2016 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
VEHICLE ELECTRONICS AND ARCHITECTURE (VEA) TECHNICAL SESSION
AUGUST 2-4, 2016 - NOVI, MICHIGAN

INCORPORATING CYBERSECURITY INTO THE SOFTWARE DEVELOPMENT LIFECYCLE

Jonathan Dorny
Chief Technical Officer
Control Point Corporation
Goleta, CA

Susan Ingenthron
Systems Engineering Lead
JTDI PMO
Huntsville, AL

Joe Erian, SW Systems Security Engineer
Matt Tarka, SW Systems Engineer
Kurt Hansen, Information Assurance Systems Engineer
Control Point Corporation
Goleta, CA/ Troy, MI

ABSTRACT

The proliferation of information technology adds expanded capabilities and exposes new vulnerabilities through cyber warfare. To combat new threats software quality must go beyond CMMI maturity levels and embrace a software development lifecycle (SDLC) with measurable cybersecurity assurance. Standard cybersecurity artifacts throughout the SDLC should be expected and available for inspection. Integrated software applications can confidently and rapidly reduce their threat exposure by incorporating reusable data management components with a pedigree of cybersecurity SDLC assurance evidence.

INTRODUCTION

Cyber warfare entails actions performed by a nation-state to penetrate computers and networks of another nation to disrupt operations [1]. Some foreign governments have made cyber warfare an integral part of their comprehensive military and political strategy. The Department of Defense (DoD) attempts to exploit the vulnerabilities of adversaries of the United States while trying to minimize our own exposure. The proliferation of information technology adds a wealth of new capabilities to defense systems including availability, access, accuracy, and rapid insertion of improved function to address changing threats. However, the increased adoption of net-centricity introduces new vulnerabilities to systems.

As combat vehicles and other DoD systems connect to global information networks, they begin to face new and more serious cybersecurity threats. Unmanned robotic systems enable a deep force projection with improved safety to the remote operators. As a consequence, additional risk is introduced by the very methods of networked control that carry command messages to equipment that is unaware of the true identity of the commanding authority.

The exploitations of security vulnerabilities within commercial organizations raise increasing concerns for defense systems using similar technologies. The demonstration of a hacker taking remote control of automobile driving functions [2] parallels that of the Iran–U.S. RQ-170 Drone incident, where a foreign government managed to capture a US Drone. The Iranian government revealed live video of landing the plane without a flaw, suggesting that the UAV command and control was compromised while in midst of a mission.

Whether or not an adversary uses a rocket propelled grenade (RPG) to damage a vehicle, renders an embedded computer system unusable, or exploits critical information, the result is the same: taking an asset and/or advantage away from the operator can result in catastrophic mission failure. An RPG can only fire once. A cyber-attack is repeatable until the vulnerability is discovered and eradicated.

The complexity of solutions has grown as threats have increased. Organizations can no longer afford to build from scratch and avoid proven integrated solutions for cybersecurity requirements any more than for other software functionality. Additionally, Accreditations through the Department of Defense Information Assurance Certification and Accreditation Process (DIACAP) or the DoD 8510.01 Risk Management Framework (RMF) process are a discrete event and must be supplemented with lifecycle responses to

new threat vectors. *This paper presents techniques for improving cybersecurity that include processes and an architecture framework that use common software components*. Common integrated cybersecurity components provide rapid, clear, concise, and repeatable risk reduction within the Risk Management Framework (RMF) regardless of the software applications being developed or integrated. Low-level security implementation for each new software development effort presents new coding vulnerabilities and new attack vectors, to cyber adversaries. Designing a tank with no armor and designing a network connected system with no common 'cyber-armor' are equally unthinkable.

A comprehensive solution to the accelerated growth of cybersecurity risks includes:

1. Software lifecycle processes adhering to specific security practices including supplier cybersecurity assurance
2. A long-term sustainment strategy that delegates a significant number of security functions to a trusted component
3. A secure data management architecture designed to consolidate protection of data storage and transfer

This paper will describe each of these cybersecurity risk reduction methods and then presents a case study of a DoD Program Management Office (PMO) implementing them for rapid approval of technical data integration across multiple service branches.

CYBERSECURE SOFTWARE LIFECYCLE

The continuous improvement of software lifecycle security requires periodic evaluations of cybersecurity processes and design provisions. Dr. William Scherlis, Director of the Institute for SW Research at Carnegie Mellon University recently addressed the DoD Maintenance Symposium focusing on software sustainment [3] including the increasing significance of cybersecurity in any software sustainment plan. Over the last several years, Control Point Corporation (CPC) has incorporated cybersecurity as an integral component of our SEI Common Maturity Model Integration (CMMI) Maturity Level 3 processes encompassing the full Software Development Life Cycle (SDLC). Simple process improvements can efficiently yield significant cybersecurity dividends on the path to compliance with all practices within the System Security Engineering Capability Maturity Model (SSE-CMM) [4].

Figure 1 illustrates how generic SDLC phases relate to distinct cybersecurity enablers. The outer ring identifies enablers in each phase that enhance long-term sustainment against new threats and discovered vulnerabilities. The sections below summarize evidential artifacts that should be available for inspection for each phase as assurance that an organization is implementing these enabling security practices.

Figure 1- Cybersecurity SDLC Assurance Process

Requirements Analysis Phase

The first SLDC cybersecurity enabler is a *reference library of cybersecurity documents* available to systems engineers, software architects, software developers, and cybersecurity professionals. Basic library contents are the DoD Instruction 8500.01 for Cybersecurity, DoD 8510.01 for the Risk Management Framework (RMF) and the current Security Technical Implementation Guides (STIGS). More robust libraries will include lower level documentation, such as the Federal Information Processing Standards (FIPS) publications. Reference libraries are not limited to instructions, publications, and directives, but should also include online resources such as the DoD Information Technology Portfolio Repository (DITPR) and Cryptographic Module Validation Program (CMVP). Advanced libraries incorporate industry vulnerability databases, ongoing threat signature/response analyses for major incidences and cybersecurity community of interest discussion group archives. These resources provide considerable guidance and constraints regarding the selection of third-party components, e.g. approved FIPS 140-2 cryptographic modules and algorithms. Failure to select components approved within these libraries introduces (1) technical risk associated with technical implementations, e.g. late development defect fix required for weak encryption algorithms or modules or (2) programmatic risk resulting from lengthy approval for 'out of library' software components or forced removal from the field.

The second security enabler is a *reusable database of cybersecurity requirements*. The requirements database extracts all pertinent elements from the library and organizes them into functional, environment, or interface requirements. Construction or approval requirements are also captured, such as test code coverage metrics. CPC has developed a DOORS repository containing 536 cybersecurity requirements extracted from Security and Privacy Controls for Federal Information Systems and Organizations (NIST SP800-53 Rev 4). The cybersecurity requirements should include traceability to their source as well as allocation to reusable software components, including 3rd party applications and operating system services. Interrogating a company representing a potential cybersecurity software component regarding their requirements management database is an essential qualifying technique.

Design and Implementation Phases

Evolving coding standards that address cybersecurity lessons learned are a key indicator of mature cybersecurity SDLC process. Lessons learned come from community-developed dictionaries of weaknesses (e.g. Common Weakness Enumeration) and frequently encountered developer errors identified through code reviews. The objective of the continuously improved coding standards is to train a complete staff of cyber-aware individuals with a consistent approach to avoiding common implementation errors. CPC has developed and maintained a set of DoD coding standards for Java and C++ applications containing fifteen potential coding issues and their implementation rules for avoiding cybersecurity specific issues. These rules are specific enough to ensure vulnerabilities are eliminated, but general enough to allow a myriad of implementation approaches.

Source code development assistance tools established within the Software Development Plan (SDP) represent effective evidence of coding error avoidance during the software realization process. Source code development assistance tools are plug in applications that automatically enforce defined coding standards, identify potential weakness to the developer during their daily coding activities, and fulfill required Information Assurance (IA) controls. Coding standard tools should be selected to integrate with an organization's integrated development environment (IDE) with configurable rules that align with the latest release of company coding standards. Integration within the IDE provides efficient visualization in daily development and avoids costly delays imposed by periodic code reviews. Additionally, static coding bug detection tools should be selected that alert developers to potentially weak coding practices that open up vulnerabilities. There are many free code style and security bug tools that any organization with a cybersecurity posture can implement without any cost to the company beyond installation.

The review of source code provides an essential cybersecurity control. *An automated static source code scanning report* is the most consistent method with the most comprehensive records. In the previous paragraph we identified free tools with security modules that provide a good first order scanning of source code. However, dedicated cybersecurity tools for automated source code scanning provide the highest level of confidence in a product under investigation regarding its cybersecurity posture. A prominent software security tool for DoD applications utilized by CPC is Hewlett Packard's Fortify software. This software checks against all STIGS enabling a standard set of rules and reports for code prior to submission for proposed release. Employment of an automated static source code scanning tool is one indication of an organization with a highly mature cybersecurity posture for their software products.

Testing Phase

While the SDLC needs to include system-level testing that incorporates environmental scanning tools designed to comprehensively test applications and software components, environment scanning will be prevalent with all developers. Typically, this is performed by a DoD organization on behalf of the software fielding agency. Environment scanning looks for vulnerabilities that can only be detected in the anticipated or target environment. *An environment scanning report* documents threats through open ports and protocols, firewall settings, and operating system configuration vulnerabilities by scanning before and after candidate software installation. Examples of environment scanning software include Retina, ACAS, Nessus, and OpenVAS. Environmental scanning can also identify software configuration risks, e.g. use of default settings, allowing null sessions, etc., that are not prevented by other means, namely coding standards or reviews.

Integration testing between developed software and 3rd party software services is fundamental to assessing risks inherited from the 3rd party. Usually this is performed by an external neutral party (black box testing) or the company's internal information security staff (white box testing). Unlike source code and environment scanning, integration testing is an active process. As the team probes for vulnerabilities, they generally perform a white box test and have all technical documentation and design documents provided to them prior to the examination. The information security team will launch a series of attacks on the software to expose exploitable vulnerabilities, such as weak authentication, transmission of clear text passwords, unencrypted and accessible data-at-rest, etc. The information

security team documents their actions, findings, and recommendations such that developers can minimize the vulnerabilities and/or exploitable components (attack vectors).

Continuous Improvement

An organization with a sustainable cybersecurity posture for their software will have deployment architecture artifacts that identify the required ports/protocols required by the software and memorandum of agreement (MOA) documentation with the computing environment owner that ensure that only those ports and protocols are open. System Interface Agreement (SIA) templates are used to quickly document inherited controls. Allocation of cybersecurity requirements to physical security owners, data warehouse owners, network owners, computing center owners, computing device owners, and operating system owners is the process of establishing cybersecurity controls. *Draft MOA/SIA documentation for rapid approval of agreements between the software fielding organization and environment owners* is a part of the cybersecurity template library for any mature cybersecurity software development organization. At CPC we have found that the time to gain approval to operating in a specific deployment is decreased by almost 50% when similar MOAs/SIAs are used and common deployment architecture from a previous fielding is leveraged.

Organizations with a mature cybersecurity posture will also be able to produce an *organizational chart that includes one or more staff members with certifications recognized by DoDD 8570.1,* including Certified Information System Security Professional (CISSP) or Certified Ethical Hacker (CEH). As mentioned before, an organization with a mature cybersecurity posture will train all organizational members in best practices for security, but the information security staff will act as "cybersecurity ninjas" overseeing the depth and breadth of the cybersecurity SDLC. These individuals also will remain trained and cognizant of the latest threats and emerging challenges.

CYBERSECURITY DELEGATION

Any "bolt-on" cybersecurity must include some knowledge and assurance of the suitability and risk to the overall application. This concept, which helps to avoid past failures, was discussed by Dr. Scherlis at the DoD Maintenance Symposium 2015. This software approach is similar to applique armor or the hardware addition of firewalls, cross domain solutions, and encryption devices. The software "bolt-on cybersecurity" posture invites the incorporation of information assurance components at a more comprehensive level of integrated capability to address standard and newly discovered vulnerabilities. The term "bolt-on cybersecurity" is subject to the misconception that that information security is applied to a finished item or system as an afterthought, rather than deliberately incorporated at multiple levels during the entire SDLC. To avoid this misconception, CPC prefers to use the term "delegation based cybersecurity", which proactively leverages security control inheritance. This common component approach combines existing, proven security software components with newly developed software to meet the desired functional requirements while possessing strong, mature cybersecurity features.

Component Based Software

Component-based software engineering (CBSE) is trending toward compilation of higher and higher levels of integration of specialized components. This CBSE trend is analogous to the way in which computing hardware design for many original equipment manufacturers (OEMs) progressed from combining elements at the transistor level, to the integrated circuit level, the circuit card level, the computing box level, and so forth. Service-oriented architecture (SOA), modular open interface applications, open source software, and rapid development frameworks shorten development times, increase stability, reduce risk, and question the financial sanity of creating any software completely from scratch.

Software defects are minimized by a strong service-based architecture composed of specialized resource components produced by expert vendors with a broad user-base. The broad user-base adds a maturity to the specified software functionality within a wide range of deployments, use cases, and conditions. The more users and uses provide maturity and minimizes the sustainment risk for new applications. At the same time, the integration of software components from outside vendors can decrease the direct control and comprehensive knowledge of the overall cybersecurity posture and vulnerabilities of your own integrated product.

Dr. Scherlis identified three dimensions of overall software lifecycle sustainment that apply to analyzing the adequacy of a cybersecurity component: (1) architecture, (2) agility, and (3) assurance. The following paragraphs address each of these components.

A cybersecurity architecture must identify the functional components for an application that address the security posture of storage, communication, data sharing, and information assurance between software components and across computing elements of a network. Additionally, the architecture should include the human element of access control and data protection. A reusable cybersecurity architecture must demonstrate a process that enables repeatable quality and consistent protection across implementations. A repeatable process is necessary for any software seeking Risk Management Framework (RMF) accreditation as an application or system, which is only

effective if it can be applied consistently across shared applications to accelerate the process of accreditation.

Adaptive Practices

Agile software lifecycle execution practices will deliver quality across a myriad of implementation solutions. For cybersecurity, this means that the daily practices of software definition, implementation, and sustainment are responsive to changing vulnerabilities, threats, approval mechanisms, and organizational administrations while adhering to the approved architecture. Execution must uncover and mitigate latent implementation vulnerabilities in your own software products as well as monitoring of Information Assurance Vulnerability Alerts (IAVAs) for 3rd party software components and operating systems. Effective standard practices will adapt to changing threats based on technology maturity and cyber warfare advancements. According to a 2014 study by Federal Financial Institutions Examination Council (FFIEC), which includes the Board of Governors of the Federal Reserve System, "participating in information-sharing forums is an important element of an institution's risk management processes and its ability to identify, respond to, and mitigate cybersecurity threats and incidents." [5] Establishment of a common cybersecurity architecture and "delegation-based cybersecurity" components can engender a community of interest for information sharing on applicable threats, incidents and mitigations including derivation of standard response practices.

The inspection of cybersecurity artifacts throughout the SDLC assures that cybersecurity quality has been achieved and is sustainable. Accreditations through the DIACAP or RMF process provide strong evidence of a rigorous development process. Obtaining an Army Certificate of Networthiness (CoN) or inclusion within the Department of Navy Application and Database Management System (DADMS) registry are intermediate assurance artifacts applicable to the Army, Navy, and USMC. However, each of these assurance certification artifacts is "after the fact" evidence. Cybersecurity process artifacts validating each phase of a software component's lifecycle provide a comprehensive depth of assurance. According to research from the Software Engineering Institute (SEI) at Carnegie Mellon University an increasing quantity of software organizations are identifying that a large number of their vulnerabilities stem from design weaknesses and not coding vulnerabilities [6]. With more than a third of the 940 Common Weakness Enumerations (CWEs) and 75% of the 25 most dangerous software errors attributable to design the research suggests that artifacts supporting design process for cybersecurity that is compliant with a common cybersecurity architecture will alleviate software vulnerabilities.

SECURE DATA MANAGEMENT ARCHITECTURE

As CPC has worked DoD systems, we have observed a need for a large number of secure data management functions. Figure 2 depicts a secure data management architecture where one common software component provides many cybersecurity functions for an integrated set of applications. Using 3rd party library components for low level security functions (e.g. Transport Layer Security- TLS) is not new. However, "delegation-based cybersecurity" aims to utilize a more high-level comprehensive resource that simplifies secure integration of software.

The Department of Defense Architecture Framework (DoDAF) identifies three primary information exchange methods, message-based, file-based, and database-oriented. As presented in the figure, the secure data management component provides a local messaging interface within a secure enclave on a computing node for message-based exchange. The data management component consumes the messages, persists the data and routes messages within that protected enclave to other applications following stored rules. Additionally, applications can pass files through the data management component digitally signed for non-repudiation. All messages, files, and database elements are encrypted at rest to meet cybersecurity requirements.

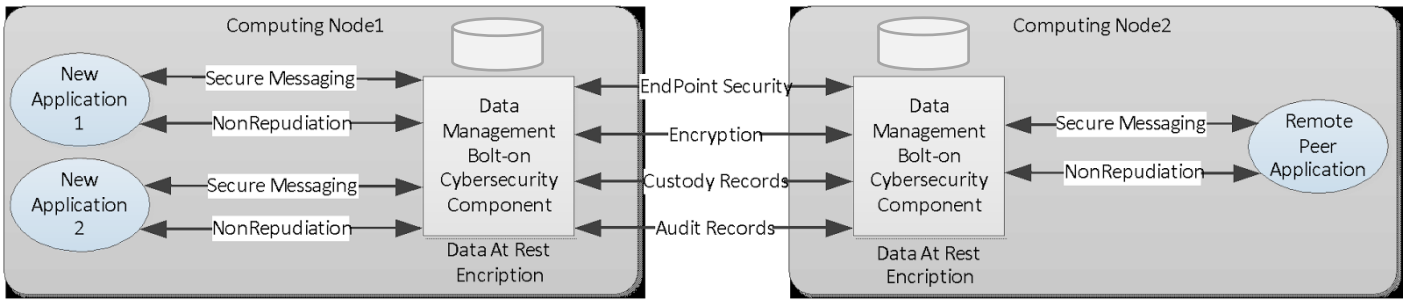

Figure 2 – A Secure Common Data Management Architecture

In the proposed architecture the secure data management component is designated as the only software component with ports and protocols exposed outside the computing

node. The data management component establishes an endpoint security channel to a peer data management component on another computing device when routing rules for messages, files, or database synchronization indicate a remote recipient application. Data is encrypted prior to transfer across the secure channel. Where a multi-hop file transmission is traversed the data management component maintains records of each computing node that has had temporary custody of files. Additionally, audit records are maintained for data integrity inspection.

The secure data management software component described combines many cybersecurity services that relieve responsibility from the consumer applications. Cybersecurity controls can be allocated to the data management component to alleviate implementation by the other applications, nevertheless, cybersecurity is maintained. Agility is achieved as each integrated application can determine one or more methods of information exchange applicable to their implementation without impact to the overall cybersecurity posture. A secure data management component with sufficient assurance evidence provides a strong, common confidence of removed vulnerabilities. Additionally, a large user base and varied deployment environment of a common secure data management component is conducive to rapid defect discovery and elimination. Combined with a continuous improvement process defects will be eliminated at an accelerated rate to create a highly supported and well-structured security infrastructure.

JDMS

A common data management service is available to DoD agencies that desire to implement cybersecurity provisions via a software component. The Joint Technical Data Integration (JTDI) Program Management Office (PMO) has responded to their operational needs statement with a Service-Oriented Architecture (SOA) based information management application indicative of the aforementioned architecture. Chartered to move technical data in a secure manner, JTDI PMO has inserted technology improvements and cybersecurity enhancements into the lifecycle of their JTDI Delivery Management Service (JDMS) software that make it a candidate component within "delegation-based cybersecurity" architecture.

JDMS supports a plug-in cybersecurity architecture designed for integration of applications and services as the secure data management service presented in Figure 2. Integrated applications share logistics information through the JDMS common resource. Logistics business messages are exchanged between an application and JDMS. Data integrity and non-repudiation are enforced by JDMS after checking that messages are well-formed and data is valid. Metadata is stored within an embedded database encrypted with an AES 256 Bit Encryption and protected by uniquely salted hashes within JDMS. Predefined rules for information organization and equipment-based configuration management associate appropriate metadata (e.g. IETM version information) with content (e.g. IETM files). Loosely coupled applications locally share data within a secure enclave protected by firewalls from external communications outside of JDMS. Equipment-based "need to know" rules trigger JDMS when communications between JDMS nodes within a JDMS network must synchronize data (messages, database content, and files). Between nodes, JDMS provides data integrity (digital signature, encryption), data distribution (certificates, endpoint security, and asymmetric encryption to establish TLS secure channel communication), access control (distribution statements) and custody logging (network and computing stops).

JDMS supports agile component-based software engineering practices for cybersecurity. JDMS is the latest technology advancement from JTDI that will subsume the Joint Knowledge Caching System (JKCS) deployed within the Army, Navy, and USMC for technical manual content file distribution and condition based maintenance (CBM) log collection. Following JTDI's reciprocation practices, JDMS will receive an Authority To Operate (ATO) with cybersecurity approvals in the Navy and then quickly gain approval within the other services. The Army CBM community has designated the information management architecture utilizing JDMS as applicable to all platforms and battalion and below logistics application integrations within the Common CBM+ Architecture (CCA). The CCA approval of plug-in data management with integral cybersecurity is a culmination of an Army Material Command (AMC) Systems Engineering (SE) Integrated Product Team (IPT) initiative involving the Tank Automotive Command (TACOM), Communications Electronics Command (CECOM) and Aviation and Missile Command (AMCOM) with their Research and Development (R&D) organizations. The AMC CBM+ SE IPT led by the Army G-6 office has developed into an information sharing forum for cybersecurity. The AMC CBM+ SE IPT has identified that the JDMS common data management architecture is applicable to embedded, mobile, PC-based, and Software as a Service (SaaS) implementations within Linux and Windows operating environments. The JDTI practices include continuous monitoring and mitigation of IAVAs. The architecture is currently part of a Course of Action (COA) analysis for applicability to Store and Forward (SaF) needs within a logistics network with intermittent connectivity.

JTDI strives to become a model for NAVIAR in using SDLC assurance evidence to efficiently receive accreditation. JDMS survives cybersecurity assurance

scrutiny with artifacts that may be reviewed by request with the JTDI PMO. (1) Many source documents used to obtain a JDMS DIACAP and ATO are controlled within the CPC cybersecurity library. (2) Extracted requirements are designated as specifically applicable to cybersecurity and traced to the library source documents to ensure requirements management when source documents are changed to address new threats. JDMS is written primarily in Java with some components written in C++. (3) CPC's Java and C++ coding standards are proprietary; however, the 15 sections regarding cybersecurity best practices are shared with the JTDI PMO. CPC has currently adopted NetBeans as our JDMS development IDE. (4) The NetBeans IDE selection affords plug-ins for code development control including CheckstyleBeans for coding rules and FindSecurityBugs for common security error detection. Several years ago the JTDI program pioneered the use of automated static source code vulnerability scanning within Naval Aviation (NAVAIR). (5) Today, the submission of HP Fortify scan reports are mandatory for each JDMS software release with all Category 1 and 2 findings eliminated or mitigated. (6) The JTDI PMO maintains the JDMS System Security Plan (SSP), DIACAP Certification and Accreditation (C&A) Plan, Security Test & Evaluation (ST&E) Plan, and DIACAP workbook of cybersecurity controls. These plans are sensitive, so they are not publically published. (7) JDMS received an Authority to Operate (ATO) for JDMS v1.0 in June 2016.

The future vision for sustainable cybersecurity incorporates "delegation-based cybersecurity" that inherits protection from common secure data management software components. For secure data management component reuse to become realized there is a need for an approved "delegation-based cybersecurity" architecture with repeatable success processes. Practitioners must be provided execution practice flexibility for independent, but compatible solutions. Cybersecurity component vendors must be accountable through evidentiary artifacts that assure adopters that their products have implemented a full software development lifecycle (SDLC) cybersecurity posture within their organization and their products. When each of these elements is present, "delegation-based cybersecurity" is a plausible and desirable solution. Data management software components, such as JDMS, provide a good opportunity to provide cybersecurity services as a byproduct of their other functions as they control data storage and communication.

REFERENCES

[1] Andress, Jason, Winterfeld, Steve, "Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners", Syngress. 2011.

[2] Claims Journal, "Jeep Hacking Incident Leads to Fiat Chrysler Recall of 1.4M Vehicles", http://www.claimsjournal.com/news/national/2015/07/27/264766.htm, 2015.

[3] Scherlis, William, "Software Sustainment- Looking Ahead", DoD Maintenance Symposium 2015", http://www.sae.org/events/dod/2015/attend/program/presentations/p4_scherlis_william.pdf

[4] International Systems Security Engineering Association (ISSEA), "Systems Security Engineering Capability Maturity Model (SSE-CMM)", ISO/IEC 21827, 2008.

[5] Federal Financial Institutions Examination Council, "Cybersecurity Threat and Vulnerability Monitoring and Sharing Statement", http://www.ffiec.gov/press/PDF/FFIEC_Cybersecurity_Statement.pdf, 2014.

[6] Kazman, Rick, Woody, Carol, "A Tool to Address Cybersecurity Vulnerabilities Through Design", https://insights.sei.cmu.edu/sei_blog/2016/02/a-tool-to-address-cybersecurity-vulnerabilities-through-design.html, 2016