5-2015

# Security Through Exclusivity: Information Assurance via a DOD Operating System

Karen M. Littlewood

## APUS Library Capstone Submission Form

This capstone has been approved for submission to and review and publication by the APUS Library.

| | | | |
|---|---|---|---|
| Student Name [Last, First, MI] * | Littlewood | Karen | |
| Course Number [e.g. INTL699] * | ITCC698 | Paper Date [See Title pg.] | 04/2015 |
| Professor Name [Last, First] * | Dr. Miriam Masullo | | |
| Program Name * See list | Master of Science in Information Technology | | |
| Keywords [250 character max.] | operating system, security, information assurance, JIE, DOD, STIC | | |
| Passed with Distinction * Y or N | Y | | |
| Security Sensitive Information * Y or N | N | | |
| IRB Review Required * Y or N | N | If YES, include IRB documents in submission attachments. | |
| Turnitin Check * Y or N | Y | All capstone papers must be checked via Turnitin. | |

* Required

## Capstone Approval Document

The thesis/capstone for the master's degree submitted by the student listed (above) under this title *

SECURITY THROUGH EXCLUSIVITY: INFORMATION ASSURANCE VIA A DOD OPERATING SYSTEM

has been read by the undersigned. It is hereby recommended for acceptance by the faculty with credit to the amount of 3 semester hours.

| Program Representatives | Signatures | Date (mm/dd/yyyy) |
|---|---|---|
| Signed, 1st Reader *<br>[capstone professor] | Miriam J. Masullo, Ph.D.<br>Digitally signed by Miriam J. Masullo, Ph.D. DN: cn=Miriam J. Masullo, Ph.D., o=American Public University, ou, email=miriam.masullo@mycampus.apus.edu, c=US Date: 2015.07.30 20:07:29 -04'00' | 07/30/2015 |
| Signed, 2nd Reader<br>(if required by program) | | |
| Recommendation accepted on behalf of the program director * | Novadean Watson-Stone<br>Digitally signed by Novadean Watson-Stone DN: cn=Novadean Watson-Stone, o=American Public University System, ou=APUS, email=nwatson@apus.edu, c=US Date: 2015.07.13 11:00:51 -04'00' | 07/13/2015 |
| Approved by academic dean * | Dan Benjamin<br>Digitally signed by Dan Benjamin Date: 2015.12.16 11:46:53 -05'00' | |

* Required

| | |
|---|---|
| **Send thesis submission to:**<br>ThesisCapstoneSubmission@apus.edu | **Attachments must include:**<br>• This completed form<br>• FINAL Thesis document as Microsoft Word file<br>• IRB Review docs (if applicable) |

# SECURITY THROUGH EXCLUSIVITY: INFORMATION ASSURANCE VIA A DOD OPERATING SYSTEM

A Master Thesis

Submitted to the Facility

of

American Public University

by

Karen Melissa Littlewood

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

April 2015

American Public University

Charles Town, WV

# DEDICATION

I dedicate this thesis to my husband.  It is with his strength, acumen, and loving support that I have been able to complete this work.  Life is best viewed through the eyes of another in order to gain different perspective and lucidity.

## ACKNOWLEDGEMENTS

I wish to thank all of the other students with which I have had the pleasure of getting to know, debating, and ultimately learning with. Additionally, I have benefited much from the real-world knowledge that has been bestowed upon me by instructors Dr. Joseph DiRenzo, Dr. David Anderson, and Dr. Francesca Catalano, who have all contributed to sparking my desire for lifelong learning.

## ABSTRACT OF THE THESIS

## SECURITY THROUGH EXCLUSIVITY: INFORMATION ASSURANCE VIA A DOD OPERATING SYSTEM

By

Karen Melissa Littlewood

American Public University System, March 29, 2015

Dr. Miriam J. Masullo, Thesis Professor

The Department of Defense (DOD) is continuously seeking to find new technologies that further the security posture of its information systems. Often these technologies surround the protection of its network infrastructure and the software that is utilized on the network, but this suggests that the underlying platform is inherently secure. It is possible to find a proprietary OS platform stack that is more secure than the current standard being utilized today. The benefits of a proprietary OS platform stack are numerous: enabling compatibility within DOD branches and agencies in accordance to DOD 8330.01 and 8100.04 Unified Capabilities and on the Joint Information Environment (JIE); a more cost-efficient method of testing, implementation, and maintenance conducted centrally through DISA and the JITC; a standard OS platform utilized on the JIE can be fully compliant with DOD IA requirements; compatibility simplifies the creation of policies and Standard Operating Procedures (SOPs) on a local level because all local commands can share documents as they are created or updated.

*Keywords:* operating system, security, information assurance, JIE, DOD, STIG

## TABLE OF CONTENTS

# LIST OF FIGURES

Security through Exclusivity: Information Assurance via a DOD Operating System

## Introduction

The Department of Defense (DOD) is continuously seeking to find new technologies that further the security posture of its information systems. Often these technologies surround the protection of its network infrastructure and the software that is utilized on the network, but this suggests that the underlying platform is inherently secure. The DOD currently deploys various OS platforms including Microsoft Windows, Linux, UNIX, and MAC most of which are Commercial of The Shelf (COTS) systems owned by commercial entities. Currently, the DOD has created DISA STIGs on various COTS platforms located on the DISA website. Because each configuration has its own STIG, it is assumed that there are many variations of OS platform implementation within DOD networks.

### Problem Statement

Microsoft Windows is the most common OS utilized on DOD networks. While Microsoft sells its products to the DOD, they essentially only sell the license to their product. At no time does the DOD actually own the rights to the OS platform itself, so the DOD is unable to customize the platform any further than is allowed by Microsoft. Additionally, there are many inherent security flaws within Windows OSs yet the DOD is unable to fully test and implement security patches because Microsoft doesn't distribute the OS source code. These security flaws exist through buggy code within the OS itself or are created by attackers who exploit vulnerabilities within the OS (Bassil, 2012). While the DOD continues to utilize Microsoft OS products, their limitations prevent the ability to fully implement a security OS platform on DOD networks. Additionally, because the DOD utilizes other OS platforms it is difficult for a network administrator to appropriately secure the network because each OS platform has unique vulnerabilities that can be exploited.

## Purpose

Historically, the DOD has sought to create a more secure OS design for use in the DOD networks. OS platforms such as the Department of Defense Kernelized Secure Operating System (KSOS), Provably Secure Operating System (PSOS), and the Kernelized Virtual Machine (KVM) were designed and tested in the with the intent of implementing the DOD's multilevel security practices and demonstrating the possibility of creating highly effective security systems (Bell, ND).

The DOD has had experience in technology research through The Defense Advanced Research Projects Agency (DARPA) who creates technologies for use within the DOD. DARPA was founded as the Advanced Research Projects Agency (ARPA) in 1958 and was involved in the creation of the Internet and global position system (Van Atta, ND). Today, DARPA is continuing to create new technologies, including research on OSs that could prove to be more secure than any Microsoft product. The Clean-slate design of Resilient, Adaptive, Secure Hosts (CRASH) is a DARPA program that is tasked with creating or finding computer systems that are more resilient to internet-based attacks by adapting to an attack, learning from previous attacks in order to prevent future attacks, and repairing itself if necessary after an attack. The CRASH research includes OSs, programming languages, processes and instructions sets, and application tools (DARPA, 2014). Capsicum is a FreeBSD OS utilizing the Linux kernel and allows compartmentalization and decomposition of larger applications and web services (DARPA, 2014).

## Research Question

The author proposes to research and discuss the information security implications of implementing a DOD proprietary OS. To accomplish this, this study is based on four research questions: who is responsible for creating and implementing DOD Information Assurance (IA) policies and procedures; what is the current DOD OS platform implementation; what is a possible OS replacement; how can this OS platform replacement provide an enhanced information security posture in comparison to the current OS platform?

## Hypothesis

The author hypothesizes that the current implementation of various OSs can be replaced by a more standardized yet customizable solution. The author also hypothesizes that all applications can still be utilized on the alternative platform with little to no downtime. Lastly, the author hypothesizes that an alternative OS can provide a greater secure security posture than the current Microsoft Windows configuration.

## Significance of Study

To answer the first area, the author will research and discuss the US governmental agencies that are in charge of DOD IA policy creation and implementation. These agencies are tasked with creating IA standards that must be implemented on all DOD information systems and its components. The author will also discuss the current IA standards utilized by the DOD and how these standards affect DOD information system component selection and implementation. The role of OSs will be discussed in relation to DOD information systems. Utilizing this research, the author can determine the policies and procedures required to evaluate and approve an OS.

Second, the author will research and discuss the current DOD OS implementation. This includes the DOD’s current selection, testing, and security control guidelines, as well as implementation requirements. The author will also provide previous research about the DOD's OS implementation and any problems with the current configuration. The current OS capabilities, sustainability, and security posture will be discussed.

Lastly, the author will discuss possible OS replacements to the DOD's current OS. In order to accomplish this, the author will discuss possible OS replacements by function, capabilities, sustainability, and security posture. If possible, the proposed OS will be analyzed based on DOD guidelines and requirements to discuss whether the current configuration will meet DOD IA security controls or what would be required to do so.

## Definition of Unclear Terms

An operating system is the intermediary between the hardware that makes up a computer and the applications software that enables the computer to execute a specific task or function. It is composed of a set of programs that allow the hardware to complete the functions required from the application software. Operating systems complete three specific tasks: to implement and monitor the computer's main resources such as memory, CPU, and disk drives; accomplish services for application software; launch the user interface. Although the operating system completes much of its tasks without a user's knowledge, it is the operating system that executes commands for application input and output (Fay-Wolfe, ND). Operating systems can be divided into stand-alone computers or a network operating system, which has many computers and sometimes a server that stores data or executes hardware and software functions for the other devices on a network (Lemley, ND).

Operating systems have historically been designed with function in mind, not security because security typically slows down the processing speed of an operating system or limits the access of a user based on the security permissions. Users continuously utilize an operating system by executing commands via applications software and the operating system constantly interacts with the user by controlling what the user can do on the computer. Herein lays the key security flaw of an operating system; although an operating system controls what the user does on the computer via restricted access to the computer's resources, a user can also elevate their access to these resources if the operating system software is vulnerable (Lemley, ND).

## Limitations/Delimitations

Because this thesis will analyze the configurations of DOD networks, the author will only be able to discuss publically available information. This means that any DOD information that hasn't been deemed for public release will not be utilized within this study. Additionally, because this thesis hasn't been approved by the DOD, the author can only provide recommendations on an academic level. However, this study can be utilized as a starting point to open up discussion for further research into the usage of proprietary OS platform stacks for use within the DOD's Global Information Grid) GIG and eventually the Joint Information Environment (JIE).

The author has limited the discussion of proprietary DOD OS platforms to only a few in order to provide an example of the types of DOD platforms that have been created in the past and which are currently being utilized or developed today. This thesis does not attempt to list all proprietary OS platforms. Additionally, this thesis will not discuss the performance factors of each OS platform and instead the author will only discuss the security controls and security capabilities of each. A more technical discussion is not possible due to the classified properties of each OS, whether it is proprietary or COTS, as they are configured on the DOD network.

## Assumptions

Within this paper, the author assumes that all DOD networks follow the information security controls that are mandated by the U.S. Government. The author also assumes that while most DOD networks utilize the same OS base image, each separate service will implement their own software services platform, which consists of software applications that pertain to work they conduct within their own military branch. For this reason, the author will not differentiate between different OS platform implementation within each DOD military branch or program. The author also assumes that the benefits of implementing a government-owned OS platform outweigh the cost of research and development and implementation of the system.

## Theoretical Framework

This thesis employs descriptive research and qualitative analysis approach methods. The author will discuss and summarize information security policies and controls that have been created and utilized within the various DOD networks as well as the impending implementation of the JIE. The author will also discuss creation and implementation of select government owned operating systems (GOTS). To ensure that the author has provided an adequate foundation to the research question, archival research will be used to discuss previous research that has been conducted in the area of GOTS conception in order to implement a greater security posture. This will consist of a review of the literature, organizational records, and other types of media that contains records of previous research. Qualitative analysis will be utilized to review, analyze, and deduce conclusions based on the archival research presented within the literature review.

# Literature Review

## DOD Information Assurance Requirements

From conception, the computer operating system has always been built with ease of use and with processing capabilities in mind. As technology became more common place it became apparent that both the data and the computer systems themselves must be protected from data loss, theft, or exploitation. The DOD has historically been aware of the value of data security and has always taken strong actions to protect its assets. In the 1990's, implementing security within operating systems became an increasing concern for federal agencies which would spend large amounts of their budget on new technology. Unfortunately, much of their technology quickly became obsolete, was incompatible with emerging systems and technology, or wasn't configured properly to adequately address the concerns of the organization. The Clinton administration sought to rectify the increasing problem of inefficiencies in government spending on technology and implemented the Clinger-Cohen Act of 1996, which required that the Office of Management and Budget (OMB) to be in charge of establishing information technology directives for all federal agencies. In 2000, the OMB released the OMB Circular A-130 which utilized appendices to provide policies and procedures that are applicable to all federal agencies within the executive branch. Appendix III, *Security of Federal Automated Information Resources*, within the OMB Circular A-130 establishes a policy, procedure, and analytic guidelines to managing Federal information resources (The White House, 2000). While the OMB Circular A-130 was a big step towards mandating information security requirements to all federal agencies, there were no federal laws or regulations that required agencies to comply with its policies because unlike laws or regulations, OMB Circulars are simply instructions on how to do something. In order to force agencies to comply with their Circular, the OMB needed a law or regulation that mandated compliance within all federal agencies. In 2002, the Federal Information Security Management Act (FISMA) was passed.

### FISMA.

The E-Government Act (Public Law 107-347) was passed in 2002 to implement a Federal Chief Information Officer within the OMB as well as creating a structure of utilizing Internet-based information technology in order to provide information and services to the general public (U.S. Congress, 2002). Title III, or FISMA, describes the requirements that all federal agencies should develop, document, and implement in order to secure their information systems and assets. This protection extends to all other organizations supporting or working with federal government agencies such as federal contractors (NIST, ND). There are several federal agencies tasked with choosing the appropriate DOD operating system and applications that are approved to be utilized on the network. However, each agency is required to follow the laws and regulations regarding the security of information systems as defined in FISMA. In order to make a more standard and comprehensive set of security controls and requirements, the NIST security team, the DOD, the intelligence community, the Committee on National Security Systems, and the DHS have teamed together to create the Joint Task Force Transformation Initiative (JTFT) which is an inter-agency working group that is tasked with crafting a Unified Information Security Framework within the U.S. government. With other documents, the JTFT released the Security and Privacy Controls for Federal Information Systems and Organizations (SP 800-53) in order to address emerging information assurance issues within the US government.

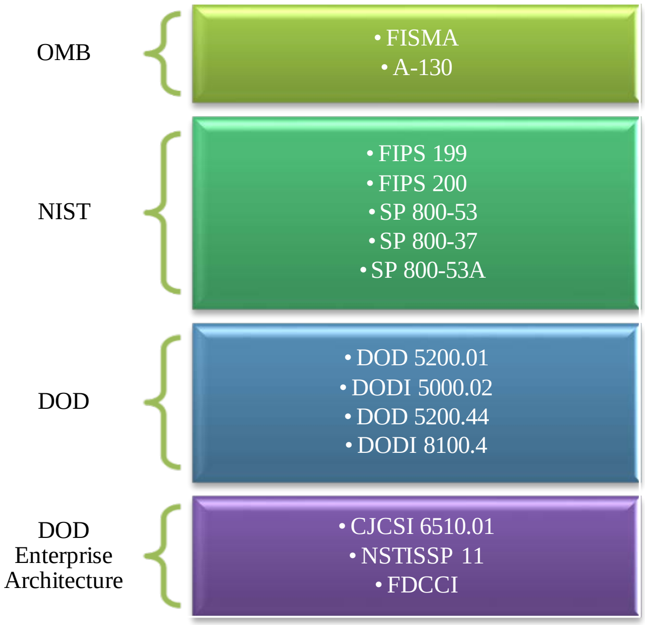


**Figure 1Hierarchy of Information Security Standards and Guidelines**

**NIST.**

The National Institute of Standards and Technology (NIST) is a federal agency with the U.S. Department of Congress which is tasked to improving economic security, U.S. industrial competiveness and innovation by utilizing science, standards, and technology (NIST, 2015). In response to FISMA requirements, NIST created various publications that address the mandatory federal standards. FIPS Publication 199, *Standards for Security Categorization of Federal Information and Information Systems* is utilized by federal agencies to determine the security category of their information systems. This document details the importance of determining at what level of compliance to FISMA an agency's information system must be before implementing controls from SP 800-53 because security controls cost money and may reduce the amount of profit or output generated by the information system. An agency that falls in a lower security category require fewer security controls than one that has a higher security category. In response to this, NIST created FIPS Publication 200, *Minimum Security Requirements for*

*Federal Information and Information Systems* which describes the minimum security controls and requirements needed to become compliant with FISMA. Once an organization has determined their security category, they can utilize FIPS 200 to determine what the impact of the security controls will be on their information system. Then the organization can tailor the security controls of SP 800-53 based on their security category and minimum security controls required to secure their information system. This is helpful in allowing an organization to tailor their baseline security posture to meet the needs of their mission and business requirements yet ensuring they meet the minimum standards as defined in FISMA. In this way, both FIPS 200 and SP 800-53 work together to allow all types of businesses to establish due diligence in securing their information systems and providing compliance to federal information systems.

To ensure that these agencies comply with FISMA, the NIST Information Technology Laboratory (ITL) guidelines were implemented by the DOD in order to create federal government security control standards (NIST, 2013). The ITL creates and provides testing, research, data, and technical analysis to further the study of information security standards and guidelines within federal information systems. They also develop research studies, guidelines, and outreach to the information system security community including government, academia, and the commercial industry.

**Special Publication 800-53.**

The DOD utilizes the NIST Special Publication 800-53, Revision 4, “Security and Privacy Controls for Federal Information Systems and Organizations” to ensure that all responsibilities outlined by the Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347 and Circular A-130 Section 8b(3), *Securing Agency Information Systems* are followed ( NIST, 2013). SP 800-53 was developed specifically for federal information systems and is utilized to provide security controls to address physical and structure failures, natural disasters, hostile cyber-attack, and human errors. Additionally it provides the ability to implement security controls within information systems while adhering to various federal executive orders, regulations, laws, policies, directives, standards, mission, and business needs as directed by the federal government. Because the federal government has many different types of information systems that are utilized for widely different missions and environments, the SP-800-53 provides the customization of information systems because each type of system differs in technology, environment, operational need, or function. NIST security controls address both functional security needs, such as the necessary requirements that allow the systems to remain secured and security assurance, which describes how much confidence is placed in the security asset. This is important because both of these factors describe how trustworthy the security controls and the systems that implement them are.

While NIST has endeavored to create comprehensive and inclusive security controls within SP 800-53, some research considers them to be flawed. NIST 800 series publications describe information security standards but they are not actually executed on the DOD information network (DOD IN) until expressively authorized by the branch or agency within the DOD (NIST, 2013). Even though an organization can ensure that SP 800-53 is fully implemented, this doesn’t mean that the organization’s information system is fully functional or that all components can work with each other (Lam & Carayannis, 2011). SP 800-53describes only two levels of implementation (yes or no) which lacks flexibility and means that an organization can only choose to become implemented instead of choosing to utilize minimal, moderate, maximum levels of implementation. Although each agency is required to implement controls on their specific level, there is no accountability method defined to hold the organizations accountable for full implementation. There are no controls for ensuring that employees are being held accountable or understand the rules of their behavior in response to the control. NIST utilizes the process of "Certification and Accreditation" which states that an organization which runs compliance testing on their information systems will create a greater security posture. However, compliance testing can only protect from vulnerabilities that have already been identified; this doesn't provide protection for unknown vulnerabilities. Since the publication of SP 800-53, NIST has addressed this problem by publishing Special Publication 800-37 *Risk Management Framework (RMF)* which addresses the process of continually identifying risk including vulnerabilities and threats to an information system (Lam & Carayannis, 2011).

FISMA itself has been criticized for lacking the ability to quickly react to emerging threats and vulnerabilities. FISMA also requires agencies to utilize "Certification and Accreditation" via annual assessments and reports in order to be deemed compliant or non-compliant. However, the cost of implementing SP 800-53 standards is extremely high and doesn't ensure that an agencies information system will be secure, nor does a properly secured system ensure compliance under FISMA. New legislation dubbed FISMA 2.0 may implement stricter guidelines in reporting data breaches currently being required. FISMA 2.0 will also be called the Federal Information Security Modernization Act, describing the modernization of the law in which it would require agencies to utilize tools that would provide automated security diagnosis and/or remediation (Sustar, 2015). If this legislation is passed, NIST SP 800-53 will also require modernization in order to remain applicable to government information system security.

## DOD Directives and Instructions

Other information assurance requirements fill in the gaps of the DOD's regulations and responsibilities by agency. The DOD utilizes Directives, Instructions, Manuals or Publications, and Memos to relay information and instructions to all DOD commands. Directives are utilized to institute policies, allocate responsibilities to a particular agent, and give the authority to the responsible DOD section who will institute the policy. There are two types of DOD Directive; one assigns direct oversight in relation to a policy, and the other establishes a charter regarding a specific DOD agency including the mission, function, authority, and its relationships to other agencies. An Instruction also institutes or implements a policy but is separated into either a policy instruction or a non-policy instruction. It may be follow-on to a directive in order to implement the policy that was described within a directive. In an Instruction, the directions are more detailed describe how to carry out the policy. A Manual or Publication also complements a director or instruction by creating a consistent procedure to manage a procedure or specific operating system. It is approved by the Directive or Instruction that is references. DOD Memos are utilized when it is necessary to create a policy through a Directive or Instruction but there is not enough time to implement a new or change an existing policy (DTIC, ND).

The DOD 5200.01 *DOD Information Security Program: Overview, Classification, and Declassification* describes DOD agencies and what their function is within the overall Information Security Program such as the Defense Technical Information Center (DTIC), who is in charge of maintaining an index of security classification guides which are accessible and utilized online via dtic.mil (DOD, 2012). These guides are integral for all other DOD agencies to communicate how a security classification guide is created and distributed and find the policy and procedures relating to their Information Security Program. DOD 5200.44 *Protection of Mission Critical Functions to Achieve Trusted Systems and Network (TSN)* describe the critical information systems utilized within all DOD and Joint Intelligence programs and the responsibilities that each agency has in order to minimize risk to the information system (DODI, 2012). Although a short document, it outlines the responsibilities of the various agencies and commands that make up the DOD in order to ensure that each agency communicates with each other, utilizes the appropriate analysis and mitigation functions to protect their systems, and documents any vulnerability and/or other findings. Both of these documents break down the various components of the IA program and who does what. The Chairmen of the Joint Chiefs of Staff Instruction *Information Assurance (IA) and Support to Computer Network Defense (CND)*, or CJCSI 6510.01F describes how the DOD information assurance program works with joint policy and who is responsible for actually implementing the policy on DOD Information Networks (INs) (CJCSI, 2011). Several more documents exist that describe different parts of DOD policy, regulation, and responsibilities that federal agencies must comply with in order to implement their information security program. These documents are outlined in *Figure 2 DOD Interoperability Certification Directives and Instruction*.

**DISA**

The Defense Information Systems Agency (DISA) was established in 1960 as the Defense Communication Agency (DCA), which united separate military telecommunications branches in order to establish a centralized and integrated DOD telecommunications network. Today, DISA delivers, controls, and secures global DOD enterprise information infrastructures that are capable of command, control, and information sharing abilities in direct support of the DOD's military programs and operations (DISA, ND). Since 1998, DISA has implemented Security Technical Implementation Guides (STIGs) which are used to provide technical guidance that enables the DOD to appropriately harden information systems and software in order to reduce vulnerabilities and exploitation by malware and attackers. STIGS are created for each product or device that is tested and approved to be utilized on the DOD's entire US government IT infrastructure, known as the Global Information Grid (GIG). Because each approved device is required to meet DOD information assurance requirements, the company may be required to alter the code of their devices in order to be approved for DOD use (NA, 2013). DISA control checklists are utilized and published online in order to allow a DOD customer to have step-by-step instructions in how to ensure their information assurance controls are implemented according to the DOD guidelines. In 2008, DISA changed from utilizing 8500 information assurance controls to utilizing NIST SP 800-53 controls.

### JITC.

As with other DOD policies, instructions, and events, there are several policies that define the requirements for Joint Services Interoperability Certification. Title 10 of the United States Code: Section 2223 states that the DOD Chief Information Officer (CIO) is responsible to, "Ensure the interoperability of Information Technology and National Security Systems throughout the DOD." (as quoted in DISA, 2009, slide 3). There are several key policies to ensure that all DOD branches and agencies accomplish this: CJCSI 6212.01E, DODD 4630.5, DODI 4630.8, CJCSI 3170.01G, and the DOD 500 Series all describe the requirement creating interoperable IT and National Security Systems (NSS) within all parts of the DOD. Additionally, most of these directives and instructions also reference the Joint Interoperability Test Command (JITC) in particular.

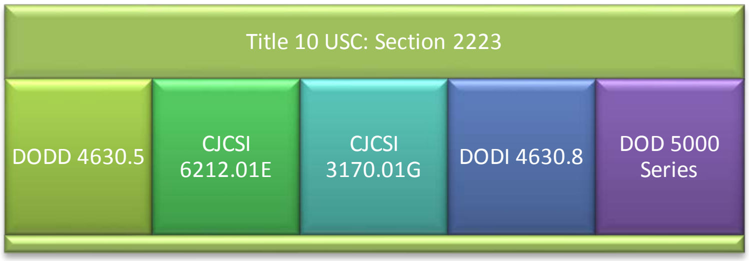


**Figure 2 DOD Interoperability Certification Directives and Instruction**

The JITC, located in Fort Huachuca, is the sole responsible non-military agency that is tasked with testing all IT and NSS for interoperability and compliance. They also issue certifications to those systems that pass as designated in “DODI 8330.01: Interoperability of Information Technology (IT), Including National Security Systems (NSS)”. Potential products that are utilized on the DOD IN must be IT and NSS compliant, and therefore must obtain certification from JITC. To accomplish this, the JITC acts as a Joint Interoperability Certifier, Operational Test Agency, and even provides Warfighter and Coalition Interoperability Support and evaluations to all military branches during their exercises. During the testing, the JITC seeks to ensure that the products not only comply with DISA’s STIG, DOD IA standards, but is also compatible with other DOD agencies, military branches, and US coalition partners. There are two different types of testing: standard testing ensures that the products conform to developmental or authentication testing; full capacity testing ensures that the product meets full operational conformance. Services are provided to JITC customers through hotline support for customers who are forward deployed, the Joint Interoperability Tool, which provides customers with access to an extensive database with key interoperability information and references, and the System Tracking Program, which tracks a particular system’s development towards joint interoperability certification. The JITC Lessons Learned Reports provides product users with more complete information about the product that isn't readily available in the product's manual or technical information.

The process to implement COTS products within the DOD IN is long for the DOD and for the manufacturer. Before any products are introduced onto the DOD IN, they must meet the security guidelines that are specified by DISA. If a vendor wishes to be approved for use on DOD IN, they must first be designated by the Joint Staff certified requirements document which states the need for the product. If not, the vendor will only receive an assessment. Certification can be carried out through the requirements as stated in DOD 8330.01 or DODI 8100.04, *DOD Unified Capabilities (2010)*, which describes the DOD's requirements to create and utilize products or systems with unified capabilities in DOD IN (DOD, 2010). While the interoperability of IT and NSS and Unified Capabilities (UC) may seem similar, they are not. UC ensures that video, voice, and data services are completely interoperability over the DOD IN network infrastructure regardless of technology, standard, or vendor. (JITC, ND). DODI 8100.04 also describes DISA and JITC's responsibility to implement IA and Interoperability (IO) testing as an UC certification authority. JITC, along with other DOD Component test labs, evaluates and certifies products in accordance to DODI 8100.04. If the product is approved, they are placed on the UC Approved Products List (APL), which is managed by JITC. It is important to note that products that go through UC testing and certification are not liable for being tested according to DOD 8330.01, because they UC APLs already meet the interoperability requirements.

There are several parts to the accreditation process. The product should first be tested for IA requirements and if the product passes it is moved on to the second phase called Interoperability (IO) testing. If the product doesn't meet IA requirements, the vendor must correct and deficiencies or vulnerabilities before the product will be considered again. The product is approved when it can pass both phases and earns both an IA accreditation and IO certification (JTIC, 2009, pg. 2). The JITC also conducts IA assessments in order to test and retest products that are on the Defense Switched Network (DSN) in order to ensure that the products continue to meet IA requirements as dictated in the DOD Information Assurance IA Certification & Accreditation (C&A) Process (DIACAP).

### STIGS.

As described earlier, STIGs are created by DISA to both test and provide technical guidance that enables a DOD agency to configure hardware, software, and other components on their IN to reduce vulnerabilities and exploitation by malware and attackers. Potential products are tested for STIG compliance by the JITC who is responsible for information system testing in order to ensure all software, hardware, and other components adhere to DISA's STIGs. These STIGs are the actual documentation that all DOD IN administrators utilize to implement all components of their information systems. For every component implemented on the network, the administrators should ensure that the component complies with the STIG. If the component doesn't have a STIG, it isn't approved to be on the network (DISA, ND). Other types of test components include bulletins, announcements, and alerts that are sent out via IA Vulnerability Management announcements. The JITC utilizes a test plan to implement STIGs on the network. Once the test is complete, the JITC creates an IA Assessment Finding and Mitigations Report, which is sent to DISA Field Security Office (FSO) for any comments. The vulnerabilities found during the test are then sent to the vendor so they can further eliminate vulnerabilities in their products. The overall goal of this test plan is to issue out a Certification and Accreditation letter to the Defense Information Security Network (DISN) Security Accreditation Working Group

(DSAWG) so these products can be utilized on the network (DISA, 2009). Both DISA and NSA utilize security configuration guides to enable IA security controls on all GOTS and COTS products.

> "All IA and IA-enabled government-off-the-shelf (GOTS) and commercial off-the-shelf (COTS) hardware, firmware, and software components must be acquired, evaluated, installed, and configured IAW National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, "National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology (IT) Products" (reference l)." (CJCSI, 2011. pg. A-3)

The CJCSI describes Ports, Protocols and Services Management (PPSM) as the requirement for all DOD project managers, engineers, and system administrators to utilize PPSM Category Assurance Lists to conduct risk management practices such as certification, accreditation, configuration, and deployment within their DOD network. DISA manages the PPSM process and it is best used for configuring network security devices such as firewalls, routers, and IDS/IPS on a DOD network (DISA, 2010). Various agencies depend on the PPSM program including the DOD, non-DOD federal agencies, DSAWG, and others. New IS software can be approved by an Authorizing Official once it has been registered and follows the DIACAP process, and been issued an approval of certification and accreditation. The DIACAP process means that the software is registered within the PPSM program, an assessment has been conducted, vulnerabilities have been defined and mitigated within documentation and the date of approval and expiration date is added on the DIACAP scorecard. PPSM utilizes assurance levels which are signified by colors and divided into separate groups according to the usage approval. Green levels describe the best security practices, services, or protocols and represent all that are agreed of by all combatant commands, services, defense agencies, DOD field activities, and joint activities to be approved for utilization in all ISs including those that are under development or redesigning. Yellow protocols are deemed acceptable when used when their technical vulnerabilities have been diminished. Orange or controlled protocols required approval based on their need during an operation and are not allowed to be utilized on developmental or procured ISs. In order to utilize orange protocols, DSAWG must review and approve the use based on the operation. Red protocols are banned from any use (DISA, 2010). In order to implement the most current PPSM Category Assurance List, the DOD releases new STIGs in which the most updated vulnerability assessment report is released. An agency can request an exception package if their asset doesn't meet the PPSM requirements by submitting a system identification profile (SIP), operational need, a DIACAP implementation plan, as well as network configuration information (DISA, 2010).

**Joint Information Environment**

The DOD is currently implementing the Joint Information Environment (JIE) which allows the sharing of information between all DOD branches via shared IT infrastructure, single security architecture, and enterprise services (Edwards, 2013). This is a fundamental shift in the way that the DOD has organized its Global Information Grid (GIG) in the past, which consisted of various networks with no enterprise management and which configured/administrated each network according to the command's individual requirements. Often these networks lack different levels of interoperability and standardization within their networks all the way down to OS platforms, system architectures, and applications. The goal of consolidating these separate networks into a JIE environment will allow the DOD to become more effective at enabling security, supporting next generation capabilities, and provide more flexibility within a single joint enterprise IT platform (DISA, 2014a). DOD Director of Information Management Michael Krieger stated, "We are hoping to enable a very agile collaborative environment" (Ackerman, 2005, para.5). The formerly named GIG was known to have extreme limitations on the ability for data to be distributed and enable collaboration between different DOD entities. Upon JIE implementation, a Global Enterprise Operations Center (GEOC) will direct all JIE operations on a global scale, prioritizing cyber missions globally, and providing a focal point for other partners within the US government and maintaining a global situational awareness. Falling under the GEOC are Enterprise Operations Centers (EOCs) which direct commands on a regional level and provide regional situation awareness. Regional EOCs also direct Computer Network Defense. Individual bases, posts, camps, or stations provide a local scope by directly working with JIE multinational partners; they provide face to face network support and infrastructure.

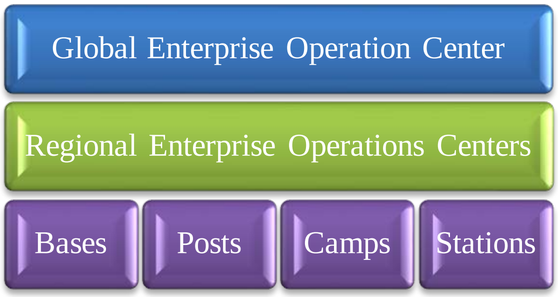

Global Enterprise Operation Center
Regional Enterprise Operations Centers
Bases
Posts
Camps
Stations

**Figure 3 GEOC Hiearchy**

DISA's (2014, pg. 10) whitepaper *Shaping the Enterprise for the Conflicts of Tomorrow* states "The current system creates too many opportunities for failure and leads to unnecessary exposure of our cyber assets. Additionally, every misconfigured system that is vulnerable to a threat leaves the rest of the enterprise just a vulnerable and further expands our cyber-attack surface." The JIE will alter the focus from securing decentralized network boundaries to creating a Single Security Architecture (SSA), centralized configuration management, and standard security implementations that are deployed at the same time as each other. Lt-Gen Ronnie D. Hawkins

Jr., DISA's Director stated in "Strategic Plan 2014-2019: Version 2" (DISA, 2014a, pg. 4) that DISA focus falls on many goals including:

- Lead the DOD in teaming with industry to build out the DOD public and private clouds, while recognizing the implied and explicit security expectations from DOD senior leadership.
- Reorient the Agency in a purposeful and strategic manner, to be more agile and responsive to a dynamic "IT Enterprise," thus requiring operational and tactical shifts in our organizational and personnel alignment.
- Establish a core capability for DOD to realize efficiencies through application rationalization, accomplished through a systematic process of identifying opportunities for consolidation, modernization and the sun setting of legacy systems.
- Institutionalize a culture of IT shared services both internal and external to the Agency. We can no longer operate multiple, stove-pipe operations and capabilities at the expense of the Agency as a whole. Neither can we, as the DOD's leader in IT, continue to unnecessarily build and house multiple one-off IT capabilities.

In addition to these objectives, DISA is also tasked with reassessing its process of evaluating devices and creating STIGs because the current process takes too long from evaluation to implementation. By the time a device or technology was approved be utilized on a DOD IN most technology is outdated or even obsolete. Former DOD CIO Teri Takai states, "This is not simply about embracing the newest technology, it is about keeping the department's workforce relevant in an era when information accessibility and cybersecurity play a critical role in mission success." (DISA, 2014a, pg. 19). One specific market that will implement changes in its STIG process is Defense Enterprise Mobility. Currently, the DOD evaluates mobile devices that are already on the market and if a device fails evaluation, the device manufacture must alter the device in order to make it conform to the security specifications of the DOD. This takes so much time that the device's configuration may be obsolete in comparison with the same model that has been released to the commercial market. The new STIG process would ensure that mobile devices are up to date with its commercial counterpart because the process of evaluating and creating STIGs has been streamlined to allow DISA to specify IA requirements before any devices have been selected or created by a specific manufacture. As the manufacturer develops a device, they are also tasked with creating STIG guidance and must deliver both the device and the STIG guidance to DISA for compliance review at the same time. This eliminates the need for DISA to test devices that are already on the market and require the manufacturer to alter the device's current configuration. Instead the device has been initially created in accordance to the DOD's IA requirements. The updated STIG process can save months to a year in time from the time a device is reviewed to implement.

However, STIG policies are not the only policy that is being revised in order to meet the requirements of the JIE. In accordance with ensuring the complete transition from the current GIG structure and implementation to a JIE enterprise architecture environment, the DOD is in the process of developing new policies, instructions, and procedures to reflect the new JIE architecture. IN 2010, DODI 8100.04 instituted a policy that determines how products and services can be listed in the Unified Capabilities APL. In 2012, the DOD created DOD IEA (Information Enterprise Architecture) v2.0 which establishes the JIE Enterprise Architecture (EA). Instituted in 2013 DODI 8330.aa discusses the institution enterprise services interoperability certification process. This DOD instruction outlines capabilities and architecture required in order for services to become certified and also discusses the methodology of interoperability analysis. Instituted in 2014, DODI 8310.aa outlines the process of detecting, developing, and suggesting IT standards within the DOD Enterprise Architecture, while DODI 8270.bb, formerly establishes the DOD EA by outlining the Department's framework, regulations, and mission. Both documents JIE EA v4.0 and DOD IEA v3.0 are under review and will be utilized to merge both the direction of DOD IEA v2.0 with actual JIE EA design and strategy into one single authoritative document (DISA, 2014a). Each Reference Architecture document created describes a specific JIE global focal point while the Solution Architectures describe the services that will be implemented to support the focal point. Because the JIE is an ongoing implementation, these documents are constantly undergoing review, evaluation, and modification in order to more correctly address the implementation process. Figure 4 illustrates the various documents utilized within the JIE.

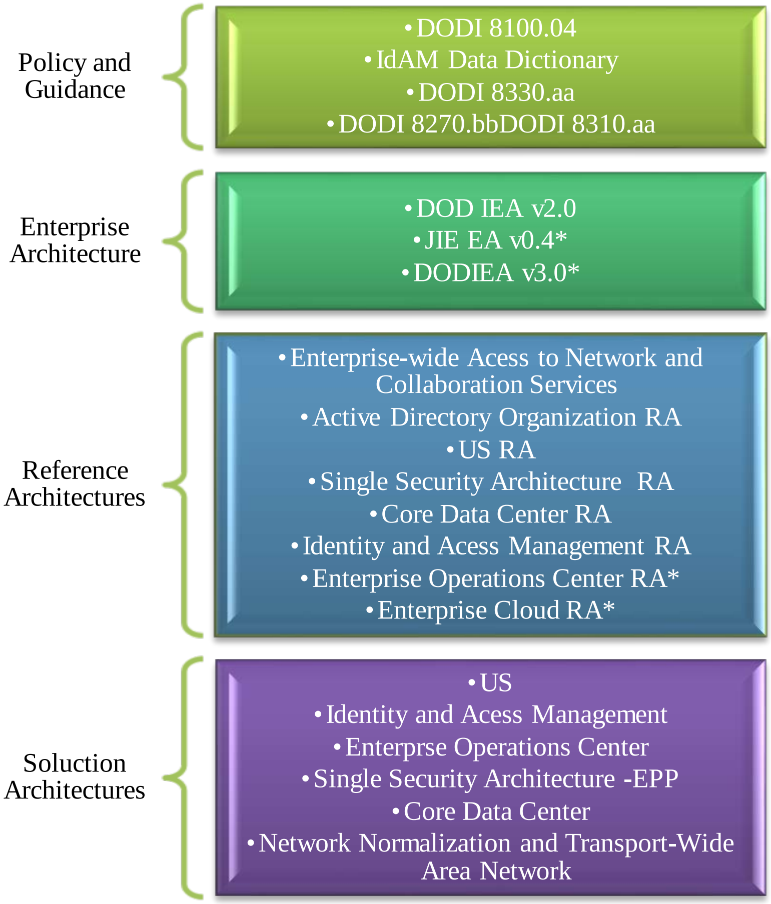

Policy and Guidance
•DODI 8100.04
•IdAM Data Dictionary
•DODI 8330.aa
•DODI 8270.bbDODI 8310.aa
Enterprise Architecture
•DOD IEA v2.0
•JIE EA v0.4*
•DODIEA v3.0*
Reference Architectures
•Enterprise-wide Acess to Network and Collaboration Services
•Active Directory Organization RA
•US RA
•Single Security Architecture RA
•Core Data Center RA
•Identity and Acess Management RA
•Enterprise Operations Center RA*
•Enterprise Cloud RA*
Soluction Architectures
•US
•Identity and Acess Management
•Enterprse Operations Center
•Single Security Architecture -EPP
•Core Data Center
•Network Normalization and Transport-Wide Area Network

**Figure 4 Key JIE Policies and Architecture Documents**

The term enterprise utilized within the GEOC is descriptive; in the future the DOD wants go away from thin clients and workstations and instead seek to implement a seamless solution that links fixed to mobile users who can access needed resources and applications securely from the cloud. For this reason, the DOD seeks to move away from simply securing their networks to actively implementing a single security architecture in which all types of mobile devices are not only approved but actively being utilized (DISA, 2014a). Although not within the context of this paper, this introduces a larger concept of a single security platform that supports mobility.

DISA has been the appointed leader of the JIE Technical Synchronization Office (JTSO) and is in charge of developing and assimilating JIE technical plans, retiring legacy systems, standardizing and coordinating wired and wireless networks, consolidating data and computing centers into a combined infrastructure, and justifying various applications and their use in various DOD programs in an effort to consolidate and standardize all while assessing the risk of transition and providing solutions to fiscal shortages (DISA, 2014c). The JTSO is actively working towards initializing the first set of Joint Regional Security Stacks (JRSS), which are software and application suites that perform various security capabilities (Jaucione, 2014). In September 2014, DISA, in concert with its Army and Air Force mission partners, implemented the first JRSS at Joint Base San Antonio. The JRSS includes many different network security functions including firewall, intrusion detection and prevention, virtual routing, and forwarding, and enterprise management in accords with the JIE (DISA, 2014d).

**Operating Systems**

The DOD utilizes several different types of OS within various DOD INs. The following is a literature review of the most common types of DOD IN including the definition of an OS, structure of the most common types of OS, strengths, and weaknesses of each type.

An operating system is the intermediary between the hardware that makes up a computer and the applications software that enables the computer to execute a specific task or function. It is composed of a set of programs that allow the hardware to complete the functions required from the application software. Operating systems complete three specific tasks: to implement and monitor the computer's main resources such as memory, CPU, and disk drives; accomplish services for application software; launch the user interface. Other tasks involve spawning processes, allocating primary memory to applications, managing data storage, creating threads, hand distributing a multi-level secure execution platform (Bassil, 2012). Although the operating system completes much of its tasks without a user's knowledge, it is the operating system that executes commands for application input and output (Fay-Wolfe, ND). Operating systems can be divided into stand-alone computers or a network operating system, with many computers that make up the backbone of a network and a server that stores data or executes hardware and software functions for the other computers (Lemley, ND).

Users continuously utilize an operating system by executing commands via applications software and the operating system constantly interacts with the user by controlling what the user can do on the computer. Herein lays the key to operating system security; although an operating system controls what the user does on the computer via restricted access to the computer's resources, a user can also elevate their access to these resources if the operating system software is vulnerable.

### Microsoft Windows.

Microsoft Windows operating systems were originally designed for a personal computer with a single user. At its creation, Windows built insecurely because the OS was built as a standalone machine and instead focused on ease of use. Windows has since gone through many OS versions that have security in mind but according to authors Marsanu et al (2010), Microsoft Windows operating systems are consistently vulnerable to attack and exploitation because they are unreliable over time and have weak security standards. Because Windows OSs tends to become insecure over time, Microsoft actively works to patch security vulnerabilities in their operating systems until they are no longer supported by Microsoft. Although Windows does acquire the largest amount of OS exploitations its users make up the largest share of the COTS OS market. Attackers may choose to exploit Windows more often because it ensures that they can get the most out of their efforts. One of Windows biggest vulnerabilities has been the use of administrator and user accounts and Microsoft has addressed this vulnerability in Windows 7 through the privileged user account. Rather than requiring a user to utilize the administrator account to accomplish tasks that require elevation, the privileged user account mitigates the use of administrator accounts in order to utilize most functions of the OS.

Windows utilizes Discretionary Access Control (DAC) grants access tokens and usage of privileges through inheritance of permissions. The owner of an object can designate user's rights to each object such as read/write/edit/delete, grant access or deny. In Windows, a user is assigned permissions based on the group that user is assigned to, and permissions are granted or denied based on that group. This is best managed through Windows Access Control List (ACL) which allows the owner to assign privileges based on groups. Windows doesn't utilize process hierarchy, which means that a created object inherits the permissions of the parent object that created it. Instead the OS treats all objects to the same generation by utilizing Job Objects, which groups different processes together under one set of rules (Microsoft, 2006). Similarly, every time a user logs onto a Windows OS, the user's ID is granted a token that signifies the user and will be utilized for the entire session. As the user accesses the session, it assigns a copy of the token (called a Session ID) to each process and thread that executes as the system is utilized and DAC enforces the permissions to all objects, which are validated by the token that has been granted to the user. While DAC represents a much easier method of granting permissions to objects based on inheritance, it doesn't provide the most secure method of ensuring objects are accessed by only those with the need to view the object.

Windows uses a HMAC-MD5 cryptographic hash to encrypt a user's password during authentication rather utilizing the plaintext version of the user's password. During authentication, the system generates a hash of the user's password, creates a 16 bit random number called a challenge, and encrypts the challenge with the user's hash with DES. The encrypted challenge (response) is sent to the server and is compared by the server who also completes the encryption steps to compare the value. A successful authentication means that the values are identical (Bassil, 2012). Windows utilizes both NTFS and Encrypting File System (EFS) to organize and store files within the file system. NTFS utilizes file compression, directory security, recoverability, and encryption through EFS. EFS utilize the Advanced Encryption Standard (AES) to encrypt files.

### UNIX.

The UNIX OS spawned from a 1960's project called Multiplexed Information and Computing Service MULTICS. AT&T researchers left the project and decided to utilize some of the MULTICS ideas while forgoing the complications of the project. It is based on utilizing small programs called tools that perform a single function. Their development of AT&T's System V and the University of Berkeley's "Berkeley Software Distribution" (BSD) quickly grew in popularity and most current versions of UNIX stem from either System V or BSD (Turnbull, ND).

The UNIX operating system was originally designed for multiple users within a shared network. This is evident through its usage of permissions, users, groups, and shared resources such as files and printers. Every user is placed in a group and all resources are maintained by a user and a group who can enable permissions on all resources within the OS environment. Because a Unix OS is essentially a networked operating environment, all resources can be accessed on the network either locally or remotely (Vanderbilt, ND). UNIX is extremely customizable because the root administrator has full rights to configure the operating system and all processes that are built into the OS. While UNIX has the potential of being extremely secure, each build depends on the root administrator, and if they don't properly configure the system it can be full of vulnerabilities exploitable by attackers all the way down to the root permissions (Marsanu et al, 2010).

The three components of a UNIX OS are the kernel, standard utility programs, and system database files. These parts interact with the security policy which grants access utilizing DAC. DAC extends to objects and when an object is given a specific permission, any object created within that object (a child object) will inherit the permissions of the parent object by default. While this can be altered within the settings of the child or parent object itself, the default settings are DAC inheritance (Shou & Shoemaker, 2007). All users are given a User ID (UID) and/or a Group ID (GID) in order to be assigned permissions to view objects. However, there are variants of UNIX who also utilize Mandatory Access Control (MAC) which provides restrictions based on security attributes on an absolute level; instead of issuing a UID to a user all users are issued GIDs which correspond to the specific security access they have been formally granted. This access may be in the form of access control labeling or through capability lists (Garfinkel & Spafford, 1996). Although the handling of groups varies between different distributions of UNIX, they either utilize DAC or MAC to control them. The most important user is root which is a superuser and has access to all passwords, accounting, and low-level system functions such as managing input/output devices. The root user enforces almost complete control over the operating system and can bypass almost all security controls (Garfinkel & Spafford, 1996). UNIX utilizes DAC to execute permissions in the file system, but it is important to note that permissions can be configured in a way that a user may be able to execute a file without being able to read it. UNIX permissions are read, write, and execute. Additionally, a user who has read but not executes access on a file can make a copy and run the file themselves (although it will have a different pathname) and will be owned by the user who copied it and not the original user. DAC allows permissions to only be changed by the file owner or root, and although ACLs are not implemented as a standard in the various UNIX distributions, most popular variants utilize ACLs. UNIX systems utilize Portable Operating System Interface (POSIX), which was developed by the IEEE and adopted as an international standard as ISO/IEC 9945. It allows different versions of UNIX to have the same common interface and can be utilized on various different machines from different vendors (Garfinkel & Spafford, 1996).

Both UNIX and Linux have similar design because Linux was designed with a similar implementation of UNIX based on read/write/execute permissions. However, it is important to note that each distribution of both UNIX and Linux varies in its security concepts and implementation. For that reason, it is not simply appropriate to choose which version is more secure. Instead, it is more appropriate to choose the right distribution of either UNIX or Linux which will meet the functionality and security needs of the user or organization (Turnbull, ND).

### Linux.

The Linux operating system was developed by Linus Torvalds and further developed by a community of software developers worldwide. It is built on the Linux kernel but consists of various software parts, utilities, system programs, and tools that make up different versions of the Linux OS known as the GNU. THE GNU is open-source software that is freely available to anyone and allows the user to both read and modify the source code. Because the source code is freely available, there have been many different distributions of Linux such as Ubuntu, Mint, Red Hat, and Fedora.

Although Linux utilizes DAC, it also enforces Mandatory Access Control (MAC). This means that an object's owner can't dictate the permissions of an object and instead the system provides access based on the information security policy as dictated by the organization. Users are granted formal clearance in order to be given access to objects (Bassil, 2012). In MAC, each object, subject, system process, or file is assigned security attributes based on polices or rules. Users are assigned a security level which describes what type of accesses they are granted on the system. Linux combines MAC and DAC by providing discretionary access within each security level (Shou & Shoemaker, 2007). Although MAC is more secure method of granting permissions and securing an OS based on objects, it is very inflexible and harder to implement than DAC.

Linux stores user information and passwords within a /etc/passwd or /etc/shadow file. If information is stored in the /etc/passwd file, they are visible to all users. Utilizing the /etc/shadow file instead ensures that only the root user can view them. While most of the information within the /etc/shadow file is stored in plaintext, the password is encrypted by the MD5 hash algorithm. Although Linux didn’t traditionally use encryption within the file system, more distributions are utilizing Extended Access Control Lists (EACL) which adds nine flag bits to each file to determine the access permissions. A file can be divided into the classes file owner, file group, and others and are used to ensure that a user can access the file only when they are authorized to do so (Bassil, 2012).

When debating the merits of Windows versus Linux and UNIX, it is apparent that each OS has vulnerabilities and flaws. Different in design, Linux and UNIX are similar in that they utilize a modular design to separate their security components into independent processes and services that work together to accomplish authentication, policy enforcement, account management, and logging events. Modulation is utilized to ensure system stability and allows the system to be updated with greater flexibility and ease. Although each OS has its own security aspects and implementation, both Windows and Linux utilize similar security mechanisms such as object and user identification, access tokens, access control lists, and file system encryption. While both Linux and UNIX employ modulation at a the kernel level, Windows doesn't quite accomplish this and many applications that can be exploited reach all the way down to the kernel (Bassil, 2012). Windows instead utilizes partitions to hide protected subsystems from users. These protected subsystems sit on top of the kernel in the user mode rather than residing in kernel mode and integrating with the kernel. Of these protected subsystems, the most important are Microsoft Win32® subsystem which controls most of the OS's operations and the security subsystem. Microsoft explains, "Partitioning of the protected subsystems and the system kernel simplifies the base operating system design and makes it possible to extend the features of an individual protected subsystem without affecting the kernel." (Microsoft, 2006, para. 13) UNIX and Linux instead utilize three operational levels: the kernel, the shell which provides the user interface, and high level tools that run utility tasks. UNIX supports multiple users simultaneously by default through the command line or through a Graphical User Interface (GUI). This occurs because once a user logs onto UNIX a shell process starts for each user and prevents the multiple processes from interfering with each other during the user's session. Multiple users can log on and utilize the same multiuser system. Windows doesn't employ this by default and instead utilizes a single user version such as Windows 7 or a multiple user version such as Windows Server 2008. Windows Server versions support multiple users by utilizing Terminal Services or Citrix but Microsoft explains that is rare for the Windows Server system to run multiple command-line users (Microsoft, 2006).

It has been noted in the past that Windows succumbs to more viruses and worms because it has about 90% of the personal computer market. However, by design, Windows may suffer more infections because of the way the OS treats redirect executables and automatic execution. If the system receives a file that attempts to open in an unrelated task, it redirects the file in order to be properly executed. This means that a Trojan posing as a Word document can be redirected to open properly in another format automatically. Additionally, Windows enables automatic execution by default so once removable media is inserted to the computer, it automatically executes without the user’s permissions. In contrast, UNIX and Linux do not utilize redirect executables and automatic execution by default. Instead, a file is opened in the format that it was received, so the system will fail to open the file properly if it is stored in an incorrect format (Perrin, 2010). However, UNIX and Linux suffer from vulnerabilities because it is up to the administrator to properly configure the system and implement security controls. Because UNIX and Linux implement security by allocating permissions on files, they may inadvertently leave their system wide open to intrusion. This happens because UNIX utilizes files to represent and organize virtually everything on the OS including devices, processes, memory, and passwords. Windows instead secures users and objects by storing information in Active Directory (Microsoft, 2006).

## Open Source Software

It is a common misconception that the DOD only utilizes COTS or GOTS software and operating systems. The DOD utilizes Open Source Software (OSS) IAW guidance to clarify when to approve public domain software products. Specifically, this guidance utilizes IA vulnerability analysis to review open source software and may disallow use if there is no way to access the source code for repairs and mitigation or there is no other author that can implement patches. If the government can’t view the source code to repair the code, they may not utilize the software unless it is a specific mission requirement (Chief Information Officer, ND).

Another initiative introduced by the DOD in 1994 is the Open Systems Architecture (OSA), formerly known as the Modular Open System Approach (MOSA). Its purpose is to blend both business and technical strategies by ensuring that only open systems are utilized within the creation and implementation of all new and existing technologies within the DOD. This ensures that these technologies can be created more affordably and with a greater chance of integration amongst other technologies. Additionally, because open systems are widely supported and utilized by other industries, they have been successfully validated, tested, and are supported in greater ways than the DOD can fund. Open systems usually are created and managed by groups and consortiums that maintain the standard and are widely adopted by other industries, thus greatly extending the possibility of its use in many different applications and software (ODASD, 2015).

The DOD analyzes open system requirements by examining these characteristics:

- The design of the system to determine if it is an open standard or protocol.
- Does the system utilize derived interfaces and does this system enable interoperability?
- Are there design constraints that prevent the system's components maintain an open interface?
- Determine the architectural attributes that allow the system to adapt, be upgraded, or reorganized.
- What are the benefits and concerns of the design itself?
- How does the system affect business strategies in order to obtain supplies needed for the system and to manage aging technologies?

DOD Instruction 5000.02 "Operation of the Defense Acquisition System" discusses OSA, which should be considered within an Acquisition Strategy or even the Systems Engineering Plan. To further ensure the use of open systems during the acquisition phase of a program, the program manager should include a summary of how the specified open system will fit into the entire program and how the open system affects program development. The summary should also discuss the technology impact that the open system will have on the program as well as the proposed adaptation and implementation of the open system within the program as well as how the program will monitor the usage of the open system to further ensure its commitment towards openness (ODASD, 2015).

The development of open systems within DOD technologies and systems represents a both an opportunity and a challenge to the DOD. On one hand, the DOD will greatly benefit from utilizing proven technologies in a greater fashion than it already has. For example, the DOD currently uses many open system protocols that are managed and maintained by other groups dedicated to the openness and security of the protocol. Some examples of these are the TCP/IP model, POP3, HTTP, and SSL. Each of these protocols is utilized regularly within the DOD as well as all other industries. While it doesn’t seem out of the ordinary for the DOD to utilize these protocols specifically, each has been created by non-military programs for general use. The usage of these and other open system protocols have their disadvantages, however. Although some open systems are continuously maintained, they still manifest security vulnerabilities that can be utilized by attackers. So even though these and other protocols are imbedded within DOD technologies, the DOD must continue to evaluate the security of the open system as well as its interoperability with other technologies. Additionally, the DOD must continually mitigate vulnerabilities of these open systems as well as new vulnerabilities that manifest by utilizing open system protocols together.

In the past, the DOD has utilized open system software in order to create proprietary technologies for use in specific programs. At first glance one may believe that the DOD has created a proprietary product, but deeper investigation shows that all DOD programs utilize some type of open source software or open system protocols as a platform to create purpose-driven applications and software. In this way, the DOD has always utilized the OSA initiative, although the primary purpose was to create a more cost-effective and secure solution rather than to ensure the interoperability of the software with other DOD programs. The author presents a discussion of DOD proprietary operating systems both past and present. These operating systems have been chosen both for their usage of open source and open systems technology as well as their focus on implementing security from the OS kernel up to high level software and applications. However, in order to understand the DOD's history of OS creation, it is also necessary to discuss the agency that is tasked with creating new technologies for use within the DOD.

## DARPA

The DOD has had experience in operating system development through the research of The Defense Advanced Research Projects Agency (DARPA) which is the leading DOD agency who creates technologies for use within the DOD. For 50 years, DARPA's mission is to, "foster advanced technologies and systems that create revolutionary advantages for the U.S. military". (Van Atta, ND, pg. 20). DARPA was founded as the Advanced Research Projects Agency (ARPA) in 1958 in response to the launch of Sputnik by the Soviet Union, who was the first nation to enter space. Because the Russians were able to reach space before the US, this agency was founded with the idea of pursuing advanced research and development in a way that traditional U.S. military research couldn't. Unlike military Research & Development (R&D) departments, ARPA was expected to examine ideas, challenge existing methods, and create farreaching products without the constraints of incremental changes. ARPA itself calls this, "high risk – high payoff" (Van Atta, ND, pg. 20). Former director Dr. Eberhardt Rechtin described ARPA as more of an "advanced" research agency rather than a development agency, because research implies much greater risk than simply development. If more emphasis is put on development of a product or idea, that means there is less risk and it is a less forward-thinking idea (Van Atta, ND, pg. 21). ARPA was created to be a flexible, adaptive, and lean organization with no military hierarchy, protocols and with the essential mission of creating advanced military technologies in both offense and defense. To be truly advanced, ARPA has never implemented research on a specific requirement that has been identified by the U.S. military or was requested to meet a need. Among the famous ARPANET which was the precursor to the Internet, ARPA is also the inventor of ground-based phased array radar, high-energy lasers, ballistic missile penetration and defense, and sensing technologies- all before 1965. Like all of these technologies, ARPA's role is the same- to create ground-breaking technology and pass this onto a defense contractor, military R&D lab, or another responsible party to further develop the technology and eventually put it into production. This is best explained through the creation of NASA, which was spawned one year from the founding of ARPA. NASA literally took half of ARPA's personnel with them which allowed ARPA to grow again and begin working on other projects (Van Atta, ND).

Although DARPA has a significant role within creation of DOD technologies, they are not tasked with the actual implementation of much of their research. Instead, they usually create an idea, technology, or some prototype, and then pass on their ideas to another agency, federal contractor, or academic affiliate in order to develop the idea or research into a product that fulfills the purpose for its creation. In the past, the DOD received large amounts of support from academic institutions, which they in turn financially supported specific project research teams. A discussion of various DOD proprietary operating systems therefore invariably has three common points: the usage of both federal contractors and academic intuitions within their research, the usage of open source and open systems, and the desire to create a truly secure operating system.

## DOD Proprietary Operating Systems

### Multics.

The DOD has historically sought to find more secure methods of securing its networks and information systems. To do this, the DOD implemented several projects to create a truly secure OS because computers were very expensive and they needed to be shared amongst several people. Specifically, the DOD was interested in utilizing one machine for different classifications in order to prevent spillage of data from one classification level to another (Perrine, 2002). During 1965 the Multics project was implemented by a combined effort between AT&T, Honeywell, General Electric, MIT, and was funded by DARPA. Multics stands for Multiplexed Information and Computing Service and was a mainframe timesharing operating system that provided secure computing sessions to remote terminal users. Multics ran on specialized CPU hardware that provided segmented, paged, ring-structured virtual memory and utilized a supervisor program that managed all hardware resources. It supported symmetric multiprocessing, multiprogramming, and paging long before more modern OSs and the overall design was created with security as the fundamental design requirement. ARPANET utilized Multics to develop applications in Project MAC, Project Cambridge and Project Guardian, which influenced the DOD's Orange Book creation after the Air Force's experience in building secure operating systems. In the 1980's the DOD purchased Multics systems and created the Naval

War Games System as well as installed a system within the NSA. When tested in the 1980's Multics received a B2 rating in the Orange Book. After becoming a commercial product of Honeywell, it was sold to academic institutions, commercial industries, and utilized in government projects until 2000 when support for Multics ended (Multics, 2015).

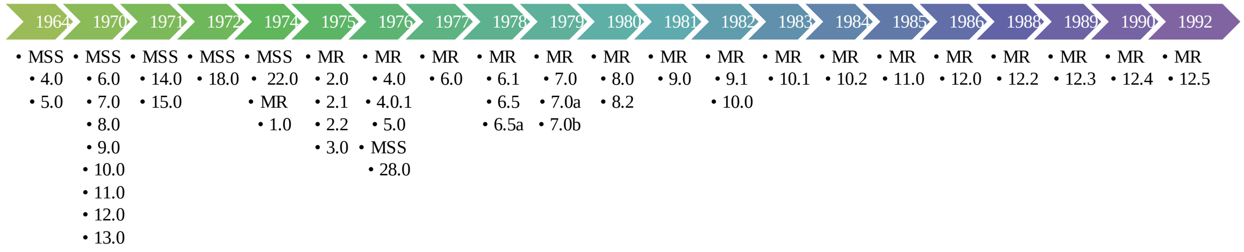


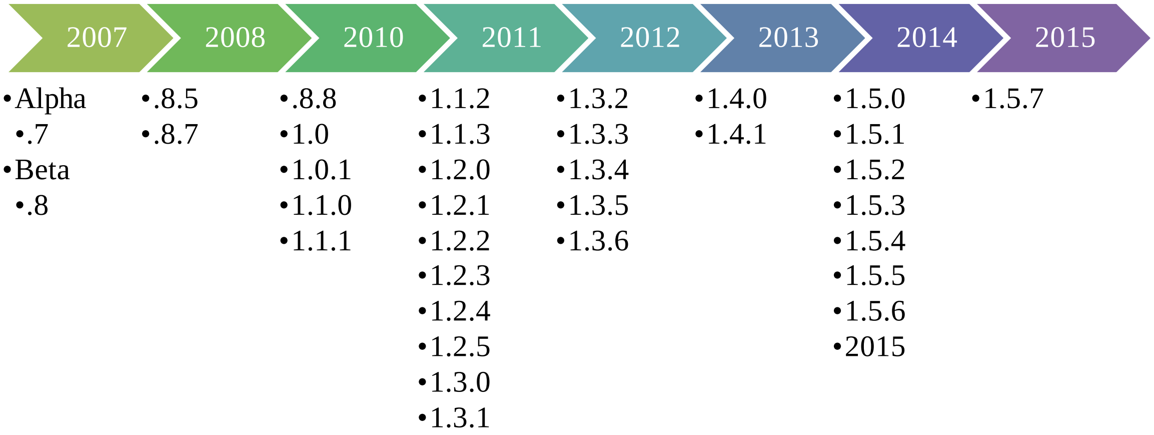
**Figure 5 Timeline of Multics Operating System Versions**

### KSOS.

In 1978, the DOD introduced the Kernelized Secure Operating System (KSOS) which was thought to be the most secure kernel operating system to have been created specifically for the DOD. Its creation originated from DARPA but was transferred and completed by Ford Aerospace & Communications Corporation and SRI International (NA, 1978). The OS was created utilizing the concept of multilevel security, which is the process of attaching a tag (also known as an access level) to every object that is maintained on the system which places limits on the relationships between access levels and the objects within the system. In this way, objects are restricted from interacting with each other depending on their access level. Multilevel security represents a much greater security posture than DAC or MAC but is much more complex to implement and maintain. KSOS was comprised of three components; the kernel, an emulator which assists the kernel in creating an environment in which the user programs can function, and support software which are called N-Kernel System Software (NKSS) (Berson & Barksdale, 1979). KSOS's emulator was initially based on UNIX and the overall OS was created with the intention of being compatible with UNIX (NA, 1978). NKSS sometimes had to violate its own security model in order to allow performance or ensure the practical use of the OS. The developers attempted to remedy this by dubbing these specific trusted processes privileged NKSS and limited their usage by instead creating finely grained user privileges and tools to assist kernel security support by defining user extended types (Berson & Barksdale, 1979). KSOS utilized the SRI International's computational Hierarchical Development Methodology (HDM) which provided streamlining of an ordered series of steps that are required to design and produce a system. By integrating a collection of support programs into a design of decomposed modules in a hierarchy fashion, the specifications, mappings, and interfaces could work together as an incremental abstract machine and eliminated the need to manually compute the generation of formulas for the design of KSOS. This means that each module has the ability to be constructed with no regard to the types of data it will be utilized for, including the types of algorithms. It also utilized a large-scale of automatic theorem generation and proof that was previously only utilized in academic environments. KSOS utilized Modula-2 programming language and was tested within the UNIX Programmer's Workbench. Author Tom Perrine (2002, pg. 39) considers KSOS to be an A1 level rating within the Orange Book because each of its 32 kernel specifications was separately verified to contain no explicit violations. The Orange Book's A1 rating is considered the highest rating and is called "verified design". KSOS was eventually adopted for several Air Force and Navy OSs, but the KSOS project was eventually cancelled along with several other computer security projects in 1988 (Perrine, 2002).

**LPS.**

In continuance with DOD intent to develop secure operating systems, the Air Force Research Laboratory's Software Protection Initiative created the Lightweight Portable Security (LPS), a portable thin Linux operating system that can be booted from a CD or USB flash stick. LPS can be booted on almost any Intel-based PC or Mac and doesn't require the OS to be mounted on the local hard drive. Because LPS is designed to run from read-only memory, it doesn't utilize any persistent storage and any vulnerabilities or malware that are introduced within the active session can only progress within that session. LPS actually consists of three different types of OS shells; LPS-Public, LPS-Public Deluxe, and LPS-Remote Access. Both LPS-Public and LPS-Public Deluxe are approved for public use in order to access web applications over the Internet in a secure manner. Each is essentially the same build but LPS-Public Deluxe provides Adobe Reader and LibreOffice support. Of all three, LPS-Remote Access was approved by the AFNIC to be connected to the DOD Global Information Grid (GIG) in 2011. The purpose of LPS is to provide the user with a secure method of utilizing web-based applications and to connect to remote networks. It utilizes remote desktop software, Secure Shell Hash (SSH) client, Firefox browser and supports the use of Common Access Card (CAC) and PIV cards, Adobe Flash, Java, file browser, Encryption Wizard-Public, and Microsoft Office-compatible software called LibreOffice. The Alpha version was created in 2007 and continues to be updated, with the newest version being Version 1.5.6 released on December 5, 2014 (ATSPI, ND). The intent of LPS is providing security on mobile devices or devices that need to connect to the GIG remotely. This reflects the DOD's objective towards mobile security.



| 2007 | 2008 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|
| •Alpha<br>•.7<br>•Beta<br>•.8 | •.8.5<br>•.8.7 | •.8.8<br>•1.0<br>•1.0.1<br>•1.1.0<br>•1.1.1 | •1.1.2<br>•1.1.3<br>•1.2.0<br>•1.2.1<br>•1.2.2<br>•1.2.3<br>•1.2.4<br>•1.2.5<br>•1.3.0<br>•1.3.1 | •1.3.2<br>•1.3.3<br>•1.3.4<br>•1.3.5<br>•1.3.6 | •1.4.0<br>•1.4.1 | •1.5.0<br>•1.5.1<br>•1.5.2<br>•1.5.3<br>•1.5.4<br>•1.5.5<br>•1.5.6<br>•2015 | •1.5.7 |

**Figure 6 Timeline of LPS OS Versions**

The collaboration of federal, private sector, and academic research has always been utilized to develop information system security standards and guidelines that can be applied in all sectors. There are many reasons for this including the ability to further research and technologies, reduce the cost of creation, development and testing, and reduce duplication of work. This also ensures that all sectors continue to improve the quality of their security posture and promote information sharing in regards to new technologies and concepts (NIST, 2013). Although much of this research is utilized to secure the DOD GIG, all of these OS programs were adapted for public use in order to further the security posture of the U.S. commercial market.

### Capsicum.

Today, DARPA is continuing to create new technologies, including research on operating systems that would prove to be more secure than any COTS product. The Clean-slate design of Resilient, Adaptive, Secure Hosts (CRASH) is a DARPA program that is tasked with creating or modifying computer systems that are more resilient to internet-based attacks by adapting to an attack, learning from previous attacks in order to prevent future attacks, and repairing itself if necessary after an attack. CRASH research includes operating systems, programming languages, processes and instructions sets, and application tools (DARPA, 2014).

Capsicum is a FreeBSD operating system utilizing the Linux kernel and allows compartmentalization and decomposition of larger applications and web services (DARPA, 2014). Capsicum is a joint project by the University of Oxford, DARPA, and Google (Watson, 2010). This platform is based on FreeBSD and UNIX kernel, but is considered much more secure because it allows each application and add-in component to be sandboxed within the Information System (IS), preventing the spread of any vulnerabilities or errors to another part of the IS (DARPA, 2014). Capsicum is an adaptation of the Portable Operating System Interface (POSIX) Application Program Interface (API) which is standards that are implemented by the IEEE in order to promote compatibility between operating systems between UNIX, Linux, and other Operating Systems. This allows the interoperability of command line shells, utilities, and software between various OS. Although DARPA is sponsoring Capsicum, it has commissioned other labs to develop the OS's capabilities. One lab working on this POSIX API is the University of Cambridge Capsicum for UNIX. The University of Cambridge model is different because it is more interested in promoting security though sandboxing of applications and allowing the interoperability of various security applications that have their own models of security. By utilizing FreeBSD, the researchers have an existing application stack to work with in their research and are creating a hybrid system as an experiment but with the expectation that it can be applied to other types of OSs and even proprietary OSs. Additionally, the University of Cambridge is trying to address the problem of mapping from the Internet's distributed security domain to the local security policy domain of a network.

The Capsicum model focuses on capabilities in Capability Mode rather than the Mandatory Access Control model. A capability is a token of authority structured around delegation, and no one can access a specific object unless it is delegated (Watson, 2010). Capabilities are utilized to evolve policy on demand in order to reflect user and administrator's need because policy is built into the actual application itself. In general, a network's security policy deals with mapping a local security domain and not mapping a distributed domain that may encompass a large network that spans the Internet. One way to map a distributed domain is to utilize sandboxing within each application or group of components. Sandboxing is a term utilized in computer security to describe the separation of programs within a separate environment to ensure that any security vulnerabilities or errors don't spread to any other part of a system (Techopedia, ND). Sandboxes can be done by testing questionable code with a proof code in which the code will change if there are errors or exploits that alter the code in the application. In this way, sandboxing also supports the use of capabilities which will allow the security policy to be enforced within each application rather than generically based on roles, discrepancy, or mandatory access. Sandboxes create sandboxes in the web interface and extensions, such as sandboxed software apps like JavaScript, email browsers and Flash (Watson, 2010).

Although Capsicum is still undergoing research, it has already proved that that a more secure operating system is capable of learning from attacks in order to repair itself and prevent future attacks through using capabilities. Additionally, the US government has already created extremely secure proprietary OSs in the past utilizing multilevel security and other concepts that have been introduced in the research of today's new OSs.

# Research Design

## Introduction

The purpose of this study is to research the information security implications of implementing a proprietary OS within DOD networks. This section will discuss the research design utilized to adequately analyze the research questions and hypotheses proposed by the author. Other areas to be discussed will be the research approach, identification and operationalization of variables, and a discussion of biases and limitations within the research design.

## Research Questions

The research design of this study has been structured to address four research questions that were proposed in the introduction.

- Who is responsible for creating and implementing DOD Information Assurance (IA) policies and procedures?
- What is the current DOD OS platform implementation?
- What is a possible OS replacement?
- How can this OS platform replacement provide an enhanced information security posture in comparison to the current OS platform?

These research questions were chosen to address the problems identified within the problem statement. The DOD is unable to fully implement and manage a secure OS platform on DOD networks because they don’t own the source code, system software, or kernel that would allow them to fully configure, alter, upgrade, harden, or test the OS platform on which their entire network utilizes. Additionally, the DOD utilizes many different variants of OS configuration on each of its INs. This makes OS security standardization not only difficult but almost impossible.

## Hypotheses

To address these research questions, the following hypotheses have been identified.

- The current OS stack can be replaced by a more customizable solution.
- All applications can still be utilized on the alternative platform with little to no downtime.
- An alternative OS can provide a greater secure security posture than the current Microsoft Windows configuration.

## Research Approach

This study utilizes the exploratory design because there is little to no current research found on replacing the current DOD OS stack with a proprietary system stack and this paper serves to spawn further research into the possibility of implementing a proprietary OS platform stack to further advance the DOD IA posture.  In order to explore the issue of DOD OSs, the author became familiarized with the process of evaluating OS platforms based on the current DOD standards as well as understand the security controls and laws that regulate the DOD's network implementation. The exploratory design allows for the discussion of these concepts as well as clarification any unfamiliar terms or technologies to the reader.

The author's methodology utilizes analytical and qualitative analysis methods. Analytical research methodology is utilized to condense large amounts of research and organize their findings.  Most research has centered on policies, processes, security controls implemented by the DOD as well as an analysis of who does what within the DOD.  However, the author also seeks to discuss the utilization of various OS platform implementations within the DOD as well as the DOD’s prior experience in utilizing this type of technology. This research also describes the types of security controls that are utilized such as a STIG and relates any challenges to implementing a STIG to multiple INs (DISA, 2014). Academic literature, organizational records, academic journals, news outlets, and discussions with IT professionals have been utilized as resources within this work. Qualitative analysis is also utilized to review, analyze, and deduce conclusions based on the archival research presented within the literature review and provides an evaluation between the OSs that is described in this thesis.

The author has identified a few reasons that the actual testing of these OS will not be feasible.

- The author doesn’t have access to the Capsicum or other OSs as well as their configuration.
- The author is unable to implement the STIGs within a testing environment on either system.
- The author doesn’t have access to the JITC network or resources that are utilized for testing purposes.
- The author doesn’t know the network configuration utilized to conduct tests.
- STIGs have already been developed for Windows OS platforms, yet not created for Capsicum or LPS.

Because of these challenges, the author can only research how an OS is tested, approved, and how STIGs are created. A separate yet interrelated challenge is the additional requirement of UC testing, approval, and certification. Similar yet unrelated to STIGs, JITC’s method of UC testing and certification in accordance to DODI 8100.04 is also conducted under conditions that can’t be replicated by the author.

## Identification and Operationalization of Variables

Two variables have been identified within the research of this paper. The first variable is the uniformity of the STIG template. Currently, the DOD utilizes STIGs created by DISA to approve or deny hardware, software, applications, and firmware on a DOD network (DISA, ND). DISA also pushes out patches, hot fixes, and bulletins that direct further security controls that should be utilized on the network. Because DISA utilizes basic templates to create STIGs, each DOD network should have the same configuration, yet they don't. Under the current GIG structure, it is up to various combatant commands and DOD agencies as to how their network is configured, how STIGs are applied, the timeline in which they are applied, and the method of application. This occurs because each DOD branch and agency has the ability to create information security policies and procedures that implement the configuration of their network. Although they must abide by the law and DOD instructions, they can create actual procedures on their own level. This means that each network is different and has the choice of applying the STIG settings differently and with different tools (Morrison, 2013). While JIE UC policies and instructions seek to eliminate the redundancies and incompatibilities that exist within the GIG, the DOD has yet to fully implement UC approved products within their entire network. Therefore, inconsistencies continue to prevent uniformity within DOD INs themselves.

Another variable identified within this paper is the usage of government off the shelf (GOTS), commercial off the shelf (COTS), and open source software (OSS). The government has authorized the usage of each of these types of software. It is a misconception that the government doesn't have the ability to create and manage their own software as well as use "free" software. In contrast, they utilize all these types on their network. This means that the government already has the resources and ability to create and utilize both GOTS and OSS software in order to develop a proprietary OS (Hissam, 1998). However, there is a big difference between the two. GOTS software is developed and maintained specifically for the government by a software development company who has been contracted by the US government. OSS software, in contrast, is freely available for use according to the GNU Free Software Definition and is also approved for use as long as it meets the DOD requirements set forth in “Clarifying Guidance Regarding Open Source Software (OSS)” (CIO, ND). Although GOTS and OSS software are different, the government would have the source code for each in order to alter or patch as they need to. By using COTS software, they wouldn’t have the source code and instead rely on patch management by the third party developers.

## Biases and Limitations of Research Design

On a perfect network, all DISA STIGs can be implemented on the DOD IN and everything will work as it is supposed to. However, many times STIGS can create intolerabilities within a network by preventing applications from working as they are supposed to (Lam, & Carayannis, 2011). This happens because the DOD has many different types of networks that utilize vastly different types of network, hardware, software, and application configurations, some of which still employ legacy software. In this case, the network administration team may need to request a waiver to implement the STIG(s). The DOD seeks to utilize the JIE as well as the Federal Data Center Consolidation Initiative to reduce intolerabilities within any DOD IN, yet there has been no conclusive proof that this has been accomplished as the program is still in progress. Because the author has experienced conflicts during software patching testing and implementation within her own networks at work, she has some bias as to how the STIG process actually works when implemented on the network. In order to combat this bias, the author has refrained from asserting her opinions on DOD IN implementation and instead discussed the findings of academic research and interviews of other IT professionals.

The author will not be able to adequately compare the OS platforms discussed within this thesis because only common OS platforms such as UNIX and Microsoft Windows 8 have been tested and approved for DOD usage. Capsicum is still in a development phase so the author can't recommend the replacement of a common OS platform to Capsicum. The author can only research the proposed security design and characteristics of the Capsicum OS platform if implemented. The author recommends that another study should be conducted to fully address this gap in knowledge. Because STIGs are large working documents, this thesis can only cover a STIG in a broad level. However, quantification analysis can't be utilized because the author doesn't have necessary time or access to resources to fully implement quantitative analysis.

## RESULTS

This thesis centers on the following three research questions:

- Who is responsible for creating and implementing DOD Information Assurance (IA) policies and procedures
- What is the current DOD OS platform implementation
- What is a possible OS replacement?

In the past The DOD hasn't actively sought to create a proprietary OS for common use, but instead created several different OS platforms for use of within specific programs and for use with specific hardware. This occurs because often the DOD seeks to implement an application or software program in response to a specific need, such as a OS platform to be utilized with a specific weapons system, a piece of software to be utilized within the testing environment for a missile guidance system, or an application that allows a service member to securely connect to an internal DOD network. The author believes this may be the case because separate DOD branches and agencies have different leadership, budgets, mission requirements, hardware and software requirements, and vendors to obtain both software and hardware, or employ different contract labor force to support their networks. Because different programs require different OS platforms in order to provide compliance to their program's software requirements, DISA has approved various OS platforms for use in U.S. government networks such as Windows, UNIX, Linux, and Mac OSs. This reflects all of these types of OS platforms are being utilized in the DOD network environment. However, the implementation of various OS platforms doesn't necessarily require that each military branch and DOD agency must utilize a different OS configuration. Instead, this presents an opportunity for DISA to implement one secure OS platform stack that can be utilized within every DOD network. In this scenario, DISA approves only one OS platform configuration for Windows, UNIX, Linux, and Mac to be utilized on all networks and to support all the different types of applications, software, and network configurations. The DOD is currently implementing the Joint Information Environment which allows the sharing of information between all DOD branches via shared IT infrastructure, single security architecture, and enterprise services (Edwards, 2013). DISA is actively working towards implementing the JIE by initializing the first set of Joint Regional Security Stacks (JRSS), which are software and application suites that perform various security capabilities (Jaucione, 2014). In the past, each base, station, post, or camp utilized their own security stack. This prevented the JIE from going forward because each system utilized different security application and functions. By implementing a common OS platform amongst all DOD branches, this greatly simplifies the DISA JRSS requirements and increases the DOD's posture towards enabling the JIE. In the past, this would not have been possible because each network utilized its own security stack. However, the JIE presents the best opportunity for complete interoperability between all the DOD. This should not stop at the usage of regional security stacks and the GEOC, but should also extend to the usage of a common OS platform stack. Additionally, with developments into secure OSs that utilize sandboxing such as Capsicum, the DOD can expect to eventually move to a single OS platform for use in all programs and with all systems. This can be possible because the use of sandboxing will allow specific applications to run without interference or incompatibilities from other software. Additionally, even legacy software can be run because the sandboxed environment means that any vulnerability present in that software can't be extended to other software. Additionally, the software will have no ability to reach towards the kernel.

The DOD currently supports the initialization of OSA requirements to utilize both open source and open systems from various vendors in order to promote interoperability and cost effectiveness. The use of proprietary OS platforms continues to promote the usage of common open system protocols and also promotes interoperability of the JIE. Implementation of an OS platform stack also promotes the centralized testing, certification, and implementation of an OS platform to reduce the possibility of intolerances. This would continue to fall within the responsibility of DISA and the JITC, yet the impact of the JITC will be much greater in that it will be responsible for ensuring the security of the entire DOD GEOC and as a result, all regional security stacks and local branches. The implications are clear: the DOD will be able to provide a great deal more support towards testing and securing the entire DOD network because they have much greater resources to dedicate towards standard OS platform stack rather than testing many different OS platform flavors or security configurations. It is clear that this is not a change that can happen overnight because there are many different proprietary and legacy software and applications that are being utilized in various INs today. However, the UC APL is also working to reduce incompatibilities within software. This supports the usage of a proprietary OS platform stack because once software is placed on the UC APL, it should also be fully compliant within the OS platform stack.

The author believes that it is possible to find a proprietary OS replacement stack that is more secure than the current standard being utilized today. However, the notion of a proprietary OS doesn’t necessarily mean that a platform is completely designed, implemented, and maintained from the ground up composed of simply GOTS products and software. As discussed within the literature review, COTS is actually composed of OSS as long as long as it falls within the guidelines of OSS IAW guidance. Rather, a proprietary government OS platform describes a configuration of OSS, COTS, or GOTS software that has been carefully crafted to create an OS platform that serves as an interoperable solution as well as ensuring the highest IA implementation possible on the JIE. For example, the Air Force Research Laboratory has implemented LPS which allows the user to securely connect to Internet via Mozilla Firefox web browser and the LibreOffice to work with Microsoft Office documents. Although LPS is a proprietary OS created by the Air Force, it utilizes a Linux kernel as the OS platform on which the proprietary application was built. If the Linux OS platform was standardized to be utilized as the one Linux OS platform for all DOD use, everyone could utilize this technology. Additionally, this application could be utilized for both mobile and virtualized environments within all DOD branches and agencies instead of just the Air Force.

There are a few reasons why the use of multiple OS platforms is not ideal. Usually each OS platform utilizes vertical data structures to reduce a user’s ability to share data with users, commands, or branches. This is usually caused by different indexing configurations as used by each OS platform.  As discussed within the Literature Review, each major OS platform has been created differently utilizing a different model.  OS platform and application incompatibilities create compatibility problems when data is shared from one DOD branch/agency to another. Because many DOD applications utilize a vertical data structure and are organized in a way that can't be searched or applied to other applications, it is a regular issue to encounter application constraints, format incompatibilities, or lack of interoperability between different branch/command/region's applications.  Software packages may not work on some networks due to hardware requirements and/or whether the supporting applications have been approved, are working correctly, and have been patched.  In this way, the DOD is only utilizing its networks and data in a limited way because its users may know the data exists but is unable to access it.  Implementing a standard OS platform is therefore a viable solution to enabling application and data compatibility within DOD branches and agencies.

The benefits of this solution are many: a greater standardization of OS platforms ensures conformance to DOD 8330.01 and 8100.04 UC and the JIE; the cost of maintaining the OS platform will be much lower as this can be done centrally via the GEOC; all software utilized on the JIE will be compliant both with IA and compatibility (non-compatibility can be tested and resolved on a central scale); this simplifies the creation of policy, instructions, and Standard Operating Procedures on a local level because all local commands can exchange these documents as they are created or updated.

DISA's evolution of the STIG process ensures that technologies and products are approved for DOD IN use by giving a vendor the IA and compatibility requirements before a product is even created. This ensures that a vendor can create the product, write a STIG in compliance with DISA's requirements, and have the product approved in much less time than in the past because DISA is no longer required to “retrofit” a product that is already on the market to the DOD’s IA specifications. The benefits this are also numerous: the DOD will be able to introduce current or emerging technologies and products into the DOD IN (and later the JIE); vendors can benefit from less work to “retrofit” a product to the DOD’s specifications and more opportunity to create new solutions to the DOD’s specifications; vendors can create new products and new contracts with the government based on their emerging technologies, ensuring more work as that technology progresses; because government technologies often trickle down into commercial products, the US can enjoy greater technology innovation within the US economy. The U.S. government has a long history of creating innovative technology and the creation of a propriety OS platform reflects the natural progression of technology within the DOD.

## DISCUSSION

### Summary

Currently, the DOD has so many different policies and instructions regarding the DOD’s OS implementation process including the acquisition, research, testing, approval, implementation, and maintenance of DOD networks that the author has spent a great deal of time to understand how the process works. The DOD realizes that there are numerous problems not only with the lack of interoperability within different branches, but also different regions and even programs within the same branch or physical location. The JIE is a forward-looking policy that is intended to allow the interoperability of DOD networks so that it is much easier to exchange data between the various DOD commands. However, this policy can be utilized to do more than that- it will also hopefully clear up the muddy chain of authorization and implementation to hardware, software, and application development and implementation in the DOD. It isn’t uncommon for the US government to utilize one new piece of legislation in order to neaten up the untidy remains of several incompetent or competing policies, laws, and regulations. New developments regarding the JIE will greatly simplify the process of researching, testing, implementing, and monitoring a proposed technology, and in turn will ensure that the DOD can utilize a new technology faster than before. The literature review explains several unambiguous policies, instructions, and regulations that require the DOD to implement a comprehensive DOD Information Security Program. However, each implementation of a DOD system is different based on the requirements of the commands that the network supports. While the software and applications are sure to be different, the OS platform on which the software is built can and must be standardized. OS platform hardening is an information security standard and ensures that only the most necessary services and applications are utilized. In this same sense, the DOD must implement a common OS platform solution on which applications can be utilized.

There are some downsides to simply implementing an across the board solution to creating a proprietary OS platform stack to be utilized in the DOD. First, the US most likely doesn’t have the necessary amount of DOD General Service and federal government contractor employees to begin working on any of these platforms for a few reasons: the US lacks the amount of professionals in the IT fields; the US government is unable to pay competitive wages to IT professionals as compared to the civilian job markets; as a whole US workers don’t require the knowledge possessed to develop, initialize, and maintain proprietary systems, because most of the US workforce is familiar with Microsoft products only. Second, the current configuration of the US DOD prevents the ability to simply create a model or several models of OS platforms that can be implemented. Simply put, there are too many chefs in the kitchen, or too many commands that have authority to amend, change, or veto the policies and instructions of the OMB, DISA, and any other regulatory agency that passes down policies and instructions. The third downside is that there is no regulatory agency that is capable of implementing regular inventories, audits, and reviews of all DOD systems in order to assess each system's compliance to the mandated standards. When audits and reviews aren't mandatory or regular, it is quite easy for a DOD network to become non-compliant, and it is certainly easier to become non-compliant than correcting these issues after they have occurred.

These issues present a serious complication to creating a proprietary OS platform stack. However, one particular problem is not funds to create a test product because the DOD has already created more than one possible candidate that can be tested for implementation in the various DOD networks. Additionally, implementation of the JIE may eventually provide the catalyst for the US government to create a simplified structure of its administration, and operations because it will be easier to administrate and audit the DOD network as it evolves.

## Conclusion

Through this research, the author has learned that the DOD has previously developed various secure OS platform solutions that were developed and implemented by different military branches and put into production for proprietary use but only to be dropped once funding had been cut. Rather than throwing good software away once a project ends, it is more cost effective to recycle and reuse DOD proprietary OS platform implementations in order to create a standard OS stack. With all good ideas comes a great deal of development, testing, and modifying before production, but the DOD has both the resources, networks, and platform models to do this. In the same way that DARPA utilizes many different laboratories to develop its products, the DOD should utilize its resources to develop the more secure standard OS platform stack and even if it means that their resources must share its ideas and products.

## Recommendations

By implementing a common OS platform stack that can be managed from within the GEOC, the DOD can dedicate more time on developing interoperable applications and software. Additionally, advances in OS platform design utilize software sandboxing in order to decrease the vulnerabilities present in software integration from the kernel level up. Sandboxing can also reduce software incompatibilities between proprietary and legacy software. Rather than test and implement patches on bloated OS platforms that contain too many extras to be considered hardened, the DOD can instead monitor standard OS platform stacks and ensure that every DOD network within the JIE is compliant. A standardized OS platform stack is the logical solution to implementing the JIE within the DOD. This thesis has been designed to foster discussion and create interest in developing further research into creating a standard information system security solution from the kernel up.

## LIST OF REFERENCES

Ackerman, R. K. (2005). “Defense Knowledge Management Hinges on Compatibility”. *Signal, 59*(9), 23-24,26-27. Retrieved from http://search.proquest.com/docview/216175856?accountid=8289.

ATSPI. (ND). “Lightweight Portable Security”. *Air Force Research Laboratory*. Retrieved from http://spi.DOD.mil/lipose.htm.

Bassil, Y. (2012). “Windows and Linux Operating Systems from a Security Perspective”. *Journal of Global Research in Computer Science*. Retrieved from http://www.jgrcs.info/index.php/jgrcs/article/view/305/259.

Bell, David. (ND). “Looking Back: Addendum”. Retrieved from http://selfless-security.offthisweek.com/papers/addendum.php.

Berrson, T., & Barksdale, J. (1979). “KSOS-Development methodology for a secure operating system*”. *National Computer Conference*. Retrieved from http://www.computer.org/csdl/proceedings/afips/1979/5087/00/50870365.pdf.

Bishop, Matt. (2003). “Early Computer Security Papers: Ongoing Collection”. *University of California at Davis*. Retrieved from http://nob.cs.ucdavis.edu/history/CD/.

Chairman of the Joint Chiefs of Staff. (2011). “CJCSI 6510.01F: Information Assurance (IA) and Support to Computer Network Defense (CND)”. Retrieved from http://www.dtic.mil/cjcs_directives/cdata/unlimit/6510_01.pdf.

Chief Information Officer. (ND). “DOD Open Source Software (OSS) FAQ”. *US Department of Defense*. Retrieved from http://DODcio.defense.gov/OpenSourceSoftwareFAQ.aspx.

DARPA. (2014). "Clean-slate Design of Resilient, Adaptive, Secure Hosts (CRASH)". *Open Catalog/ Information Innovation Office*. Retrieved from http://www.darpa.mil/OpenCatalog/CRASH.html.

DARPA. (ND). "High-Assurance Cyber Military Systems (HACMS).". Retrieved from: http://www.darpa.mil/Our_Work/I2O/Programs/High-Assurance_Cyber_Military_Systems_(HACMS).aspx.

DISA (ND). "APL Testing and Certification". Department of Defense. Retrieved from http://www.disa.mil/services/network-services/ucco.

DISA. (ND). "Our Work/DISA 101". *Defense Information Systems Agency*. Retrieved from http://www.disa.mil/About/Our-Work.

DISA. (2009). "Defense Switched Network Information Assurance Test Plan Version 2". *DISA Joint Interoperability Test Command*. Retrieved from http://jitc.fhu.disa.mil/apl/ucapl/dsndocs/dsn_ia_test.pdf.

DISA (2010). "Ports, Protocols, and Services Management (PPSM)". *DISA*. Retrieved from http://www.disa.mil/news/conferences-and-events/~/media/Files/DISA/News/Conference/CIF/Briefing/ppsm_briefing.pdf.

DISA. (2014a). "Enabling the Joint Information Environment (JIE): Shaping the Enterprise for the Conflicts of Tomorrow". Retrieved from http://www.disa.mil/About/Our-Work/~/media/Files/DISA/About/JIE101_000.pdf.

DISA. (2014b). "Security Technical Implementation Guides (STIGs)". *Information Assurance Support Environment*. Retrieved from http://iase.disa.mil/stigs/Pages/index.aspx.

DISA. (2014c). “Joint Information Environment: Strategic Plan 2014-2019”. Retrieved from http://www.disa.mil/~/media/Files/DISA/About/Strategic-Plan.pdf.

DISA. (2014d). “Press Release- DISA: We Must Build On JRSS Momentum”. Retrieved from http://www.disa.mil/News/PressResources/2014/JRSS.

DTIC. (ND). “Overview of Department of Defense Issuances”. DOD. Retrieved from: http://www.disa.mil/About/DISA-Issuances.

Department of Defense. (2010). “DODI 8100.04 DOD Unified Capabilities (UC)”. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/810004p.pdf.

Department of Defense. (2012). “DOD 5200.44: Protection of Mission Critical Function to Achieve Trusted Systems and Networks (TSN)”. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/520044p.pdf.

Department of Defense. (2012). “DOD 5200.01: DOD Information Security Program: Overview, Classification, and Declassification”. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/520001_vol1.pdf.

Department of Defense. (2014). “DODI 8330.01: Interoperability of Information Technology (IT), Including National Security Systems (NSS)”. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/833001p.pdf.

Fay-Wolfe, V. (ND). “Operating Systems”. *University of Rhode Island*. Retrieved from: http://homepage.cs.uri.edu/book/operating_systems/operating_systems.htm.

Garfinkel, S., Spafford, G. (1996). *Practical UNIX and Internet Security: Second Edition.* Retrieved from http://www.diablotin.com/librairie/networking/puis/index.htm.

Hissam, S. (1998). “DOD Security Needs and COTS-Based Systems. *Carnegie Mellon Software Engineering Institute*. Retrieved from: https://resources.sei.cmu.edu/asset_files/WhitePaper/1998_019_001_29666.pdf.

Janssen, C. (ND). “Sandboxing”. *Techopedia*. Retrieved from: http://www.techopedia.com/definition/25266/sandboxing.

Kenyon, H. (2012). “DARPA’s CRASH Program Reinvents the Computer for Better Security”. *Breaking Defense*. Retrieved from: http://breakingdefense.com/2012/12/darpa-crash-program-seeks-to-reinvent-computers-for-better-secur/.

Koester, D. (2010). “Test & Evaluation of the NR-KPP”. *DISA*. Retrieve from http://jitc.fhu.disa.mil/jitc_dri/testEvalNrkpp101209.ppt.

Lam, D. D., & Carayannis, E. G. (2011). “Standard Insecurity: How, Why and When Standards Can Be a Part Of The Problem”. *Journal of the Knowledge Economy*, 2(2), 234-248. Retrieved from http://dx.doi.org/10.1007/s13132-010-0029-0.

Landwehr, C. (2004). “Improving Information Flow in the Information Security Market: DOD Experience and Future Directions”. *University of Maryland*. Retrieved from: http://download.springer.com/static/pdf/905/chp%253A10.1007%252F1-4020-8090-5_12.pdf?auth66=1421580994_e309ab12b119474b365770b9775d955c&ext=.pdf.

Lemley, L. (ND). “Chapter 8: Operating Systems and Utility Programs”. Retrieved from: http://uwf.edu/clemley/cgs1570w/notes/Concepts-8.htm.

Marsanu, N., Sichitiu, C., Sichitiu, G. (2010). "Consideration about Computer Networks Security under Various Operating Systems". Retrieved from http://www.jaqm.ro/issues/volume-5,issue-4/pdfs/4_marsanu_sichitiu_sichitiu.pdf.

Maucione, S. (2014). "Cybersecurity Upgrade Brings DOD Closer To JIE Implementation". *Inside the Pentagon, 30*(43). Retrieved from http://search.proquest.com/docview/1615200821?accountid=8289.

Microsoft. (2006). "Chapter 1: Functional Comparison of UNIX and Windows". *TechNet*. Retrieved from https://technet.microsoft.com/en-us/library/bb496993.aspx.

Mishory, Jordana. (2014). "CIO: DOD Must Finish JIE Requirements Before Defining Workforce Needs". *Inside the Pentagon*. Retrieved from http://search.proquest.com.ezproxy1.apus.edu/docview/1562618503?pq-origsite=summon.

Morrison, M. (2013). "The Acquisition Supply Chain and the Security of Government Information Technology Purchases". *Public Contract Law Journal, 42*(4), 749-792. Retrieved from http://search.proquest.com/docview/1428978021?accountid=8289.

Multics. (2015). "Summary of Multics". Retrieved from http://www.multicians.org/history.html.

NA. (2013). "DISA Team Develops New STIG Process". *US Fed News Service, Including US State News* Retrieved from http://search.proquest.com/docview/1355503446?accountid=8289.

N.A. (1978). "Secure Minicomputer Operating System (KSOS) Executive Summary- Phase 1: Design of the Department of Defense Kernelized Secure Operating System". *Ford*

*Aerospace & Communications Corporation*. Retrieved from http://csrc.nist.gov/publications/history/ford78.pdf.

Nakashima, E. (2012). “With Plan X, Pentagon Seeks to Spread U.S. Military Might to Cyberspace”. *The Washington Post*. Retrieved from: http://www.washingtonpost.com/world/national-security/with-plan-x-pentagon-seeks-to-spread-us-military-might-to-cyberspace/2012/05/30/gJQAEca71U_story.html.

NIST. (ND). “National Vulnerability Database: Automated Vulnerability Management, Security Measurement, and Compliance Checking”. Retrieved from https://nvd.nist.gov/home.cfm.

NIST. (ND). “FISMA: Detailed Overview”. *Computer Security Division: Computer Security Resource Center*. Retrieved from http://csrc.nist.gov/groups/SMA/fisma/overview.html.

NIST. (2013). “NIST Special Publication 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations”. Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.

NIST. (2013). “Security and Privacy Controls for Federal Information Systems and Organizations”. *Joint Task Force Transformation Initiative*. Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.

NIST. (2015). “NIST General Information”. *Public Affairs Office*. Retrieved from http://www.nist.gov/public_affairs/general_information.cfm.

ODASD. (2015). “Initiatives: Open Systems Architecture (OSA)”. Office of the Deputy Assistant Secretary of Defense: Systems Engineering. Retrieved from http://www.acq.osd.mil/se/initiatives/init_osa.html.

Perrin, C. (2010). “UNIX vs. Microsoft Window: How System Designs Reflect Security Philosophy”. *TechRepublic*. Retrieved from http://www.techrepublic.com/blog/it-security/unix-vs-microsoft-windows-how-system-designs-reflect-security-philosophy/.

Perrine, T. (2002). “The Kernelized Secure Operating System (KSOS)”. *;login: The Magazine of USENIX & SAGE*. Retrieved from http://c59951.r51.cf2.rackcdn.com/5085-1255-perrine.pdf.

Sustar, Lee. (2015). “Defense from the Top: FISMA”. *SC Magazine*. Retrieved from http://www.scmagazine.com/defense-from-the-top-fisma/article/393024/.

Turnbull, J. (ND). “Security Difference Between Linux and Unix”. *TechTarget*. Retrieved from http://searchenterpriselinux.techtarget.com/answer/Security-differences-between-Linux-and-Unix.

Van Atta, R. (ND). “Fifty Years of Innovation and Discovery”. *50 Years of Bridging the Gap*. Retrieved from http://www.darpa.mil/WorkArea/DownloadAsset.aspx?id=2553.

Vanderbilt. (ND). “Introduction to UNIX- Part 1: Basic Concepts”. *Vanderbilt University Center for Structural Biology*. Retrieved from: http://structbio.vanderbilt.edu/comp/unix/part01.php.

Wait. P. (2012). “DARPA Seeks ‘Plan X’ Cyber Warfare Tools”. *Information Week: Dark Reading*. Retrieved from: http://www.darkreading.com/risk-management/darpa-seeks-plan-x-cyber-warfare-tools/d/d-id/1105938.

Watson, R. (2010).  “Capsicum: Practical Capabilities for UNIX”. USENIX Association Retrieved from: https://www.youtube.com/watch?v=raNx9L4VH2k.

Watson, R. (2012). “Capsicum: Practical Capabilities for UNIX”. University of Cambridge. Retrieved from: http://www.cl.cam.ac.uk/research/security/capsicum/.

The White House. (2000). “Circular No. A-130 Revised”. Office of Management and Budget. Retrieved from http://www.whitehouse.gov/omb/circulars_a130_a130trans4/.

## Appendix A

### Glossary of Terms

| | |
|---|---|
| ACL | Access Control List |
| AES | Advanced Encryption Standard |
| API | Application Program Interface |
| APL | Approved Products List |
| ARPA | Advanced Research Projects Agency |
| BSD | Berkeley Software Distribution |
| CAC | Common Access Card |
| CIO | Chief Information Officer |
| COTS | Commercial off the Shelf |
| CJCSI | Chairman of the Joint Chiefs of Staff |
| CPU | Central Processing Unit |
| CRASH | Clean-slate design of Resilient, Adaptive, Secure Hosts |
| DAC | Discretionary Access Control |
| DARPA | Defense Advanced Research Projects Agency |
| DCA | Defense Communications Agency |
| DHS | Department of Homeland Security |
| DIACAP | DOD Information Assurance IA Certification & Accreditation Process |
| DISA | Defense Information Systems Agency |
| DISN | Defense Information Systems Network |
| DOD | Department of Defense |
| DTIC | Defense Technical Information Center |
| EA | Enterprise Architecture |
| EACL | Extended Access Control List |
| EFS | Encrypting File System |
| EOC | Enterprise Operations Center |
| FISMA | Federal Information Security Management Act |
| FSO | Field Security Office |
| GEOC | Global Enterprise Operations Center |
| GID | Group ID |
| GIG | Global Information Grid |
| GOTS | Government off the Shelf |
| GUI | Graphical User Interface |
| JIE | Joint Information Environment |
| JITC | Joint Interoperability Test Command |
| JRSS | Joint Regional Security Stacks |
| JTSO | JIE Technical Synchronization Office |
| KSOS | Kernelized Secure Operating System |
| KVM | Kernelized Virtual Machine |
| LPS | Lightweight Portable Security |
| MAC | Mandatory Access Control |
| NIST | National Institutes of Standards and Technology |
| NKSS | N-Kernel System Software |

| | |
|---|---|
| NSS | National Security System |
| NTFS | New Technology File System |
| IEEE | Institute of Electrical and Electronics Engineers |
| IN | Information Network |
| IO | Interoperability |
| OMB | Office of Management and Budget |
| OS | Operating System |
| OSA | Open Systems Architecture |
| OSS | Open Source Software |
| POSIX | Portable Operating System Interface |
| STIG | Security Technical Information Guide |
| UC | Unified Capabilities |

School of Science, Technology, Engineering, and Math

MS in Information Technology

The thesis for the master's degree submitted by

Karen Melissa Littlewood

Security Through Exclusivity: Information Assurance Via A DOD Operating System

has been read by the undersigned. It is hereby recommended

for acceptance by the faculty with credit to the amount of

3 semester hours.

(Signed, first reader) ____________________________________ (Date) _____________

(Signed, second reader, if required) ________________________ (Date) ______________

Recommend for approval on behalf of the program

(Signed) ___________________________________ (Date) _______________

Recommendation accepted on behalf of the program director

(Signed) ___________________________________ (Date) _______________

Approved by academic dean

This capstone has been approved by Dr. Novadean Watson-Stone for submission, review, and publication by the Online Library.

Author's name: Karen Melissa Littlewood

Title: Security Through Exclusivity: Information Assurance Via A DOD Operating System

Professor: Dr. Miriam Masullo

Second reader, if required:

Program: Master's of Science in Information Technology with a concentration in Information Assurance and Security

Pass with Distinction: □ YES □ NO

Keywords/Descriptive Terms:

[ ] Contains Security-Sensitive Information