the cloud. *New York Times*. July 19.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California