
CrossTalk
March / April 2013 The Journal of Defense Software Engineering Vol. 26 No. 2
Supply Chain Risk Management

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
APR 2013		**00-03-2013 to 00-04-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
CrossTalk, The Journal of Defense Software Engineering. Volume 26, Number 2. March/April 2013	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **517th SMXS/MXDEA,6022 Fir Avenue,Hill AFB,UT,84056**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

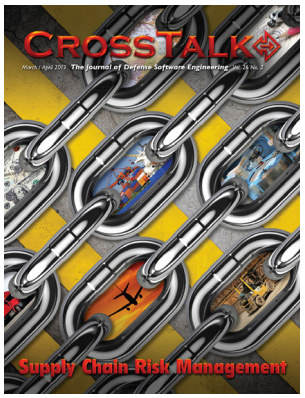
14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**44**	

TABLE OF CONTENTS

Departments



Cover Design by Kent Bingham

Supply Chain Risk Management

CrossTalk

NAVAIR Jeff Schwalb
DHS Joe Jarzombek
309 SMXG Karl Rogers

Publisher Justin T. Hill
Advisor Kasey Thompson
Article Coordinator Lynne Wade
Managing Director Tracy Stauder
Managing Editor Brandon Ellis
Associate Editor Colin Kelly
Art Director Kevin Kiernan

Phone 801-775-5555
E-mail stsc.customerservice@hill.af.mil
CrossTalk Online www.crosstalkonline.org

CrossTalk, The Journal of Defense Software Engineering is co-sponsored by the U.S. Navy (USN); U.S. Air Force (USAF); and the U.S. Department of Homeland Defense (DHS). USN co-sponsor: Naval Air Systems Command. USAF co-sponsor: Ogden-ALC 309 SMXG. DHS co-sponsor: National Cyber Security Division in the National Protection and Program Directorate.

The USAF Software Technology Support Center (STSC) is the publisher of **CrossTalk** providing both editorial oversight and technical review of the journal. **CrossTalk's** mission is to encourage the engineering development of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

Subscriptions: Visit <www.crosstalkonline.org/subscribe> to receive an e-mail notification when each new issue is published online or to subscribe to an RSS notification feed.

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the **CrossTalk** editorial board prior to publication. Please follow the Author Guidelines, available at <www.crosstalkonline.org/submission-guidelines>. **CrossTalk** does not pay for submissions. Published articles remain the property of the authors and may be submitted to other publications. Security agency releases, clearances, and public affairs office approvals are the sole responsibility of the authors and their organizations.

Reprints: Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with **CrossTalk.**

Trademarks and Endorsements: **CrossTalk** is an authorized publication for members of the DoD. Contents of **CrossTalk** are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, the co-sponsors, or the STSC. All product names referenced in this issue are trademarks of their companies.

CrossTalk Online Services:
For questions or concerns about crosstalkonline.org web content or functionality contact the **CrossTalk** webmaster at 801-417-3000 or webmaster@luminpublishing.com.

Back Issues Available: Please phone or e-mail us to see if back issues are available free of charge.

CrossTalk is published six times a year by the U.S. Air Force STSC in concert with Lumin Publishing <luminpublishing.com>.
ISSN 2160-1577 (print); ISSN 2160-1593 (online)

CrossTalk would like to thank DHS for sponsoring this issue.

Each person and organization in the supply chain path "touches," or has influence on, the security and resilience of software used to control products, systems, and services.

Just as with food and pharmaceuticals, software can be corrupted in ways that put users, organizations, and missions at risk. Software can become tainted by malware, exploitable weaknesses, and vulnerabilities. But no matter the method of compromise, those at the end of the supply chain are unwittingly exposed to the residual risk.

Information and communications technology supply chains are interdependent global ecosystems that consist of organizations, people, activities, information, and resources. And these complex ecosystems are vulnerable to a host of threats and hazards such as natural disasters, accidents, and malicious attack. Globalization of the commercial information and communications technology marketplace provides increased opportunity for anyone intent on harming the United States to gain unauthorized access to systems, data, and communications. Securing the global supply chain is integral to securing both our national security and the world economy.

The government and private sector own separate parts of the supply chain risk equation. This means that no single organization independently controls all the processes or possesses all the information required to manage the full risk. Public/private collaboration is crucial to supply chain risk management.

Our Supply Chain Risk Management (SCRM) program promotes the improvement of formal threat sharing processes, planning and investment documentation, supply chain incident reporting, national security systems standards, and the Federal cybersecurity workforce.

In concert with the DoD and NIST, the DHS Software Assurance program co-sponsors a forum during which our Federal, academic, and private sector partners discuss Software Assurance (SwA) risks and mitigation methods. This Software Assurance Forum has contributed several excellent resources to the software supply chain risk management community. These resources are available on the SwA Community Resources and Information Clearinghouse at <https://buildsecurityin.us-cert.gov/swa>.

Venues such as the SwA Forum are critical to our understanding of how suppliers incorporate security-aware practices into the production of software. Baseline understanding can inform risk-based decisions when purchasing software or contracting for software-reliant systems or services.

This issue of **CrossTalk** includes articles focused on advancing SCRM that we hope will provide valuable insights into SCRM techniques, research methods, and models that target vulnerabilities in the supply chain. Thank you for taking advantage of this excellent resource.

Roberta Stempfley
Acting Assistant Secretary
Office of Cybersecurity and Communications
Department of Homeland Security

We Cannot Blindly Reap the Benefits of a Globalized ICT Supply Chain!

Don Davidson, Office of the DoD Chief Information Officer
Stephanie Shankles, Booz Allen Hamilton

Abstract. Information and Communication Technology (ICT) Supply Chain Risk Management (SCRM) seeks to manage and mitigate cyber and supply chain risk throughout an acquisition and sustainment lifecycle for an element or a system. It is a multi-disciplinary challenge that requires contributions and collaboration among many disciplines. Key areas include systems engineering, system security engineering, information security, software development, application security, supply chain and logistics planning and management, IT resiliency, and risk management. While many areas are making great strides in developing and implementing best practices and tools to combat their individual cyber challenges, it is imperative for successful enterprise risk management to view the challenge holistically and align common best practices and initiatives, some from/for the public sector and some from/for the private sector.

Introduction

A holistic view of supply chain risk management is one of the 12 key areas in the United States Comprehensive National Cyber Security Initiative (CNCI). CNCI-SCRM is a Federal Government wide multi-pronged approach for managing risk while operating in a global supply chain. Managing this risk requires a greater awareness of the threats, vulnerabilities, and consequences associated with acquisition decisions; the development and employment of tools and resources to technically and operationally mitigate risk across the lifecycle of systems, products and/or elements (from design through retirement); the development of new acquisition policies and practices that reflect the complex global marketplace; and partnership with industry to develop and adopt supply chain and risk management standards and best practices.

"Software and hardware are at risk of being tampered with even before they are linked together in an operational system. Rogue code, including so-called logic bombs, which cause sudden malfunctions, can be inserted into software as it is being developed. As for hardware, remotely operated "kill switches" and hidden "backdoors" can be written into the computer chips used by the military, allowing outside actors to manipulate the systems from afar. The risk of compromise in the manufacturing process is very real and is perhaps the least understood cyberthreat. Tampering is almost impossible to detect and even harder to eradicate."
(DEPSECDEF Lynn in FOREIGN AFAIRS in Sep 2010.)

Globalization has brought a unique set of SCRM challenges and threats to the U.S. Government and industry, especially with our ever-increasing reliance on ICT products and services to meet mission and business needs and the interconnected nature of our IT systems. Threats to the ICT systems are varied, complex and demonstrate a wide array of motivations for attack. They range from counterfeit items made for a quick profit, intentional threats such as malicious code or hardware Trojans, to poor software development practices that create software vulnerabilities or hardware quality issues. These are all the more dangerous because ICT is found everywhere in our environment, from our home entertainment systems, mobile devices that hold/move our personal information, to our infrastructure's financial and energy sectors, and even to national security systems and weapons systems.

Challenges With Globalization

Globally, USG represents a relatively minor share of the ICT product and service market for the industry and alone does not command the market power to drive commercial suppliers to substantially change their SCRM practices. However, USG is an important stakeholder in the process because of their role in national and global security and the variety of valuable lessons learned and best practices they can provide because they are such a diverse organization. The ICT SCRM challenge is not limited to USG, it impacts every government and commercial organization that acquires and uses ICT products and services. Furthermore, many of the suppliers of ICT products and services also find themselves acquiring ICT products and services to integrate into their own solutions and therefore have a common interest in facing the ICT SCRM challenge.

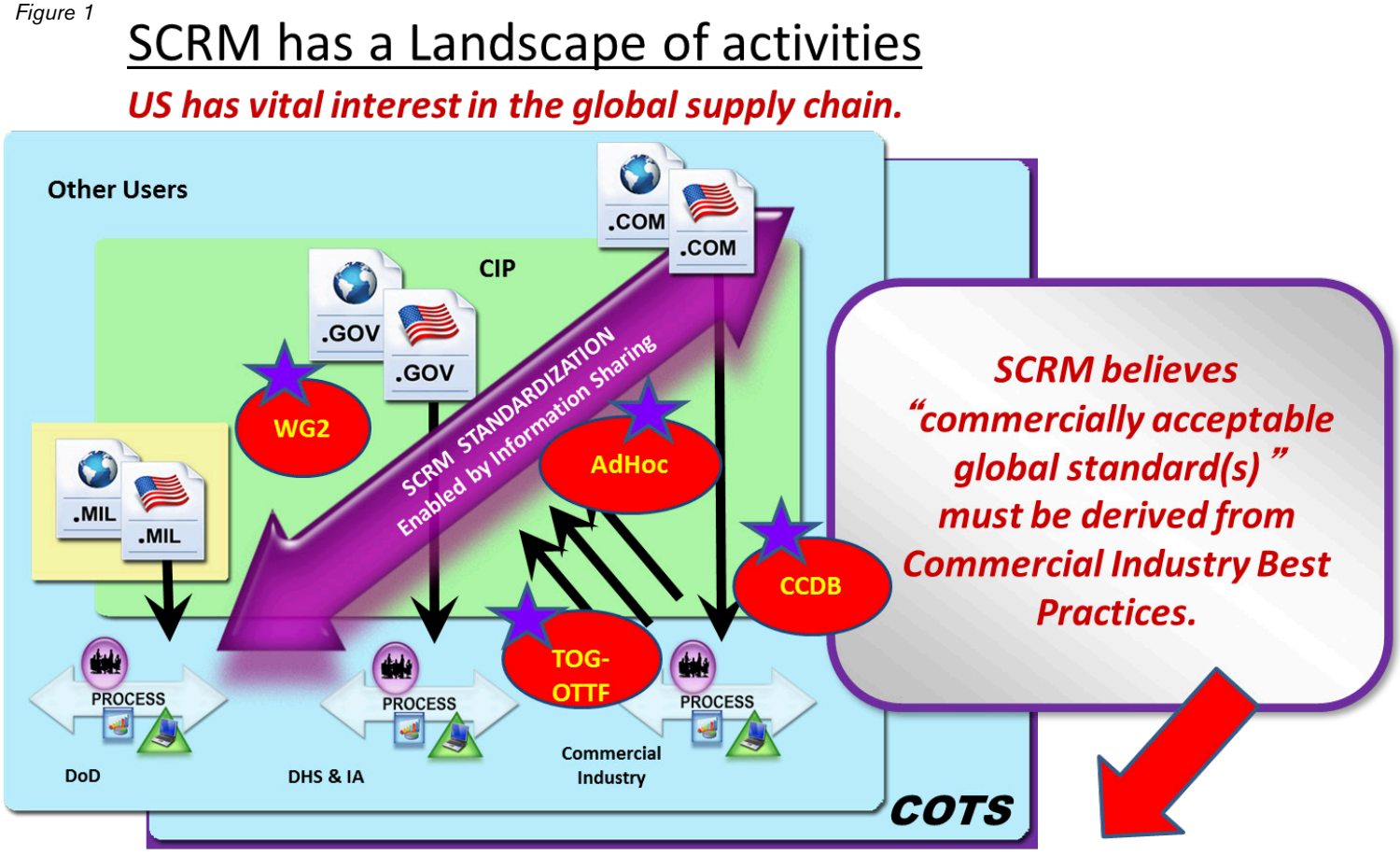
Federal acquirers and commercial acquirers and suppliers are all increasingly interconnected and interdependent in a global supply chain, both physically and digitally. We, in USG are not as independent as we used to be; we have fewer unique capabilities, systems and components. We all leverage an increasing number of COTS products, including hardware, software and services. However, our mission remains unique, and in the interest of national security and warfighter support, mission critical acquisitions need to be evaluated in terms of product integrity, mission assurance and SCRM best practices.

In this budget-conscious environment, there is no way to return to a supplier base of "all-American" companies for the U.S. Government's ICT acquisitions, nor can we have complete confidence that even American made products are free of supply chain vulnerabilities. Knowing what challenges we face by applying SCRM practices and guidance to our acquisition processes will help us tackle our next big challenge, which is to build weapons systems and information networks that are resilient against the most sophisticated cyber adversaries using mostly commercial and potentially untrustworthy products and services. This is both a sourcing and a systems engineering challenge.

Addressing the SCRM Challenge

GAO recently published GAO Report-12-361 Code 311064, "IT Supply Chain: National Security-related Agencies Need to Better Address Risks." They endorsed DoD SCRM strategy and implementation and recommended it as a model to others. The Committee on National Security Systems (CNSS) recently published CNSS Directive 505 on Supply Chain Risk Management. GAO said DoD's efforts to implement SCRM can be a learning tool for others in the Federal government. DoD is currently imple-

Figure 1



menting a strategy for achieving trusted systems and networks to address this challenge which has four key tenets: prioritizing resources based on mission dependence; comprehensive program protection planning; enhanced vulnerability detection, and industry partnership. This trusted systems and networks strategy is being implemented through existing Program Protection and Information Assurance processes through the recently published DoD policy DoDI 5200.44 – "Protection of Mission Critical Functions to Achieve Trusted Systems and Networks." It integrates existing disciplines of SCRM, system security engineering, counterintelligence, hardware and software assurance among others, to reduce the likelihood that warfighting capabilities will be impaired due to vulnerabilities in system design or sabotage of a system's critical functions and components. The policy builds on best practices, lessons learned, and evolving thinking from more than four years of piloting and incremental implementation within the Department by requiring specific program protection and SCRM activities to protect the most critical DoD systems. We continue to work across the Department and with our fellow interagency partners, our suppliers, and our system integrators to implement a risk management strategy into other government organizations (and their suppliers) and the country's wider Critical Infrastructure Protection initiatives.

As we develop better visibility into the global supply chain and improved trust in the products we consume or use we will be able to develop more resilient system designs, which will move us from a "risk response posture" to a more proactive, "risk prevention, risk mitigation, or even risk endurance posture."

In Figure 1, the large purple arrow highlights information sharing as the key to harmonizing SCRM efforts currently being addressed by different stakeholders, such as the civil government agencies, defense agencies and private industry. A number of active joint efforts and information sharing forums exist, as noted by the red circled items. The Open Group's Open Trusted Technology Forum is a collaborative effort between government and industry and is currently developing a framework of SCRM best practices for use by industry. The SCRM Lifecycle Processes and Standards Working Group meets almost monthly and is a DHS-DOD CNCI-11 (SCRM) effort co-chaired by DoD and NIST representatives and serves as an interagency sharing and collaboration venue. The Cyber Security 1 (CS1) ICT SCRM Ad Hoc group is comprised of civil and defense/government representatives and industry stakeholders. The primary focus is SCRM input and development of international standards, as CS1 supports the International Organization for Standardization (ISO). Finally, the Common Criteria Development Board (CCDB) is an ISO based effort supported by industry and government participation and is actively incorporating SCRM into the CC certifications for global use.

Working With Industry

Product development (from design through manufacturing, integration, and delivery) typically involves an array of developers and suppliers around the world, many of whom the end user

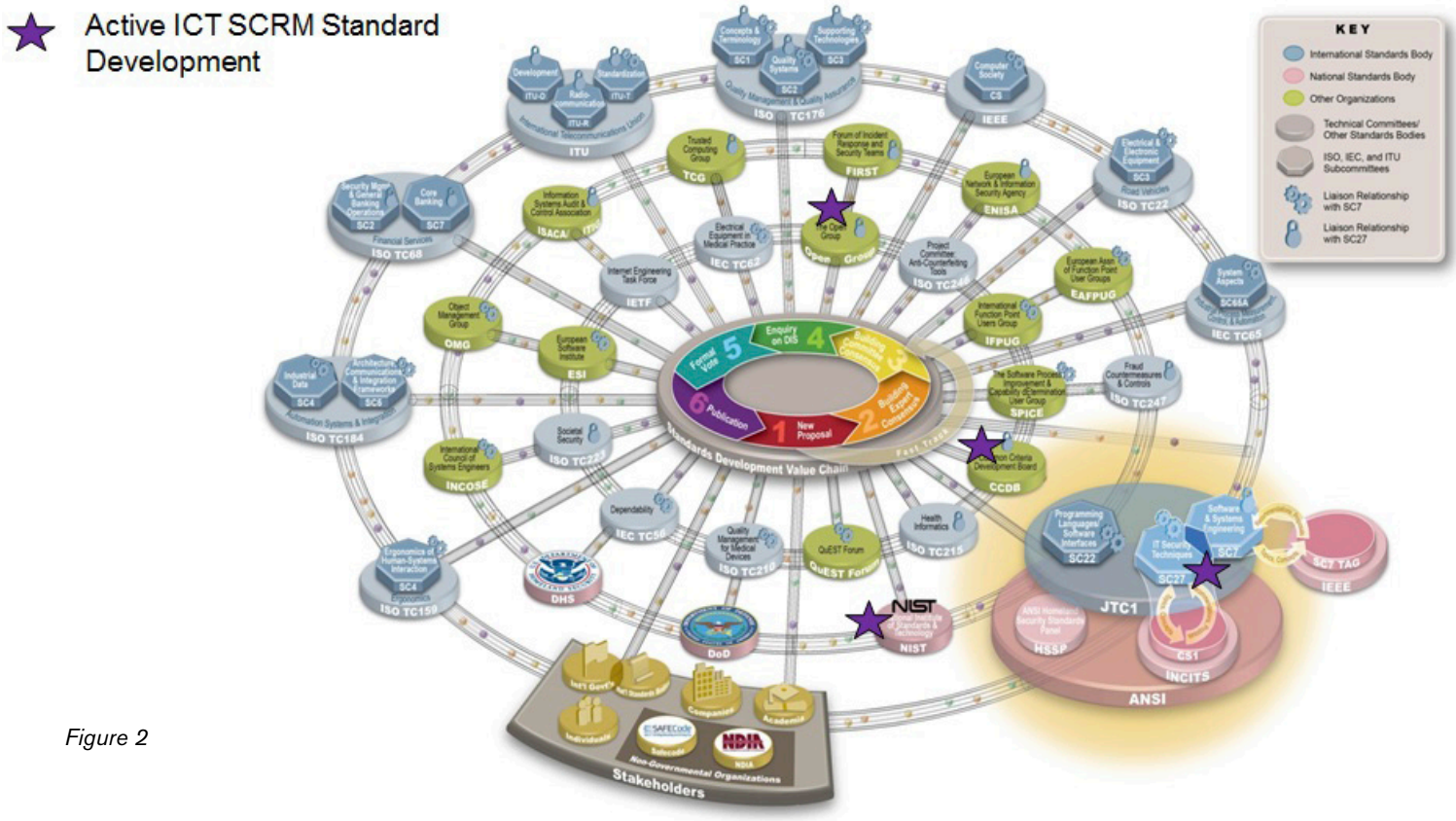


Figure 2

does not know. As a consequence of our global supply chain, adversaries have more opportunities to corrupt technologies before we take ownership and introduce malicious, tainted or counterfeit code or hardware into the supply chain. And even outside of the maliciously altered products, these incredibly complex, commercial products may at times have vulnerabilities unintentionally left in as they leave the product line. These vulnerabilities may be tolerable in cell phones and video games, but could prove catastrophic in a fighter jet or classified network as such vulnerabilities may make it easier for adversaries to use remote access attacks to otherwise gain access to the USG's systems and networks.

DoD has been working internally to enhance its acquisition, engineering, and sustainment processes, while simultaneously working externally with commercial industry to advocate improved product development standards to reduce vulnerabilities in commercial products related to global sourcing. The study of ICT SCRM standards landscape was completed in January 2010 in the form of a document and a key graphic provided in Figure 2.

The graphic and the corresponding Standards Landscape document are based on the portfolio of two international committees under the auspices of ISO/IEC JTC1 – SC 27 that focuses on IT Security Techniques and SC7 that focuses on System and Software Engineering. The graphic is color-coded as follows:

- Blue indicates Standards Development Organization (SDO) groups associated with ISO, IEC, or ITU
- Green indicates other SDOs
- Pink indicates US-based organizations including the Technical Advisory Groups for SC7 (SC7 TAG) and SC27 (CS1), their parent organizations (IEEE and ANSI), as well as US government agencies engaged in the development of ICT SCRM standards (NIST, DoD, DHS)
- Purple stars indicate specific SDOs currently engaging in the development of ICT SCRM content, both nationally and internationally, including SC27, SC7, The Open Group, CCDB, and NIST. Note these same starred areas are where DoD chose to engage with their information sharing activities.

The standards landscape identified a variety of groups that are engaging in the collection/development of ICT SCRM or related content and helped prioritize DOD engagement in these groups, as well as the areas of focus. Based on the outputs of landscape DOD has engaged with multiple stakeholders and continues identifying other potential stakeholder groups to facili-

tate information sharing.

The standards landscape review led DOD standardization activities towards specific SDOs to focus on standardization for ICT SCRM. The standards landscape also recommended relevant standards efforts within SC27 and SC7 for participation, influence, and monitoring based on the overall DOD engagement framework. DoD is actively working to coordinate external standards efforts with the DoD IT Standards Registry.

Based on the landscape DOD focused its standardization on CS1 and worked with CS1 to establish and Chair CS1 ICT SCRM Ad Hoc that is a joint group with SC7 TAG. The Ad Hoc is a non-voting group that has the authority to review SC27 and SC7 standards distributed to US National Bodies (CS1 and SC7 TAG) for review and comment, works to achieve consensus on a single position, and then recommends positions for vote and approval by CS1 or SC7 TAG as US positions to be submitted to SC27 or SC7.

As the efforts progressed, other areas of focus were identified including The Open Group, North American Security Products Organization, Information Security Forum, Object Management Group, Common Criteria / ISO15408 and SAE(G19), etc. Trusted Mission Systems and Networks continues identifying additional SDOs for potential collaboration through the current participation in various SDOs. These efforts were identified based on the inputs received from individual participants in the standardization processes, as well as to ensure that CS1/SCRM AdHoc WG references relevant documents that are either already in the standards domain or in the process of being developed.

Conclusion

The SCRM community/stakeholders know that change will not happen overnight and the implementation of this kind of comprehensive acquisition risk management for all of our systems and networks will take the investment of resources, time and funding. Therefore a key element of the SCRM strategy is to prioritize capabilities and their enabling systems and subcomponents; identify our critical systems and plan for and build in more trust, using a risk based approach.

In DoD we continually seek to improve our capabilities and cyber posture; improving our capability to detect cyber problems in our day-to-day operations, but that still puts us in a "risk response posture"; we need to better understand the components within our systems that enable our mission critical capabilities (we call this criticality analysis); where do we source the critical hardware, software, and services for those systems (especially national security systems and critical infrastructure), and how should we better design and manage our systems to minimize vulnerabilities and assure critical functions, even when a system is under attack. Understanding and managing the risk associated with those systems and their components, will make us and our systems more resilient.

Recently, there has been a lot of news on microelectronic counterfeits, malicious or poor quality software and data breaches. All of these topics have roots in our global supply chain. Do not misunderstand our intent, this is not about becoming isolationists—DoD embraces globalization and will continue to reap cost and schedule benefits from it every day—but we do need to be more sensitive to the system and/or information security and product and/or data integrity implications, to our systems and ultimately our capabilities, when outsourcing key components and capabilities. We need to better "see" into some legs of the supply chain, especially where critical components are involved.

DoD is doing well in our strategy and implementation on SCRM, however we are developing capability through a "crawl-walk-run" process which has dependencies on potentially diminishing resources and external support, like private sector cooperation.

For additional information or to get involved in SCRM efforts, contacts are listed in Table 1.

Stakeholder Audience	Ongoing Effort	Points Of Contact
Department of Defense	Trusted Systems and Networks Round Table	Joe Wassel – joe.wassel@osd.mil Melinda Reed – melinda.reed@osd.mil
Interagency Coordination	CNCI SCRM Working Group 2	Don Davidson – don.davidson@osd.mi Jon Boyens – jon.boyens@nist.gov
Critical Infrastructure Protection	DHS SCRM DHS Software Assurance	Joe Jarzombek – joe.jarzombek@hq.dhs.gov
ISO Standards and Harmonization	CS1 ICT SCRM Ad hoc	Don Davidson – don.davidson@osd.mil Nadya Bartol – nadya.bartol@utc.org

Table 1: SCRM Effort Contacts

ABOUT THE AUTHORS

Don Davidson is assigned to Trusted Mission Systems and Networks in the Office of the Department of Defense Chief Information Officer (DoD CIO), as Chief, Outreach, Science, and Standards (CNCI-SCRM). He has 37 years of federal service to include 11 years active duty, as well as civilian assignments in Army Research Laboratory, Army Materiel Command, Army Secretariat, US Joint Forces Command, OUSD-Acquisition, Technology & Logistics (AT&L), and DoD CIO.

E-mail: Don.Davidson@osd.mil

Stephanie Shankles of Booz Allen Hamilton, is a subject matter expert in software assurance and ICT supply chain risk management. She supports projects ranging from IT policy development to IT security training to helping clients integrate security processes throughout their project lifecycle. She is currently supporting industry efforts to develop and implement ICT supply chain risk management guidelines and standards. She has spoken at multiple industry events on software assurance implementation, benchmarking and measurement.

E-mail: shankles_stephanie@bah.com

Managing Risk in the Software Supply Chain Through Software Code Governance

Kristin Brennan, Coverity

Abstract. With the increasing complexity of software applications, shrinking IT budgets and the spiraling cost of developing software, many organizations in both the public and private sectors are turning to third-party software suppliers including outsourced teams, partners and open source to develop their applications. According to a recent study conducted by Forrester Consulting and Coverity [1], almost all organizations are using some form of third-party code in their products, and over 40% rely on software from three to five different software suppliers.

The use of COTS tools in military environments is no longer limited to hardware. COTS software is increasingly making its way into military platforms. In systems where the use of existing commercial components is both possible and feasible, it is no longer economically feasible for the government to specify, build, and maintain a large array of comparable proprietary products.

However, commercial third-party code is typically not tested with the same level of rigor as internally developed code. As software complexity grows, additional software capabilities bring many more lines of code, and greater opportunity for error. That means a defect could be lurking in the third-party code that could cause a significant breach or security issue.

Organizations are recognizing the need for end-to-end accountability for the quality and security of the code in their products, regardless of who actually created the code. There is a need for efficient processes to enforce consistent software code governance across the software supply chain.

Software Code Governance

The initial focus of software code governance was to assure software quality and security of in-house developed code by establishing clear guidelines and procedures such as the FDA's recommendation that infusion devices be tested with static analysis and DO-178C, Software Considerations in Airborne Systems and Equipment Certification for the avionics industry. Today, we see software code governance gaining momentum in a wide variety of industries as organizations seek to drive greater accountability and efficiency within distributed development teams and to achieve better visibility and control over third-party code.

A Multi-Step Process

Software code governance cannot be achieved with the click of a button. It is a process that needs to be embraced by the organization and enforced across the internal and external supply chain. The process will vary by organization based upon whether you are trying to establish governance across internal teams, with outsourcers, offshore development teams, or partners, and whether you have access to source code for the application.

Establish Acceptance Criteria

Automated code testing solutions enable managers to establish and enforce consistent measures for quality and security across the software supply chain. Organizations can use automated code testing to establish acceptance criteria with their suppliers. For example, it could be mandated in the contract that all code must be tested with static analysis. Static analysis testing produces results that are repeatable, measurable and objective. To support static analysis testing, policies can be automatically established to ensure that there are no uninspected defects and no high impact quality and security defects in the code. A strict acceptance criteria can be established so that all found defects must be addressed before the code is accepted. This approach puts the onus on the software supplier to ensure their code is of high enough quality to pass the established acceptance criteria and would be a practical solution in situations where you do not have access to source code.

Auditing Mode

Another approach to ensuring quality and security across the supply chain is to establish auditing rights with suppliers. Organizations that purchase source code can reserve the right to analyze the supplier's code and report back results. This could be implemented as part of the integration phase of the lifecycle. This auditing right helps the organization measure quality in a consistent manner across their supply chain and with their internal teams. It also enables the organization to provide recommendations and results of the analysis back to the supplier giving them an opportunity to fix the defects. Once a baseline for quality and security has been established with a supplier, a policy can be enforced that no new defects are allowed as new defects could introduce risk into the overall project.

Self-Certification

Organizations who are supplying code can also be encouraged to take proactive measures to "self-certify" the quality of their code before delivering it. NNG, a pioneer of navigation software and the developer of iGO Navigation solutions has adopted such an approach in the private sector. It has deployed static analysis to deliver high quality software and accelerate time-to-market for software delivery to its supply chain. NNG has delivered navigation solutions to more than 150 business customers including the world's leading original equipment manufacturers. Its navigation software is at the heart of millions of products from in vehicle infotainment systems and smartphones to personal navigation devices.

NNG has embedded static analysis into their development process so every new line of code is tested before it is released into the market. It enables them to track and manage defects between 28 projects and different code branches comprising over 1 million lines of code. As a result of development testing, NNG has been able to establish standardized metrics for measuring software quality across the supply chain and remove cost and complexity from its own software development activities.

Conclusion

Establishing and enforcing acceptance criteria and negotiating the right to audit the software quality are two concrete steps organizations can take to maintain the highest levels of quality across their software supply chain. As software and supply chains continue to become more complicated, and organizations continue to deliver more innovation, and at the lowest cost possible, the ability to enforce consistent standards for quality will become increasingly important.

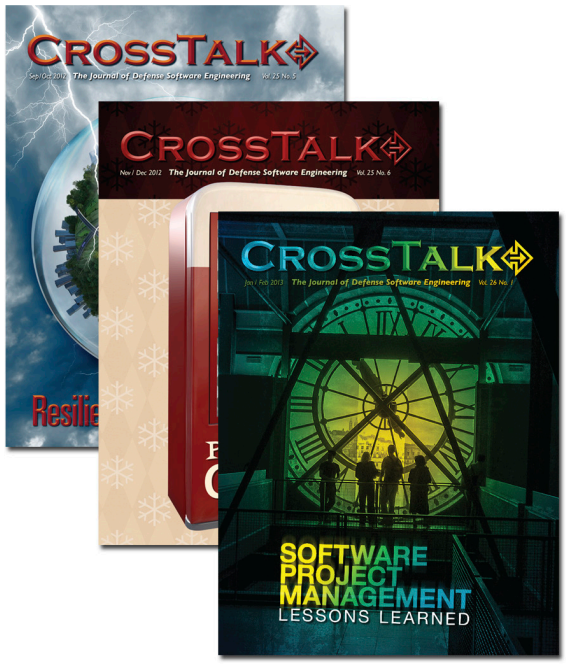
ABOUT THE AUTHOR

Kristin Brennan, Senior Director of Product Marketing, Coverity, has more than 15 years of technology marketing experiences with expertise in development testing, static analysis, code governance, and automation. Prior to joining Coverity, she help senior marketing positions at Hewlett Packard and Wells Fargo Bank. She has a BA from UC Irvine and MBA from UC Berkeley.

Phone: 415-321-5236
E-mail: kbrennan@coverity.com

REFERENCES

1. Forrester Consulting. Software Integrity Risk Report, 2012



How International Standard Efforts Help Address Challenges in Today's Global ICT Marketplace

Stephanie Shankles, Booz Allen Hamilton
Michele Moss, Booz Allen Hamilton
Jed Pickel, Microsoft's Trustworthy Computing group
Nadya Bartol, Utilities Telecom Council

Abstract. An increasingly distributed and global Information and Communication Technology (ICT) supply chain brings challenges to U.S. Government and industry. Identifying and mitigating risks involves looking beyond your organization and understanding and managing risks caused by the lack of visibility in the ICT supply chain. Recent research indicates that current ICT supply chain risk management practices tend to have a tactical focus motivated primarily by compliance rather than a strategic integrated approach. However, there are a number of existing international standards and several under development that when used in combination will help this problem. Using these standards together will provide a security assurance process for information security governance, software development, Supply Chain Risk Management (SCRM), and should result in reducing ICT supply chain risk.

Challenges in Today's Global ICT Marketplace

ICT supply chain risk management covers both software and hardware. Several recent industry reports focused primarily on software and on the general state of information security provide insights into current ICT supply chain practices, and what motivates their selection:

- Software Security: Think Big, Start with What Matters, 2009 The Burton Group [1].
- Cyber Supply Chain Security and Software Assurance Research Report, 2011 Enterprise Strategy Group [2].
- Borderless Security: Global Information Security Survey, 2010 Ernst and Young [3].
- State of Application Security, 2011 Forrester Consulting [4].
- Global Information Security Workforce Study, 2011 Frost & Sullivan and (ISC)2 [5].
- Software Integrity Controls, 2010 SAFECode [6].
- Assessing SCRM Capabilities and Perspectives of The IT Vendor Community: Toward a Cyber-Supply Chain Code of Practice, 2010 University of Maryland [7].
- Verizon and Secret Service (USSS) - Data Breach Investigations Report, 2010 and 2011 [8][9].

Together, these reports present the following key findings in Table 1.

Although each report is focused on different aspects of information security and therefore touches on different aspects of ICT supply chain security, they share a common message that holistic processes are needed to mitigate risks. Table 2 summarizes the results of three studies, which while conducted at a different level of detail, presented similar conclusions.

It is evident from the studies' results that to advance the state of the software security practice, stakeholders across an organization will need to bridge the communication gap with the purpose of effectively balancing the executive priorities and the implementation of operational, technical, and management practices for software security and supply chain.

The Enterprise Strategy Group study [2] identified that over 40 % of the surveyed organizations trust their developers to know how to develop secure software. Several key trends emerged from this and other studies appear:

1. Only 47% of acquirers are performing acceptance testing of third party code. As a result, vulnerabilities in the code are not identified until the code is in production and organizations that acquired this software are left with the consequences. While the problem originated in the software supply chain, the acquirers have to address the risk.

2. Compliance requirements to run scans of the operations environment result in the identification of code level vulnerabilities. Subsequently, the realization of insecure coding practices is not identified as an issue until the operations and maintenance phase of the lifecycle, when it is more difficult and costly to fix. Again, the problem that originated in the software supply chain is left up to the acquirer to address.

3. Shifting from responding to vulnerabilities identified during operations to preventative practices in technology acquisition and development takes time and effort (potentially increasing time to deliver and cost of initial product—even though, this cost has been demonstrated to be less than the lifecycle sustainment costs when fixed later). With 46% of respondents using a development method, the foundation is in place for a coordinated approach to maintaining legacy technology and minimizing the impact of security incidents.

ICT SCRM Practices

The University of Maryland (UMD) and the Enterprise Strategy Group conducted studies focused on procedures that organizations use to manage security and risk in their supply chains for ICT products and services. Supply chain risk management is one of the initiatives in the United States Comprehensive National Cyber Security initiative. As such, it has been the focus of discussion and study by a number of organizations. However, the practice of securing ICT supply chains is still in its infancy and is often a misinterpreted problem. The September 2011 article, "Renewable Industry in Turmoil Latest Sign: American Superconductor Accuses Chinese Firm—Its Biggest

Key Findings	
What ICT supply chain practices organizations currently employ?	▪ Security practices surrounding software development are tactical in nature and do not address software security risks in a strategic manner. ▪ Supply chain risk management practices are tactical and are not addressed in a strategic manner.
What informs and motivates the selection of ICT supply chain processes?	▪ The largest motivator for secure software development practices is compliance ▪ Lack of leadership support and resistance to changing existing software development practices is a barrier to the adoption of secure development methodologies. ▪ To advance the state of the software security practice, stakeholders across an organization will need to bridge the communication gap to effectively balance the executive priorities and the resources needed for the implementation of operational, technical, and management practices for software security and supply chain. ▪ 40% of organizations surveyed do not employ a secure software program because they trust their developers know how to develop secure software and/or don't believe they have a security issue.

Table 1:

VDBR (Verizon)	Executive Perspective (Ernst & Young)	Security Professional Perspective (Frost)	Common Objective
▪ 96% of breaches were avoidable through simple of intermediate controls ▪ Approximately half of the breaches utilized hacking or malware	▪ Increase in risk due to social networking, cloud computing and personal devices in the enterprise ▪ Data leakage is the primary concern for organizations	▪ New technologies are boing deployed without adequate security ▪ Application vulnerabilities are the #1 threat to organizations	▪ Implement essential security practices (such as access control, network management, secure development, and log management/analysis) to mitigate the risk of a data breach and loss of sensitive data

Table 2:

Customer—of Espionage" in the Wall Street Journal highlights [10] several facets of the supply chain challenge that involved a prominent American wind turbine manufacturer. Proprietary software was stolen and given to a major Chinese competitor. An affiliate of the competitor was a major Chinese parts supplier to the American manufacturer Superconductor. It appears that the American manufacturer's supplier was intentionally providing them with faulty parts and components. Not only was American Superconductor a victim of a malicious insider, they were also the victim of supply chain tampering. This challenge will grow as large and small organizations operate in increasingly complicated and globally dispersed supply chains.

It appears that the American manufacturer's supplier was intentionally providing them with faulty parts and components. Not only was American Superconductor a victim of a malicious insider, they were also the victim of supply chain tampering. This challenge will grow as large and small organizations operate in increasingly complicated and globally dispersed supply chains.

"American Superconductor Accuses Chinese Firm—Its Biggest Customer—of Espionage"

INCIDENT:
An employee of American Superconductor Corporation working in Austria is charged with stealing and modifying valuable software that controls turbines. The employee gave the software to Sinoval Wind Corporation, the Chinese company that is under contract with American Superconductor. Sinvolal then passed it on to an affiliate, a fierce competitor of American Superconductor, who then repackaged it and sold it back to Sinoval at a very low cost.

IMPACT:
The stolen software has already appeared in turbines sold by Sinvoel in China. It has given the Chinese valuable intellectual property and may lead to huge financial losses for American Superconductor. The company has gone from $30 a share early in the year to about $6 a share in September, falling sharply on the news of legal action.

MITIGATION:
Austrian authorities confirmed that they have arrested the employee and he is cooperating. American Superconductor is seeking criminal redress through Chinese and Austrian courts and the Beijing Arbitration Commission. They claim they are owed $250 million for breach of contract claim.

http://online.wsj.com/article/SB10001424053111903374004576578971052370768.html

Figure 1

Common Development Practice:
The RuggedCom Operating System (ROS) contains a vulnerability that could allow an authenticated, remote attacker to access sensitive information on a targeted device. The vulnerability exists because the RSA private key infrastructure (PKI) for SSL communications between an affected device and the user is hard-coded (visible) in the affected software. An authenticated, remote attacker could exploit this vulnerability by reverse engineering the key from the affected software.

Operational Impact:
The vulnerability was discovered in smart grid networking devices produced by industrial networking equipment manufacturer RuggedCom. RuggedCom manufactures ethernet switches, network routers, wireless devices, serial servers, media converters and other communications equipment for use in harsh electrical and climatic environments like those found in electrical power substations, oil refineries, military installations or roadside traffic control cabinets.

Risk:
The vulnerability allows an attacker to decrypt any SSL-based communications between an end-user's web browser and the RuggedCom device. In addition, the hard-coded private key could be used in a man-in-the-middle (MITM) attack to impersonate a RuggedCom device undetected. A successful exploit could lead to the loss of system integrity and lead to additional attacks. Attackers could access power station communications, intercept sensitive data and communications, and even potentially take control of critical infrastructure systems.

RUGGEDCOM
INDUSTRIAL STRENGTH NETWORKS™

- http://www.computerworld.com/s/article/9230516/ICS_CERT_warns_of_SSL_security_flaw_in_RuggedCom_industrial_networking_devices
- http://tools.cisco.com/security/center/viewAlert.x?alertId=26713
- http://www.smartplanet.com/blog/thinking-tech/new-security-flaw-uncovered-in-smart-grid-gear/12823

Source: Don Davidson, DOD-CIO Trusted Mission Systems and Networks

Figure 2

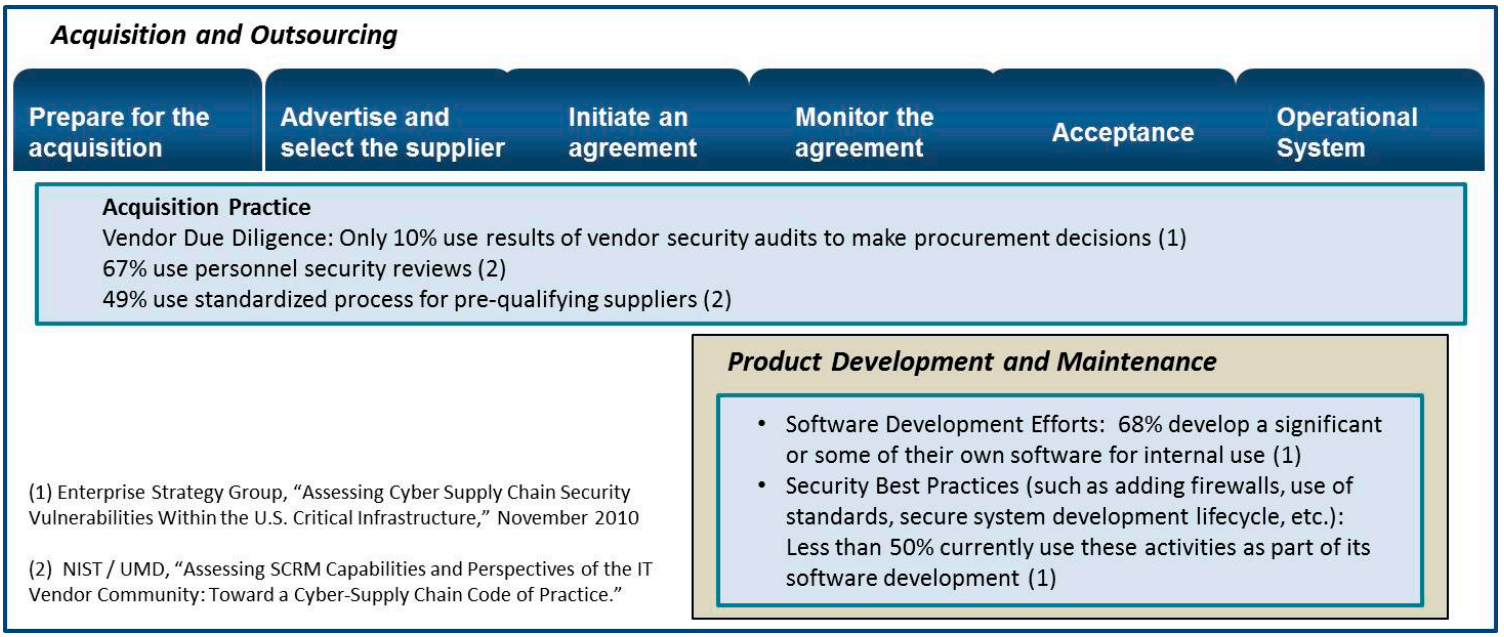


Figure 3

Similar to the security practices surrounding software development, currently used ICT supply chain risk management practices are also tactical and are not addressed in a strategic manner. For example, procedures that allow organizations to understand activities of suppliers, such as auditing vendor practices, are rarely used. When audits are used, organizations rarely allow those results to influence procurement decisions.

In addition to identifying that about 40% of the organizations surveyed do not employ a secure software program because they trust their developers know how to develop secure software and/or do not believe they have a security issue, the Enterprise Strategy Group study [2] also indicated that most development efforts are not employing essential security practices. The Verizon Data Breach Report identified similar issues.

According to the UMD study [7], there is a divide between small and large companies with regard to employing security measures and small companies are falling behind. However, the study findings suggest that incentives can lead to positive changes in this area. UMD research indicates that smaller organizations are highly motivated to use government cyber-SCRM practice guidelines. Many of the smaller organizations view this as an opportunity to gain acceptance into the federal acquirer community. Likewise, larger organizations view SCRM practice guides as a means of differentiating themselves by making these practices a "condition of membership" in a premier industry organization.

International Standards Environment and ICT SCRM

The issues and challenges identified in the reports can be addressed by applying several existing and emerging international standards. Figure 4 illustrates the relationship between existing and emerging ISO standards that provide the framework for addressing ICT SCRM concerns evident from the various studies.

There is a divide between small and large companies with regard to employing security measures and small companies are falling behind. However, the study findings suggest that incentives can lead to positive changes in this area. UMD research indicates that smaller organizations are highly motivated to use government cyber-SCRM practice guidelines.

The Overview layer in the figure depicts three overview standards that address the overall information security management (ISO/IEC 27000), information security in supplier relationships (ISO/IEC 27036-1), and application security (ISO/IEC) 27034-1. Collectively these standards provide the fundamentals and vocabularies for these three disciplines. The Requirements layer depicts the two relevant requirements standards. ISO/IEC 27001 provides requirements for managing information security for the enterprise using a risk-based approach. ISO/IEC 27036-2 provides requirements to be used to protect enterprise information when working with suppliers or acquirers. Finally, the Guidance layer depicts the guidance standards associated with the requirements standards above. ISO/IEC 15288/12207 (Systems and software engineering—System lifecycle processes and Systems and software engineering—Software lifecycle processes) acknowledges integration of both ISO/IEC 27036 and ISO/IEC 27034 with system and software engineering standards. ISO/IEC 27036-3 provides specific guidance on ICT supply chain security in addition to the requirements in ISO/IEC 27036-2. ISO/IEC 27002 provides guidance for implementing security controls selected as a result of a risk assessment required by 27001. It should be noted that of these standards, ISO/IEC 27036 is in draft, while the rest are published standards. ISO/IEC 27001 and 27002 are currently under revision. These standards provide processes, controls, and practices for resolving many of the issues identified earlier.

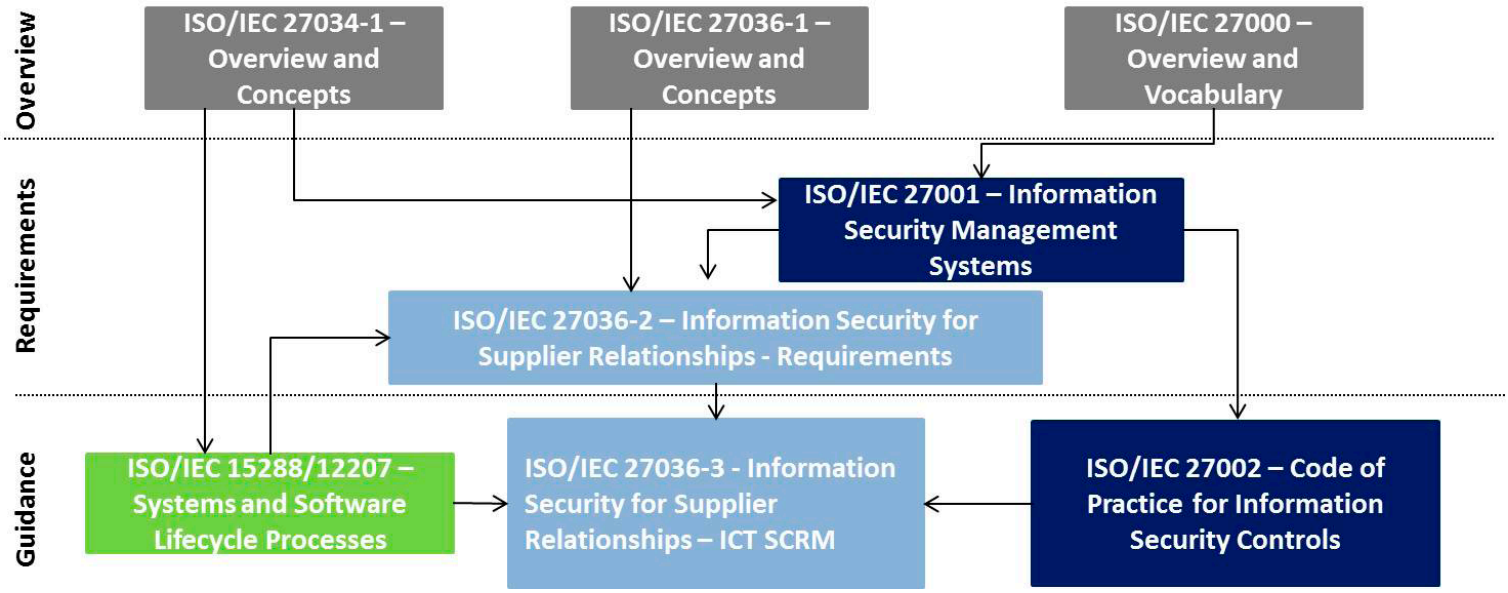


Figure 4

Information Security Management Governance

The ISO/IEC 27000 family of standards provides a number of standards for establishing and implementing an information security management system. Specifically, ISO/IEC 27001, Information Security Management System Requirements, provides a governance framework for information security. Implementing this framework will help gain leadership support for approaching the challenges of today's global marketplace such as implementing appropriate operational, technical, and management practices for software security and supply chain.

ICT Supply Chain

While there are a number of published standards that can help organizations manage information security and associated risks, none of those currently published standards provides guidance on how to protect an organization's information security interests in a relationship between acquirers and suppliers. ISO/IEC 27036 which is currently in draft, provides an approach for protecting sensitive enterprise data within the context of acquiring and supplying products and services. This multipart standard covers managing the information security aspects of a portfolio of supplier relationships, as well guidance for how to manage individual supplier relationships. The standard provides requirements that cover a broad variety of products and services, as well as context-specific guidance. Specifically, Part 3 focuses on ICT supply chain security.

The standard introduces a number of requirements and concepts that while not new in supply chain and sources contexts, are new in the information security context:

- Having a registry (inventory) of all suppliers.
- Assigning responsible individuals to manage information security aspects of relationships with each supplier.
- Assessing the criticality of such relationships and associated risks and using this criticality to prioritize supplier relationships and associated security requirements.
- Having a minimal set of information security requirements applicable to any supplier relationship.
- Monitoring the information security aspects of supplier relationships.
- Ensuring protection of data and information when terminating those relationships.

Software Development Security Practices

Secure software development practices are important to supply chain security risk management efforts. The Forrester study referenced earlier in this article provides a good basis for understanding potential risks from software development process gaps. Results of the study are concisely summarized in the statement: "While a majority of organizations have implemented some form of application security measures, very few have put in place an end-to-end strategic approach that incorporates security throughout the software development lifecycle." The supply chain implication is that reviewing a vendor's secure development practices is an important step in managing supply chain risks.

This raises several important questions, such as: what a secure development process should include, how an organization should manage that process, and how a vendor's process should be evaluated? An obvious start is to ensure that a vendor has a secure development process, that it incorporates techniques that address real-world security threats, and that the vendor's organization is clearly committed to supporting that process. But what is the right approach to creating such a process? And what else should be considered? In November 2011 the International Standards Organization published part 1 of ISO 27034, an internationally recognized application security standard that may help simplify the answers to those questions. Currently Part 1: Overview and concepts is published and latter parts are still in development.

ISO 27034-1 provides frameworks and a process that can help inform a vendor's approach to build and operate a comprehensive application security program. The standard can

also help an organization validate and identify gaps within their current application security program. Additionally, the standard can help an organization implement aspects of ISO 27001 via the systematic approach to risk management shared by the standards. ISO 27034-1 includes an annex that demonstrates how an existing development process based on the Microsoft Security Development Lifecycle aligns to ISO 27034. This may help simplify an organization's efforts to implement the standard.

An organization that has reviewed and is considering adoption of ISO 27034-1 is likely to be taking a strategic approach to software security and be applying relevant application security controls through all phases of their software development lifecycle. Consequently, ISO 27034 may be a helpful tool to simplify the process of managing supply chain risks by providing a standards based approach for understanding if vendors in your supply chain are taking a strategic and holistic approach to software security.

Conclusion

Modern supply chains have introduced greater risks to organizations. Globalization and the proliferation of technology around the globe have presented new significant threats to national security, economic security and protection of intellectual property (investment). The combination of international standards efforts described in this paper will help to provide a solid foundation for organizations to integrate an organizationally driven, risk based implementation of ICT SCRM practices.

Acknowledgements:

We wish to thank Don Davidson of the DoD for encouraging us to write this article and Ken Lyle of Booz Allen Hamilton for his assistance with analyzing the studies and identifying examples to include in the article.

REFERENCES

1. Ramon Krikken, "Software Security: Think Big, Start with What Matters", Burton Group, June 2009.
2. Jon Oltsik, "Assessing Cyber Supply Chain Security Vulnerabilities Within the U.S. Critical Infrastructure", Enterprise Strategy Group, November 2010.
3. "Borderless Security: Ernst & Young's Global Information Security Survey", Ernst & Young, February 2010.
4. "State of Application Security: Immature Practices Fuel Inefficiencies, But Positive ROI IS Attainable", A Forrester Consulting Thought Leadership Paper Commissioned By Microsoft, January 2011.
5. "The 2011 (ISC)2 Global Information Security Workforce Study", A Frost & Sullivan Market Survey Sponsored by (ISC)2, 2011.
6. "Software Integrity Controls: An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain", SAFECode, June 2012.
7. "Assessing SCRM Capabilities and Perspectives of the IT Vendor Community: Toward a Cyber-Supply Chain Code of Practice", A NIST-Sponsored Project Conducted by The Supply Chain Management Center Robert H. Smith School of Business University of Maryland College Park, 2010.
8. "2010 Data Breach Investigations Report", Verizon Corporation, July 2010.
9. "2011 Data Breach Investigations Report", Verizon Corporation, April 2011.
10. Smith, Rebecca. "Renewable Industry in Turmoil: Latest Sign: American Superconductor Accuses Chinese Firm–Its Biggest Customer–of Espionage." The Wall Street Journal, 19 Sept. 2011.

ABOUT THE AUTHORS

Stephanie Shankles of Booz Allen Hamilton, is a subject matter expert in software assurance and ICT supply chain risk management. She supports projects ranging from IT policy development to IT security training to helping clients integrate security processes throughout their project lifecycle. She is currently supporting industry efforts to develop and implement ICT supply chain risk management guidelines and standards. She has spoken at multiple industry events on software assurance implementation, benchmarking and measurement.

E-mail: shankles_stephanie@bah.com

Michele Moss of Booz Allen Hamilton, is a recognized thought leader in the integration and benchmarking of assurance practices. She is co-chair of the DHS Software Assurance Working Group on Processes & Practices. She represents Booz Allen within the U.S. International Committee for Information Technology Standards Cyber Security 1 (CS1) technical committee and the U.S. Technical Advisory Group (TAG) for ISO/IEC JTC1/SC7. She is the liaison from SC7 TAG to CS1.

E-mail: moss_michele@bah.com

Jed Pickel is a senior security program manager in Microsoft's Trustworthy Computing group. Jed is focused on alignment of Microsoft's Security Development Lifecycle with international security standards and sharing Microsoft SDL best practices with the software development ecosystem. Jed's 15 years as a security professional started at the CERT Coordination Center as member of the technical staff. He has since been working in a variety of security focused roles at Microsoft.

E-mail: jpickel@microsoft.com

Nadya Bartol, of Utilities Telecom Council, is a U.S. technical expert working on the ISO/IEC 27000 series standards and Project Editor for ISO/IEC 27036. In her role at UTC, she is responsible for creating a cybersecurity information sharing platform for the utilities industry to deliver practical solutions to emerging cyber challenges. Prior to UTC, Ms. Bartol led numerous strategic groundbreaking cyber security engagements for Federal government and commercial clients for Booz Allen Hamilton.

E-mail: nadya.bartol@utc.org

Open Source and the Software Supply Chain

A Look at Risks vs. Rewards

Wayne Jackson, Sonatype

Abstract. There is a dynamic shift occurring in the software development landscape. No longer are applications written, today most are assembled using open source components. The growing reliance on externally sourced, open-source components as core building blocks for modern application development, coupled with the complexity of the ecosystem, has ushered in new risks for the software supply chain.

This article will explore the licensing, security, and quality risks associated with component-based development and its direct impact on the integrity of the software supply chain.

Introduction

For most of its history, software has been written—applications consisted primarily of custom developed code and internally developed components with only a small fraction of code sourced from outside the organization. Development efforts followed a "waterfall" methodology and projects spanned months or even years. The widespread use of cloud-based infrastructures and the rise of open-source technologies during the past decade have heavily influenced the software development landscape with startups and established organizations demanding increased flexibility and improved time to value in the way software is developed and delivered. As a result, modern software development and the resulting software supply chain have become increasingly component-based, where applications are assembled from existing components rather than written from scratch. Enterprise applications today are typically built using 75% to 80% open source components [1], with custom code comprising the rest. So, what does today's software development landscape look like and what are the risks to the software supply chain?

Software Development Once Was…	Software Development Now Is…
Waterfall Methodology	Agile Development
Code-Based	Component-Based
Developed	Assembled
Independent	Collaborative
Proprietary	Open Source

Table 1:

First, modern software development is increasingly component-based. The vast majority of these components are sourced from outside the organization.

Second, open source has become an integral part of modern applications. In most cases, externally sourced components are open source. Modern applications often rely on hundreds of open source components and frameworks.

Third, development organizations have embraced agile software development processes. The modern development process is rapid, continuous, and collaborative.

While development teams have embraced agile software development processes, the shifting software development landscape has also introduced new risks and requirements in the software supply chain. Applications can be composed of hundreds of components sourced from a myriad of open-source projects and these components can in turn, depend on other components, known as transitive dependencies. This creates an enormously complex software supply chain, where a single application may contain components originally published by dozens of individual projects.

To see just how far reaching externally sourced open source components are in the software supply chain, look no further than the Central Repository, the industry's primary source for open source components. The Central Repository receives 7.5 billion requests annually and is used by more than 60,000 organizations worldwide. Large organizations that rely on custom software for competitive advantage are the biggest consumers of the 400,000 components in the repository, but demand comes from every industry and geography.

Whether provided by commercial vendors or open source initiatives, components can introduce significant management, security and licensing challenges. Think of today's software as being assembled rapidly with a very complex supply chain like that of a car manufacturer. Like a car, the final product (an application) may contain hundreds or thousands of externally sourced components from dozens or hundreds of original suppliers. Each of these components has its own lifecycle, its own bug fixes and feature enhancements, and its own potential risks.

Like a car, a single flawed component could cause significant problems for the user. In the worst case, these problems could lead to security breaches, data leaks, stability, and performance issues, or legal actions related to intellectual property.

An easy example of a potential problem is in the area of security. Recent analysis by Aspect Security, using data from the Central Repository, uncovered widespread security vulnerabilities among the most commonly used open source components.

Often risks are caused by flawed components nested deep in an application's dependency tree where flaws are not easily apparent. Dependencies may allow flawed components to quickly infiltrate and undermine the software supply chain.

Users of the Central Repository regularly consume outdated, flawed or insecure components even years after newer fixed versions are available. An example of this can be seen when the United States Computer Emergency Readiness Team and NIST issued a warning in March 2009 that the Legion of the Bouncy Castle Java Cryptography API artifact was extremely vulnerable to remote attacks. Almost two years later in January 2011, more than 1,500 organizations downloaded the vulnerable version of Bouncy Castle from the Central Repository in a single month [3].

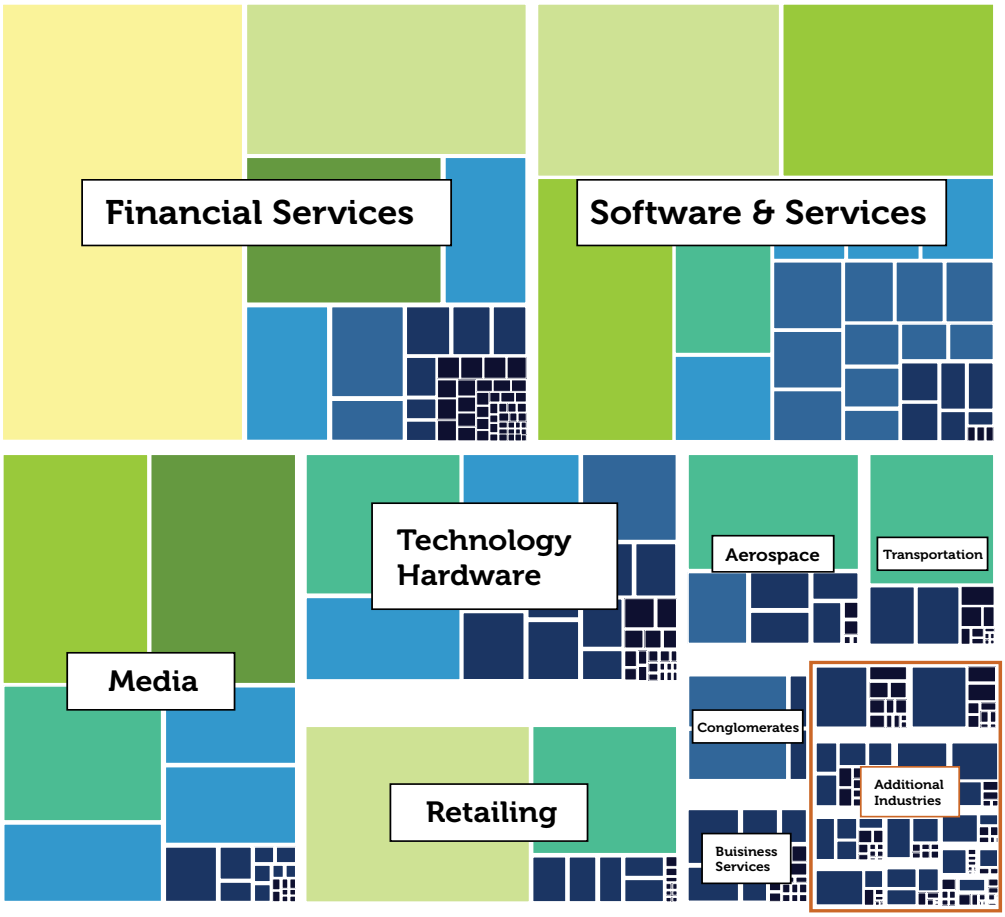


Figure 1:

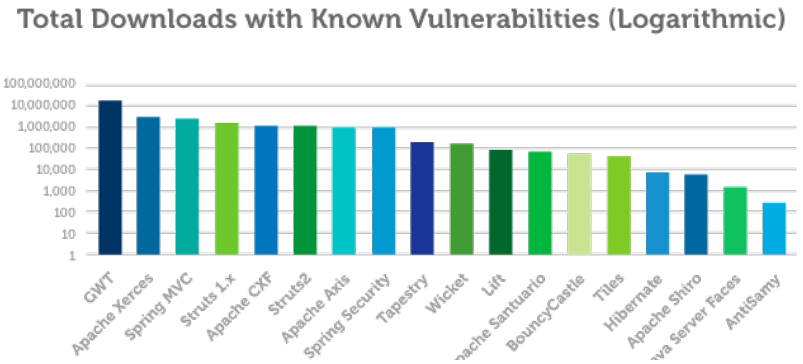


Figure 2: From reference [1]

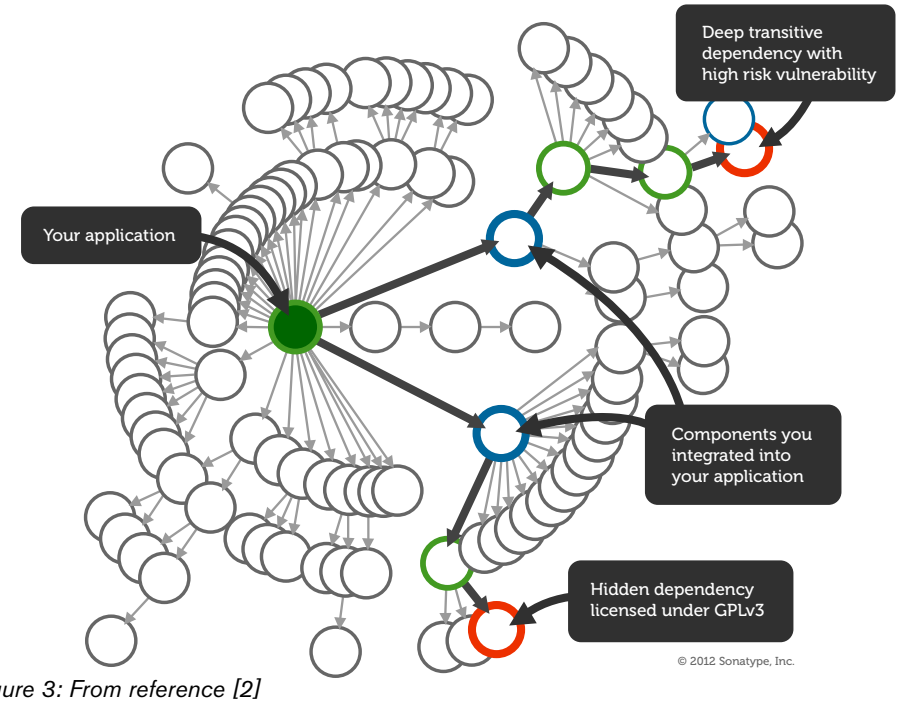


Figure 3: From reference [2]

Open source projects innovate rapidly and release frequently. However there is no update notification infrastructure for open source components. Therefore there is no easy way for component consumers to know when a new version has been released, much less which defects have been fixed.

Because open source usage generally occurs under the corporate radar, it is not uncommon for organizations to be unaware of which components are being used in their software supply chain or within key production applications.

Agile software development, incremental deployment and continuous integration have all resulted in many more builds over the life of a software project. Measurements of software quality and risks must be conducted in-band during the development and build process. Development teams are increasingly geographically dispersed and often include external contractors. Keeping disparate teams in sync and enforcing standards is increasingly important to minimize waste and risk.

To firmly establish both control and visibility across today's complex and agile software supply chain, organizations should take the following steps toward Component Lifecycle Management (CLM)—or the practice of proactively managing the use of components throughout the supply chain.

Step 1: Inventory – Gather information about your current component usage:

- Track component downloads and usage to understand consumption.
- Inventory internal component repositories to determine what is being distributed to development teams.
- Understand the software supply chain to determine which components and dependencies are being introduced to the organization.

Step 2: Analyze – Understand vulnerabilities in applications and repositories:

- Analyze key applications to uncover known security vulnerabilities.
- Analyze internal component repositories to discover vulnerable components.

Step 3: Control – Establish controls throughout the development lifecycle:

- Establish policies regarding security, the use of viral licenses and the out-of-date or out-of-version components.
- Eliminate or blacklist known vulnerable components in internal repositories.
- Establish mechanisms to prevent known flawed components from entering the organization.
- Implement controls in build and continuous integration systems to prevent inclusion of flawed components in software builds.

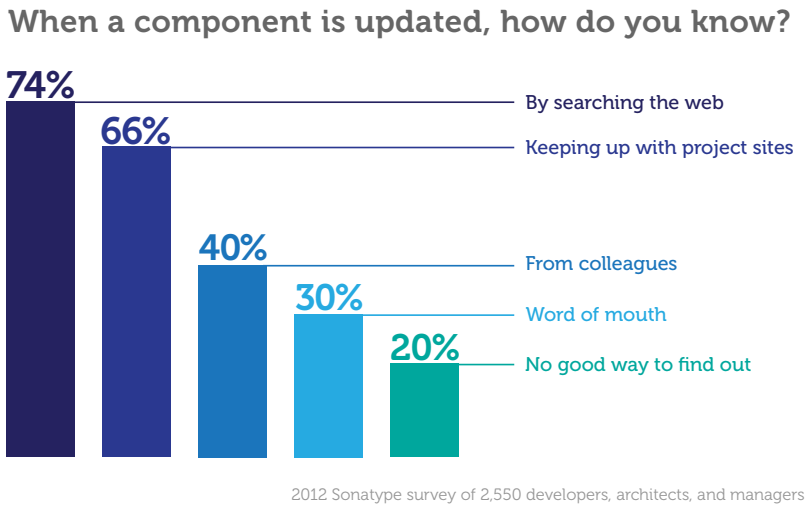


Figure 4: From reference [4]

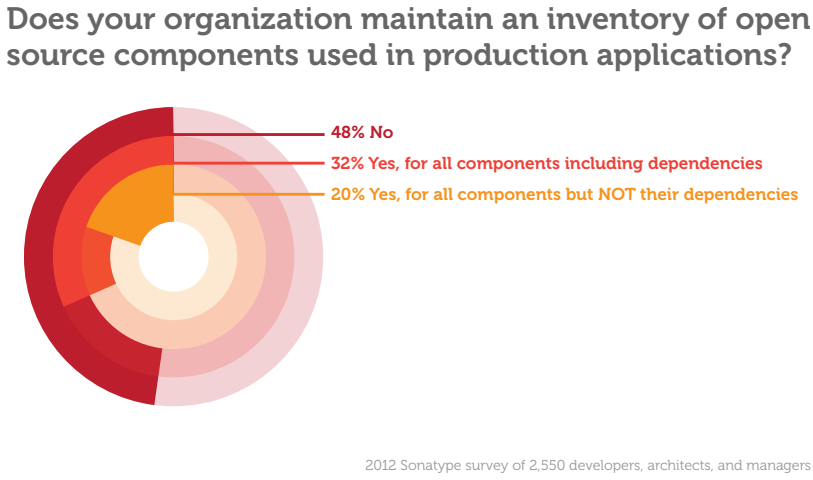


Figure 5: From reference [4]

Step 4: Monitor – Maintain awareness of component updates:

- Maintain an inventory of all components and dependencies used in production applications.
- Continuously monitor application bill-of-materials for updates and newly discovered vulnerabilities.

Properly managing the use of open source components throughout the software development lifecycle will enable organizations to ensure the integrity of the software supply chain and focus on the cost savings and wealth of innovation open source software can bring.

ABOUT THE AUTHOR

Wayne Jackson is CEO of Sonatype, a provider of component lifecycle management tools. Prior to joining Sonatype, he was the CEO of open source network security pioneer Sourcefire, Inc., which he guided from fledgling startup through IPO in March of 2007 to a peak valuation of over $750 million. Before joining Sourcefire, Mr. Jackson co-founded Riverbed Technologies, a wireless infrastructure company, and served as its CEO until the sale of the company for approximately $1 billion. While at Riverbed, he built strategic relationships with industry leaders Palm, Oracle, IBM, Symbol and Microsoft, growing the company from startup to category winner in less than two years. Mr. Jackson holds a B.B.S. in Finance from James Madison University and has completed the Executive Education program for Corporate Governance at Harvard University.

Phone: (301) 684-8080
E-mail: wjackson@sonatype.com

REFERENCES

1. Aspect Security, "The Unfortunate Reality of Insecure Libraries," March 2012
2. Sonatype Inc., "Executive Brief: Addressing Security Concerns in Open Source Components," March 2012
3. The Central Repository (2012)
4. Sonatype Inc., "2012 Open Source Software Development Survey," April 2012

Advancing SCRM with Standardized Inspection Technology

Roger Stewart, Stewart-Priven Group

Abstract. Technology exists today that can make a huge improvement minimizing risk along the supply chains and improve delivery of secure, high-quality products, on time and within budget. And no changes to standards, legislation, or acquisition models are needed. Accepting this approach to Supply Chain Risk Management (SCRM) by industry and government means adopting policies to:

1. Impose contract stipulations on the prime contractor, such as common tools and process use, that must also apply to all vendors along their supply chain, and their vendor's supply chains,
2. Require a common, standardized platform of: tools, process and training along the supply chains for consistently performing ongoing product risk assessments through defect identification and removal on pre-code product definition artifacts (e.g., contracts, requirements, architecture, design, interfaces), where past studies report more than 70% of product defects historically originate [1].
3. Require results of each product risk assessment be made available to both the prime contractor and the government program manager in the form of an automated, tool-generated report with common format and content.
4. Include contract leverage for the government program office based upon their ongoing evaluations of the resulting product risk visibility.
5. Use a tool-enabled, standard-compliant inspection process for defect identification and removal in Product Definition artifacts along the supply chain and for achieving the ongoing product risk assessments and reports.

Introduction

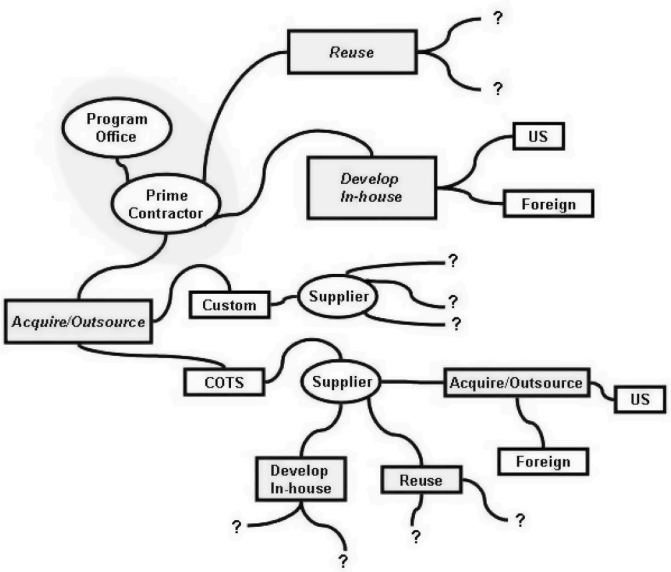


Figure 1. Potential Software Supply Chain Paths

The supply chains in today's software acquisition world consist of a wide variety of suppliers spread across the world (Figure 1). Each of these suppliers may have their own standards for development and quality assurance. Therefore, the responsibility for software assurance must be shared not only by software suppliers in the supply chain but also by the acquirer in the supply chain who purchase the software [2].

A key to advancing SCRM is through standardized inspections using tool-enabling technology for correction of past inspection perceptions and shortfalls.

The first 4 of the 5 proposed policies (see Abstract) are contract related actions that are dependent upon the viability of standardized inspections to provide sustained results in removing pre-code defects and reporting on the associated product risk assessments inspections can provide. This article will explore the viability of incorporating standardized inspections along the supply chain (Figure 2) for visibility into pre-code product definition activities.

Note: Federal Acquisition Regulation Part 46.202 on quality assurance currently addresses inspection on government contracts [3].

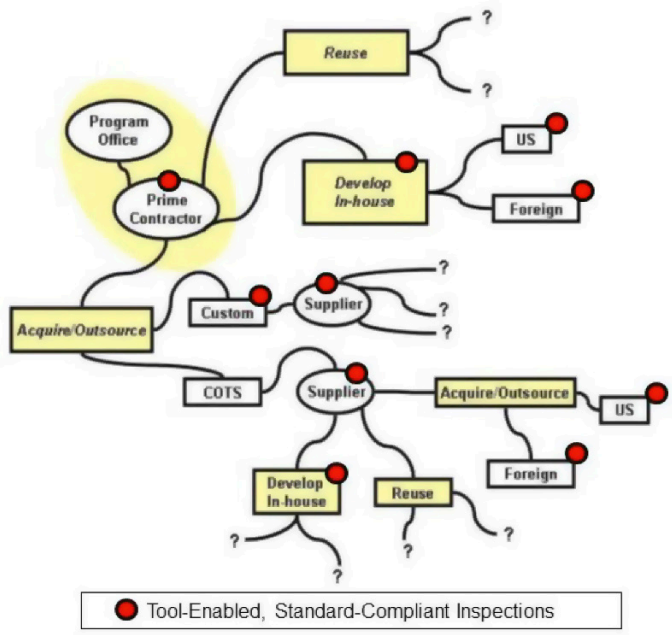


Figure 2. Standardized Inspection of Product Definition Artifacts along the Supply Chain

Inspections are a preventative systematic analysis of a work-product or portion thereof, by a team of three to five peer stakeholders to remove defects at or closest to their points of introduction. Inspections provide the best value when applied to up-front, pre-code product definition artifacts where more than 70% of product defects are historically introduced [1], and on change instruments (e.g., test fixes, change requests) which are defect-prone due to their late application to a product.

Using a tool-enabled, standard-compliant[1] inspection process implementation[2] is a key to removing product definition defects and eliminating past dramatic variances in inspection results and benefits.

For example, recent training in tool-enhanced, standard compliant inspections on a DoD agency project resulted in the six participating teams collectively finding 236 non-trivial defects in the 290 lines of actual requirement text their project had previously generated; a defect density of 814 defects per 1,000 lines of requirements. Afterward on this project's first post-training inspection using the tools and techniques they had learned in class, the trained inspectors discovered 163 non-trivial defects in another 180 lines of requirement text; a defect density of 906.

Comparing fixing these defects at their points of introduction using a compliant inspection process (as depicted above and in Figure 3), versus randomly discovering and fixing problems throughout development (e.g., requirement reviews, design, code, test) and operations without compliant inspections, revealed potential estimated net project savings of more than 20,000 hours from the training results and an additional 10,000 or more hours from their first post-class inspection! When further considering the resulting quality, security and schedule issues that can spawn from defects not found early, the benefits of uniformly applying compliant inspections throughout the supply chains, shown by the circle indicators in Figure 2, to pre-code activities can be better appreciated.

TOOL-ENABLED, STANDARD COMPLIANT INSPECTION RESULTS		
DOD AGENCY PROJECT - EXAMPLE	Training Class Inspections	Post-Class 1st Inspection
1. Requirement Lines Inspected	290	180
2. Non-Trivial* Defects Found	236	163
3. Defect* Density (per 1,000 lines)	**814**	**906**
4. Hours to Find-Fix Defects* by Inspection	343	225
5. Hours Est. to Find-Fix Defects* without Inspection	21,289	10,898
6. Projected Hours Saved by Early Defect* Removal	20,946	10,673
7. ROI (Return on Investment)	**61.1**	**47.4**
* non-trivial defect: incorrect, missing, unclear or ambiguous statement		

Figure 3. Example of Compliant Inspections applied to Product Definition Material

Standardizing Inspections

Tragically, effective inspections are no longer used by most organizations, despite the perform peer reviews specific goal and practices of CMMI® Maturity Level 3. Reasons why this previous best practice [4] is no longer in favor include the following:

1: Inspection Pitfalls – the 10 most important reasons were captured in Stewart-Priven Group's article [5] published in January 2008 explaining why organizations experience poor inspection value and inability to sustain early defect removal results due to unknowingly encountering any of the 10 inspection Pitfalls described:

1. Immature development infrastructure
2. Management responsibilities not understood
3. No inspection planning tools
4. Insufficient time allotted
5. No inspector execution tools for consistency–rigor–completeness, and no tools/reports for management monitoring
6. Limited result tracking & analysis
7. No post-training follow up
8. Lack of project-wide facilitation
9. Slow implementation
10. No inspection process capture

Adding to these pitfalls are a lack of education of software engineers, and the generic non-specific meaning that the terms peer-review and inspection have evolved to.

2: Code Inspection vs. Product Definition Inspection - Organizations then and now tend to believe inspection value applies mainly to code, which is not true. Code analyzer companies introduced high-tech static analysis capabilities in the early 2000s with marketing that downplayed inspection value and Fagan inspections in particular, attempting to convince industry and government to employ their automated defect removal technologies. Code analyzers, for the most part, are not succeeding in achieving high-quality software systems. General William Shelton's opening remarks at the April 2009 DoD Systems & Software Technology Conference focused on the deteriorating state of software-driven systems. More recently, this was a focus of J.M. Gilligan's article in the February 2012 issue of Software Technology titled "A Roadmap for Reforming the DoD's Acquisition of Information Technology[6]." How could code analyzers alone remedy a situation where the majority of defects are introduced pre-code, during product definition activity?

3: Inspection Cost vs. Project Savings - Organizations tend to believe inspections are a cost rather than a savings/investment despite huge quantities of data to the contrary. The Stewart-Priven Group introduced savings estimation capability for software-driven projects to easily demonstrate their expected net project savings from inspection across multiple disciplines before actually committing to inspection. Inspection planning tools can guide projects in using their own past development history (or estimates) to perform what-if analysis to pre-determine their projected net savings from using inspection. More visibility is needed into the merits of inspection planning and saving estimation tools.

4: Manual Inspection vs. Tool-Enabled Inspection - Computerized inspection tools bring added value to inspections by enabling adherence to the inspection standard in section six of IEEE Std 1028™-2008, ensuring consistency, completeness and rigor. Tool-enabled inspections can also provide interim and final one-page automated management reports of inspection process deviations, find/fix defect progress, ROI for individual inspections, accumulated labor and dollar savings, and rolled-up project inspection results. Unfortunately

The overriding value tools provide is in achieving compliance to the standard for adherence to the performance criteria boundaries required for effective inspections, and their ongoing product risk assessments.

misleading material clouds the perception of inspections or can imply the inspection process can be tailored by organizations; where resulting consequences would actually weaken or undermine the performance criteria upon which inspections depend for success. The establishment of performance criteria that defined and enabled effective inspections was the output of the five-year experiment in the 1970s, envisioned by Lew Priven who hired Michael Fagan to lead the effort to improve IBM software product quality while reducing cost and schedule. The experiment's result became known as "inspection" or "software inspection," used initially by IBM internally, and now the foundation of the 2008 Inspection Standard.

5: Streamlined Inspections vs. Compliant Inspections – Using inspections for effective early defect removal and ongoing product risk assessment must employ inspection specific tools to ensure any non-compliance with the inspection standard is identified during an inspection. Without tool-enabled inspections, deviating from the inspection standard/process is inevitable and organizations will fall into the same pitfalls that contribute to the tarnished reputation of traditional manual inspections. Organizations that streamline, tailor, or re-do inspection material to fit their needs, or for their own use under the guise of public domain, do not understand the limits of inspection performance criteria, or the importance of rigor, repeatability and consistency that standard-compliant inspection tools provide for ongoing early removal of defects in product definition artifacts that code analyzers are typically unsuccessful in removing. Inspections are required for early defect removal in product definition artifacts, as recent history seems to demonstrate [7]; plus, there is no current alternative to inspections for ongoing product risk assessment! CMMI for process adherence is necessary but history and DoD studies have shown it not to be sufficient for attaining consistent high-quality, on-time, and within budget deliveries [2]. Without using standard-compliant inspection execution tools in real-time throughout the seven-step inspection process, then effective early defect removal from product definition artifacts and change instruments cannot be sustained leading to project efforts becoming prohibitively expensive or failing.

6: Human Shortfalls: These include resistance to change, preserving the status quo, protecting personal income and influence, and egos that make peers resist detecting errors in their work. All these contribute to late and costly, problem-ridden capabilities.

The Way Forward

Inspection process adherence using standard-compliant, computerized tools must be the prime focus of inspection training and repeatedly reinforced to consistently reap the benefits of pre-code early defect removal, as the results in Figure 3 demonstrate. There are other advantages of an integrated inspection tool set to supply chain risk management such as consistent and complete data collection, result tracking, and accumulated savings. However the overriding value tools provide is in achieving compliance to the standard for adherence to the performance criteria boundaries required for effective inspections, and their ongoing product risk assessments. Product risk visibility derived from each compliant inspection across the supply chain, and uniformly formatted by a contract specified inspection tool can provide the prime contractor and government program office a powerful vehicle to manage product risk.

Awareness of the vendor offerings for standard-compliant, tool-enhanced inspection capabilities that eliminate past inspection shortfalls and misleading perceptions; and awareness that inspection must be used during product definition activity where most defects are introduced, and on change instruments due to their defect-prone nature. As for code, the code analyzers continue to be necessary but are not sufficient. This message needs to stress that tool-enabled, standard-compliant inspections are the most effective alternative before coding. J. M. Gilligan's February 2012 article [6], with its several references to recent government reports, highlights the inadequate state of government software-driven systems, and that a path forward is available with current technology.

Summary/Conclusion

Consistently attaining on-time, high-quality, cost-effective, and secure software requires agile techniques, process adherence discipline, sophisticated code-analyzers, and rethinking supply-chain risk management. These are all being done today, but they are not enough to end quality, schedule, and cost struggles! Ongoing product risk assessments also need to be the norm, coupled with effective early defect removal from pre-code product definition artifacts. Tool-enhanced, standard-compliant inspection provides both.

Deploying standardized inspections for product definition artifacts along the supply chain shown in Figure 2 would go a long way to mitigating current supply chain risk and improve product quality, security, schedule and cost.

Disclaimer:

CMMI® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

NOTES

1. Standard: document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context
2. Process (formal): set of interrelated or interacting activities which transforms inputs into outputs; and in the context of this article, implementing a Standard

ABOUT THE AUTHOR

Roger Stewart is co-founder and Managing Director of the Stewart-Priven Group. He is an experienced Lead Systems Engineer and Program Manager in both government and commercial system development, including Systems Engineering, Software Development, System Integration, System Testing, and Process Improvement.

Previously, Stewart taught the Fagan Defect-Free Process for Michael Fagan Associates (8 years) after spending 31 years with IBM's Federal Systems Division, (now part of Lockheed-Martin) managing and developing systems for Air Traffic Control, Satellite Command & Control, On-Board Space Shuttle, Light Airborne Multipurpose System (LAMPS Helicopter); and in Commercial Banking, Telecommunication and Networking systems.

Roger has a BS in Mathematics from Cortland University; published other Inspection articles and presented on the topic at the annual DoD Systems & Software Technology Conference, including a plenary session breakfast in 2009.

P.O. Box 2174
Mt. Juliet TN 37121
Phone: (615) 754-6685
E-mail: rstewart025@gmail.com

REFERENCES

1a. McGraw, Gary. <http://www.cigital.com/whitepapers/dl/Making_Essential_Software_Work.pdf> "Making Essential Software Work" Citgal, Inc. Mar. 2003;
1b. Bender, Richard. <http://www.glensfallsregion.com/business/bender-rbt-inc-5233/website/> [Position Papers] "The Business Case for Software Quality" page 23, 2004
2. The Software Assurance (SwA) Acquisition Working Group. "Software Assurance in Acquisition: Mitigating Risk to the Enterprise." October 22, 2008
3. Federal Acquisition Regulation, "FAR –Sub-Part 46.202, Quality Assurance" (FAC 2005-53), August 2010
4. McConnell, Steve. '10 best influences on SW Engineering over the past 50 years', IEEE Software, January/February 2000
5. Priven, Lew, and Stewart, Roger. "How to Avoid Software Inspection Failure and Achieve Ongoing Benefits." Crosstalk The Journal of Defense Software Engineering, January 2008
6. Gilligan, John M. "A Roadmap for Reforming the DoD's Acquisition of Information Technology"' Journal of Software Technology, February 2012
7a. House Armed Services Committee Panel on Defense Acquisition Reform, Interim Findings and Recommendations, DAR Interim Report, March 4, 2010;
7b. National Research Council, Committee on improving Processes and Policies for the Acquisition and Test of Information Technologies in the Department of Defense, Achieving Effective Acquisition of Information Technology in the Department of Defense, 2010;
7c. Business Executives for National Security (BENS) Task Force on Defense Acquisition Law and Oversight, Getting to Best: Reforming the Defense Acquisition Enterprise, July 2009;
7d. Tech America, Recommendations on Information Technology Acquisition Reform, Presented to the Defense Acquisition Reform Panel of the House Armed Services Committee, February 23, 2010.

Building a Body of Knowledge for ICT Supply Chain Risk Management

Dan Shoemaker, Ph.D., University of Detroit Mercy
Nancy R. Mead, Ph.D., SEI

Abstract. This paper proposes a set of Supply Chain Risk Management (SCRM) activities and practices for Information and Communication Technologies (ICT). This set can be used as a starting point to create a body of knowledge in SCRM to ensure the integrity of ICT products.

Introduction

ICT is a vital part of our culture. In fact, many would argue that computers and their associated communications technologies have created that culture. Because we depend so much on our ICT products, it is critically important to be able to trust their integrity. Yet, commonly used ICT development and sustainment practices still permit dangerous defects that allow attackers to compromise millions of computers every year [1]. The increasing trend toward building systems out of purchased parts just enhances the importance of getting the acquisition of ICT components right [2].

Early in this decade, NIST estimated that exploitation of ICT defects costs the U.S. economy an average of $60 billion annually, and there is no reason to think that those numbers have improved since then [3]. But the real concern is not cybercrime; it is that the exploitation of a point of failure in an infrastructure component like power or communication could have severe consequences. Therefore, it is not surprising that the U.S. government is addressing the problem of product integrity through a comprehensive program to get better SCRM practices into the workforce. This program includes education, training, and awareness initiatives, which are the traditional means of leveraging the required change in workforce behavior. However, when it comes to SCRM, although much progress has been made [4, 5] there is still no single reference to define what should be taught [6, 7].

An authoritative Body of Knowledge (BOK) of best practices for SCRM is an attractive idea. Such a BOK would portray the SCRM process as a complete set of topics. The BOK would integrate the knowledge needed for effective management of supply chain risk into a framework that contains all of the advice necessary to ensure ICT product integrity. The aim would be to characterize and relate all the detailed knowledge elements needed to develop precise workforce learning requirements, as well as the methods to deliver that learning. In addition, a commonly accepted BOK could be used as leverage to develop new education and training curricula as the field evolves [6, 7].

Several conventional disciplines could be part of a discipline of ICT SCRM, such as hardware and software engineering, systems engineering, information systems security engineering, safety, security, reliability, testing, information assurance, and project management [6]. In addition, it would be possible to consider academic areas such as intelligence analysis and law as potential parts of the discipline. Because these are highly disparate fields, it is important to create a detailed model of the relationship between all of the logical components in order to judge whether the right content is being provided in each education and training setting.

ICT SCRM in Common Standards

ICT products are developed through a global supply chain. Supply chains are no different from any other organizational function in that they are intended to accomplish a specific purpose. The purpose of all supply chains is to provide a product or service through coordinated work that involves several organizations. The concerns about supply chains fall into five categories. Each category has slightly different implications for product integrity: "Installation of malicious logic on hardware or software, installation of counterfeit hardware or software, failure or disruption in the production or distribution of a critical product or service, reliance upon a malicious or unqualified service provider for the performance of a technical service and installation of unintentional vulnerabilities on software or hardware [2]."

Proper SCRM mitigates these concerns by providing a consistent, disciplined environment for developing the product, assessing what could go wrong in the process (i.e., assessing risks), determining which risks to address (i.e., setting mitigation priorities), implementing actions to address high-priority risks and bringing those risks within tolerance [8]. Typically, supply chains are hierarchical, with the primary supplier forming the root of a number of levels of parent-child relationships. From an assurance standpoint, what this implies is that every individual product of each individual node in that hierarchy has to be correct as well as correctly integrated with all other components up and down the production ladder. Because the product development process is distributed across a supply chain, maintaining the integrity of the products that are moving within that process is the critical concern.

The weak link analogy is obvious here, so, whether the product is a common household item or sophisticated military hardware, the activities within that product's supply chain have to be precisely coordinated and carefully controlled. Authoritative control processes already exist, which specifically address the existing coordination and control concerns. These processes are embodied in the activities and tasks of two international "umbrella" standards. The recommendations of these standards have been validated worldwide. So besides providing authoritative real-world advice about how to manage supply chain risk, the detailed activities and tasks that are specified in those standards also provide a coherent and detailed logic for a BOK of SCRM best practices.

Building a Framework for the BOK of SCRM Education From Two International Standards

At present, there is no complete classification structure for the BOK for ICT SCRM. Thus our aim was to derive a conceptual model for the discipline based on existing standards. A standard conceptual model is essential in order to ensure proper associations between the many disparate knowledge, skills, and abilities required to produce, maintain, and acquire trustworthy ICT products. The DHS uses the term Enterprise Security Framework (ESF) to describe the specific set of actions needed to ensure the reliability of purchased products [9]. The aim of an ESF is to factor everybody's actions for achieving secure products into a "who, what, when" structure of defined activities and interrelationships. To create this structure, DHS suggests that the ESF must include responsibilities beyond the typical system and software security activities seen in most organizations. These responsibilities can be implemented by blending top-level risk management activities contained in the ISO 16085 Lifecycle Risk Management standard with the activity and task recommendations of the Agreement processes of the ISO 12207-2008 standard.

The activities embodied in the 12207 Agreement process convey the steps that an organization should take to, "manage the procurement of a system, software, or service product." The agreement processes are particularly relevant to those interested in defining the discipline of SCRM in that they provide a structured and rigorous set of activities and tasks to carry out the effort. The 12207-2008 activities specified for acquisition convey the practices that have to be performed when an organization procures a software system or service, while the supply process (6.1.2) delineates the obligations of the supplier. Using the 12207-2008 standard, it is possible to form a detailed definition of the standard customer supplier activities involved in ICT procurement. However, that definition does not take risk management into consideration.

The purpose of ISO 16085 System and Software Supply Chain Risk Management is to identify potential managerial and technical actions to reduce or eliminate the probability and impact of risk [10]. The standard may be used for managing risk at the organizational, enterprise, or project level in any domain or lifecycle stage [10]. The aim is to support the perspectives of managers, suppliers, acquirers, developers, participants, and other stakeholders and provide them with a single set of process requirements suitable for the management of a broad variety of risks in the supply chain [10]. The standard prescribes a continuous process for risk management and is useful for managing the risks associated with organizations dealing with system or software [10]. Moreover, 16085 is specifically designed to be used in conjunction with ISO 12207-2008.

Thus, the recommendations of the standard can be directly aligned with the risk management activities specified by the 12207 project process area (6.4). When used with ISO 12207-2008, the 16085 standard assumes that necessary managerial and technical processes to perform the treatment of risk are called out by the ISO 12207-2008 model. The addition of the risk management component to the standard procurement model represented in the 12207 agreement processes provides a complete set of practices for ICT SCRM.

Creating an Instructional Model for SCRM

SCRM issues are different for the acquirer, supplier, and integrator. In addition, there are at least four different types of environments that require a specific approach to SCRM: high assurance (trusted), government-off-the-shelf, commercial-off-the-shelf, and services. Given that diversity, our aim was to derive a standard set of activities and practices from the two standards discussed in the prior section. Our goal was to derive a point of reference for content and teaching development.

The relevant lifecycle process activities that were incorporated in our approach are from the 12207-2008: Agreement, Reuse, Technical, and Supporting and Project Management process areas. These recommendations were integrated with the ICT Risk Management process recommendations that are specified by ISO 16085. The content model derived from integrating these two references ultimately leads to a set of lifecycle activities and practices, which can form a starting point for development of a complete BOK for management of supply chain risk.

Once such a BOK has been perfected, explicit learning behaviors can be derived for each content item. Then, appropriate standard instructional content can be designed and created to reinforce each behavior along with a set of proficiency requirements specified for each action. Instructional content can be customized from the BOK to address each situation in which it will be applied. The approach to content delivery can be referenced to learning and proficiency specifications. From an evaluation standpoint, the ability to perform each task can be characterized as a nominal set of observable actions. The knowledge needed to perform each task and/or the skill required to perform each task can be characterized as an ordinal judgment of proficiency.

The list below summarizes the general subject areas that evolved from our process of creating an instructional model for SCRM, using the recommendations of the two standards.

Subject Area One: Project Initiation and Planning

- Strategic management and policy
- Project management
- Business and assurance case development
- Supply chain component definition and labeling
- Threat/risk and mitigation identification and planning

Subject Area Two: Product Requirements Communication

- Requirements elicitation and specification
- Requests for proposals (RFP) documentation
- Statement of work (SOW) documentation
- Project assurance criteria development (including SCRM)
- Project measurement and metrics development (including SCRM)
- Formalization and documentation of product assurance case requirements
- Preparation and documentation of acceptance criteria

Subject Area Three: Source Selection and Contracting
- Source selection process
- Source evaluation process
- Contract negotiation
- Contract writing
- Lifecycle contract management planning
- Lifecycle project management planning

Subject Area Four: Supplier Contract Execution
- Document framework for ICT project management.
- Document plan to manage the quality and security of the project.
- Implement and execute the project management plan(s).
- Monitor and control progress throughout the contracted lifecycle.
- Manage and control the subcontractors.
- Interface with the independent verification, validation, or test agent.
- Coordinate contract review activities and interfaces.
- Conduct joint reviews in accordance with ISO standard specifications.
- Perform verification and validation to satisfy that requirements are met.

Subject Area Five: Customer Agreement Monitoring
- Monitor the supplier's activities in accordance with the contracted software assurance process
- Develop plan to supplement monitoring with verification and validation as needed
- Develop plan to ensure necessary information is provided in a timely manner

Subject Area Six: Customer Acceptance
- Plan acceptance process based on contracted acceptance strategy and criteria
- Plan test cases, test data, test procedures, and test environment
- Conduct acceptance review and acceptance testing of the deliverable
- Accept product from supplier when all acceptance conditions are satisfied
- Plan to migrate product from supplier to customer

Subject Area Seven: Project Closure
- Make payment or provide other agreed consideration to the supplier
- Install the product in accordance with established requirements
- Ensure agreement terminates when payment is made
- Transfer legal responsibility for the product or service to the customer
- Provide assistance to the customer in support of the delivered product

The subject areas and detailed activities and practices of SCRM provide support for the development of a formal discipline of SCRM. Once the discipline is codified, this material can be integrated into traditional ICT education and training programs. ISO 12207-2008 provides a commonly accepted definition of best practices for ICT acquisition and supply, while activities and tasks specified in ISO 16085 provide an excellent collection of assurance practices for ICT work. This makes it possible to construct a detailed picture of SCRM practices and activities that can be used in building an SCRM body of knowledge.

Example Activities and Practices for SCRM

Procurement Program Initiation and Planning
- Develop the concept to acquire (business case).
- Define project scope and boundaries.
- Develop an acquisition strategy and/or plan.
- Define constraints.
- Make decision to contract.
- Identify and mitigate outsourcing definitions.
- Install risk management process.
- Perform product assurance risk assessment.
- Develop product assurance risk mitigation strategies.
- Ensure product assurance risk monitoring.

Product Requirements Communication
- Issue written requests to prospective suppliers.
- Standardize elements of the RFP.
- Document SCRM needs and requirements in the RFP.
- Specify SCRM terms and conditions.
- Specify information security features.
- Specify acceptance criteria for COTS integrations.
- Implement common criteria (if required).
- Create a specification.
- Specify SCRM measures and metrics.
- Create assurance language for a statement of work (SOW).
- Assure requirements for C&A in SOW.
- Ensure SOW specifies SCRM education and training.
- Develop SOW to acquire COTS.
- Provide SCRM language in instructions to suppliers.
- Ensure response reflects the specified capabilities.
- Ensure supplier has submitted adequate information.
- Specify initial product architecture.
- Specify product assurance case management procedure.
- Specify product assurance lifecycle.
- Specify product requirements and traceability criteria.

Source Selection and Contracting
- Specify evaluation criteria.
- Ensure standard product assurance evaluation criteria.
- Specify assurance criteria in the Source Selection Plan.
- Perform contract negotiations.
- Perform project/contract management.

- Plan to oversee product assurance reviews and audits.
- Ensure competent product assurance professional(s).
- Oversee the supplier's delivery of product assurance.
- Define the rate at which the supplier will provide assurance statements.
- Define how performance will be evaluated if an SLA is used.
- Define the role that product assurance plays in product C&A.
- Define how the product architecture will be managed.
- Define what will be reviewed from an assurance perspective.
- Define how often the risk management plan will be updated.
- Define how often the product assurance risks will be evaluated.
- Devise an issues resolution plan and process.
- Define circumstances for intelligence updates.
- Define how corrective actions will be monitored.
- Define how product assurance savings will be measured.
- Define how experience level will be monitored.
- Define how to identify key product personnel.
- Define how key personnel will be monitored.
- Define how assurance training program will be monitored.

Supplier Contract Execution

- Create a management framework for the ICT project.
- Select a lifecycle model.
- Select processes, activities, and tasks and map them to lifecycle model.
- Develop a plan to manage the quality and security of the project.
- Develop document project management plan(s).
- Implement and execute the project management plan(s).
- Monitor and control progress throughout the contracted lifecycle.
- Develop the software product using internal resources.
- OR develop the software product by subcontracting.
- Buy off-the-shelf software products from internal or external sources.
- Monitor the progress of the project.
- Manage and control the subcontractors.
- Ensure all contractual requirements are passed to subcontractors.
- Interface with the independent verification, validation, or test agent.
- Interface with other parties as specified in the contract and project plans.
- Coordinate contract review activities and interfaces.
- Conduct joint reviews in accordance with ISO standard specifications.
- Perform verification and validation to satisfy that requirements are met.
- Make reports available as specified in the contract.

Customer Agreement Monitoring

- Monitor supplier's activities using the Software Review Process.
- Supplement monitoring with verification and validation as needed.
- Ensure necessary information is provided in a timely manner.

Customer Acceptance

- Prepare for acceptance based on the acceptance strategy.
- Prepare test cases, test data, test procedures, and test environment.
- Define the extent of supplier involvement in acceptance.
- Conduct acceptance review and acceptance testing of the deliverable.
- Accept product from supplier when all acceptance conditions are satisfied.
- Arrange to make customer responsible for configuration management.

Project Closure

- Make payment or provide other agreed consideration to the supplier.
- Install the product in accordance with established requirements.
- Ensure agreement terminates when payment is made.
- Transfer responsibility for the product or service to the customer.
- Provide assistance to the customer in support of the delivered product.

Conclusion

We have proposed a set of SCRM activities and practices in this paper. These activities and tasks comprise an initial picture of the knowledge needed to correctly and effectively conduct a practical SCRM process. We derived this set of activities and practices from established models of practice for acquisition and supply of software and systems, along with additional risk control elements. We believe that it will be possible to create a true body of knowledge in SCRM using this set as a starting point. SCRM is clearly a huge field composed of a number of not clearly related subjects. The first step in creating a body of knowledge for the field is to provide a top-level classification structure of its practices and activities, which we propose in this paper.

ABOUT THE AUTHORS

Daniel P Shoemaker, Ph.D., is Principal Investigator and Senior Research Scientist at UDM's Center for Cyber Security and Intelligence Studies. He is also a full time Professor and former Department Chair at University of Detroit Mercy. As the Co-Chair for the, National Workforce Training and Education Initiative he is one of the authors of the DHS Software Assurance Common Body of Knowledge (CBK). He also helped author the DHS IA Essential Body of Knowledge and he serves as a SME for the NIST-NICE workforce framework. Dan's doctorate is from the University of Michigan and within the State of Michigan he leads the International Cyber-Security Education Coalition. This Coalition covers a five state region with research partners as far away as the United Kingdom. Dan also spends his free time authoring some of the leading books in Cyber Security. His book "Cyber Security: The Essential Body of Knowledge," is Cengage publishing's flagship book in the field. His first book, "Information Assurance for the Enterprise," is McGraw-Hill's primary textbook in IA and is in use all over the globe. His next book, "Engineering a More Secure Software Organization," which is also published by Cengage, will be out soon.

E-mail: dan.shoemaker@att.net

Nancy R. Mead, Ph.D. is Senior Member of the Technical Staff, CERT Secure Software and Systems, in the CERT Program at the SEI. She is also a faculty member in the Master of Software Engineering and Master of Information Systems Management programs at Carnegie Mellon University. She is currently involved in the study of security requirements engineering and the development of software assurance curricula. She also served as director of education for the SEI from 1991 to 1994. Her research interests are in the areas of information security, software requirements engineering, and software architectures.

Prior to joining the SEI, Mead was a senior technical staff member at IBM Federal Systems, where she spent most of her career in the development and management of large real-time systems. She also worked in IBM's software engineering technology area and managed IBM Federal Systems' software engineering education department. She has developed and taught numerous courses on software engineering topics, both at universities and in professional education courses.

Mead has more than 150 publications and invited presentations, and has a biographical citation in Who's Who in America. She is a Fellow of the Institute of Electrical and Electronic Engineers, Inc. (IEEE) and the IEEE Computer Society, and a Distinguished Member of the ACM. Mead serves on the Editorial Boards for the International Journal on Secure Software Engineering and the Requirements Engineering Journal, and is a member of numerous advisory boards and committees.

Mead received her Ph.D., in mathematics from the Polytechnic Institute of New York, and received a BA and an MS in mathematics from New York University.

E-mail: nrm@sei.cmu.edu

REFERENCES

1. Clark R.A. and H.A. Schmidt, "A national strategy to secure cyberspace," The President's Critical Infrastructure Protection Board, Washington, DC, 2003.
2. GAO Report to Congressional Requesters. IT Supply Chain: National Security-Related Agencies Need to Better Address Risks. United States Government Accountability Office, March 23, 2012.
3. Newman, Michael. Software Errors Cost U.S. Economy $59.5 Billion Annually. Gaithersburg: National Institute of Standards and Technology (NIST), 2002.
4. Ellison, Robert, Christopher Alberts, Rita Creel, Audrey Dorofee, and Carol Woody. Software Supply Chain Risk Management: From Products to Systems of Systems. CMU/SEI-2010-TN-026. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2010. <http://www.sei.cmu.edu/library/abstracts/reports/10tn026.cfm>
5. Ellison, Robert, John Goodenough, Charles Weinstock, and Carol Woody. Evaluating and Mitigating Software Supply Chain Security Risks. CMU/SEI-2010-TN-016. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2010. <http://www.sei.cmu.edu/library/abstracts/reports/10tn016.cfm>
6. Redwine, Samuel T., ed. Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software, Version 1.1. Washington: U.S. Department of Homeland Security, 2006.
7. Mead, Nancy, Julia Allen, Mark Ardis, Thomas Hilburn, Andrew Kornecki, Richard Linger, and James McDonald. Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum. CMU/SEI-2010-TR-005. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2010. <http://www.sei.cmu.edu/library/abstracts/reports/10tr005.cfm>.
8. Alberts, Christopher, and Audrey Dorofee. A Framework for Categorizing Key Drivers of Risk. Rep. no. CMU/SEI-2009-TR-007. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2009.
9. Allen, Julia H. "Security Is Not Just a Technical Issue." Build Security In Website. Department of Homeland Security. 2009. <https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/management/563-BSI.html>.
10. International Standards Organization. Systems and Software Lifecycle Process Risk Management – ISO/IEC 16085. ISO, 2006.

Ensuring Your Development Processes Meet Today's Cyber Challenges

Mary Beth Chrissis, Carnegie Mellon University
Mike Konrad, Carnegie Mellon University
Michele Moss, Booz Allen Hamilton

Abstract. While security in the physical world can be addressed using controls such as guns, gates, and guards, the virtual world requires other mechanisms to ensure the confidentiality, availability, and integrity of products and services. Much of what today's products, services, infrastructures, and institutions do is automated by software, thereby increasing our dependence on how safety and security are addressed in the virtual world. As software continues to evolve and we find new ways to leverage the virtual world in our day–to-day activities, the volume of and our reliance on software grows exponentially. Therefore, it is increasingly important to have confidence that products operate as intended and only as intended to ensure the resilience and reliability of the functions they support. Much of software is acquired forcing consideration of these critical qualities into the supply chain. Achieving such confidence ultimately relies on good system and software engineering knowledge, processes, and technology. Fortunately, many resources are available. This article provides a brief survey of some of these resources, such as process capability frameworks, secure lifecycle practices, and implementation approaches.

Introduction

Global markets, funding, and shareholder commitments often drive companies to get their products and services out to the market quickly. However, ignorance of vulnerabilities, heuristics, and biases [1] result in exploitable weaknesses that may affect the safety and security as well as the reputation of products. It is the market demand for newer, faster, and cooler that drives the pace of technology. However, software developers make it happen. We find a gap in security and safety in the products today because of the lack of both market requirements and developer skills. Much of software is acquired forcing consideration of these critical qualities into the supply chain.

Today consumers expect products to have safety and security built in. They take such quality attributes[1] for granted. However, over half of the 240 companies surveyed in a recent Forrester study reported at least one web application security incident since last year. The most frequently cited causes were misused default password accounts, SQL injection-related vulnerabilities, and security misconfigurations [2]. This is particularly challenging since many organizations acquire software products and must ensure their expectations are met through acquisition where requirements and design expectations must be clearly conveyed.

Developers initially tend to focus exclusively on product functionality, treating safety and security as something to test at the end of product development. In other words, they miss the opportunity to identify and mitigate vulnerabilities early in the development lifecycle. If problems are not detected and addressed early, they frequently are too expensive to fix when discovered later in the lifecycle.

The Threat Landscape

According to the October 2011 report from the Office of the National Counterintelligence Executive to Congress on Foreign Economic and Industrial Espionage, "Because the United States is a leader in the development of new technologies and a central player in global financial and trade networks, foreign attempts to collect U.S. technological and economic information will continue at a high level and will represent a growing and persistent threat to U.S. economic security. The nature of the cyber threat will evolve with continuing technological advances in the global information environment [3]."

The report further identified specific technology areas (i.e., information and communications technology, military technologies, clean technologies, advanced materials and manufacturing techniques, healthcare, pharmaceuticals, and agricultural technology) and types of business information (i.e., energy and other natural resources, business deals, and macroeconomic information) as targets of foreign attack.

For the past five years, Verizon has published its annual Data Breach Investigations Report. In these reports, Verizon analyzes the causes of breaches and provides recommendations for how they could have been prevented. These studies provide valuable insight into the level of complexity in today's attacks as well as organizational behaviors that either enable or hinder attacks.

	2011 Verizon Data Breach Investigations Report [4]	2012 Verizon Data Breach Investigations Report [5]
What commonalities exist?	▪ 83% of victims were targets of opportunity ▪ 92% of attacks were not highly difficult ▪ 86% of incidents were discovered by a third party ▪ 96% of breaches were avoidable through simple or intermediate controls	▪ 79% of victims were targets of opportunity (-4%) ▪ 96% of attacks were not highly difficult (+4%) ▪ 92% of incidents were discovered by a third party (+6%) ▪ 97% of breaches were avoidable through simple or intermediate controls (+1%)
How do breaches occur?	▪ 50% utilized some form of hacking ▪ 49% incorporated malware *(lower percentages included physical attacks, privilege misuse, and social tactics)*	▪ 81% utilized some form of hacking (+31% increase) ▪ 69% incorporated malware (+20% increase) *(lower percentages included physical attacks, privilege misuse, and social tactics)*

Table 1. Data from Verizon Data Breach Investigations Report

Good Development Practice

To assure safe and secure products and services, good development practice must be used from the beginning. Safety and security should be viewed as enablers, not constraints because they impact an organization's goals and reputation. One general principle that facilitates stakeholders to focus on safety and security throughout the software lifecycle is to use iterative, continuous, and evolutionary approaches to direct product acquisition, development, and delivery. Such approaches provide developmental agility, which is helpful to effectively address high uncertainty and to respond to unanticipated change.

Good development practice that is specific to safety and security includes:

• Provide early and careful consideration to quality attributes. Experience shows that safety and security cannot be added at the end of the development lifecycle [6]. Rather, products need

to be developed with safety and security in mind from inception through disposal.

- Provide early and careful attention to an architecture that effectively addresses tradeoffs among quality attributes and product functionality.
- "Step to the left." Most human error can be best detected shortly after it is committed (and it is more cost-effective to remove). Organizations need to make the most of this principle when detecting errors in individual, team, and organizational processes.
- Use processes that encourage careful consideration of how errors may be introduced, detected, removed, and prevented. For example, explicit task kickoff and inspection checklists can be incorporated at multiple points in the software development lifecycle (SDLC) to sustain attention on common error patterns affecting quality. Also, requirements elicitation, architecture, risk management, and decision analysis processes (and thus software development teams) should encourage explicit attention to safety and security (as well as other critical quality attributes).
- View safety and security as an enabler of the organization's core mission and objectives and not as a constraint on creativity and innovation. Make sure these attributes are commonplace in the development of all products and services.
- Use reviews and code analysis tools to reduce code-induced vulnerabilities hereby developing higher quality code.
- Be informed; what you do not know can kill you. A team's overconfidence is an attacker's best friend. Most exploitable errors are not the result of a lack of creativity or motivation but the lack of knowledge about vulnerabilities and human oversights. Better knowledge, processes, and technology can help overcome these weaknesses and the limits of our self-awareness [1]. Learn to think like an attacker.

Changing Technology

Opportunities and challenges arise with our ever-increasing reliance on technology. Digital thievery and espionage are concerns that organizations have to think about for their employees. The nature of cyber threats is changing at an alarming rate. You must commit to knowing as much about security as the attackers know to hope to stay ahead of them. Learn what resources are available and weave it into your learning and business processes. Your development practices must consider security issues as they relate to technology as well as to what others have learned and are sharing about safety and security.

According to a Forrester study, "ROI was greater for those who employed a coordinated, prescriptive approach [7]."

Unfortunately, many organizations are unable to communicate the return on investment effectively enough to gain management support in driving the adoption of more secure practices. Lack of management support and resistance to change is a barrier to the adoption of secure development methodologies.

In a more recent Forrester study, challenges contributing to the volume of insecure software included [2]:

- Developers being unable to keep pace with the volume of code they produce.
- Struggles to build the business case for additional funding.
- The lack of adequate tools.

Diversity of Resources

Do not limit yourself to the perspectives offered by only one process improvement model. As George Box is famous for saying, "All models are wrong but some are useful [8]." Models often become more useful when used in aggregate. Process improvement can and should incorporate knowledge of superior practice and vulnerabilities. Process measurement can help to determine the effects of particular development practices, making both cause and effect more salient, and elevating the state of software programming from ad-hoc practices toward evidence-based software development.

The DHS, NIST and the DoD are tackling this problem with their Software Assurance (SwA) working groups and forums, which seek to engage multiple communities working together to develop solutions and guidelines that reduce software vulnerabilities, minimize exploitation, and improve the routine development and deployment of trustworthy software products. Together, these activities will enable more secure and reliable software that supports mission requirements across enterprises and the critical infrastructure.

It is important to be aware of activities associated with safety and security and to ensure your organization has the capability to achieve your software assurance goals. To help with this awareness, the DHS SwA Processes and Practices Working Group has synthesized the contributions of leading government and industry experts into the set of high-level goals and supporting practices shown in Figure 1. Using these practices, organizations can identify their assurance practices implementation baseline.

The Assurance Process Reference Model [9] addresses assurance in the organization from executive to developer. It can be used to help organizations conduct a gap analysis of their existing practices versus industry recognized security practices. The results of the gap analysis can then be used to prioritize and track SwA implementation efforts.

What CMMI Says About Security

CMMI® models reflect what leading organizations do to acquire, develop, and sustain software-intensive products and services. It is not surprising that safety and security are addressed implicitly (and sometimes, explicitly) in CMMI models.

Safety and security are regarded as quality attributes in CMMI models. This term is mentioned extensively in the Acquisition Engineering process areas of the CMMI for Acquisition model [10], the Engineering process areas of the CMMI for Development model [11], and the Service System Development process area of the CMMI for Services model [12]. In particular, CMMI for Development includes coverage of quality attribute requirements (and thus safety and security requirements).

Safety and security are addressed at an abstract level. Such coverage makes sense in a CMMI model because it is rare that products need to be only safe or secure. Instead, desired quality attributes are addressed through careful architecture evaluation and tradeoffs. This more holistic treatment of quality attributes is essential to effective product design and is addressed in the 2011 SEI Webinar Capability Maturity Model Integration V1.3 and Architecture-Centric Engineering [6].

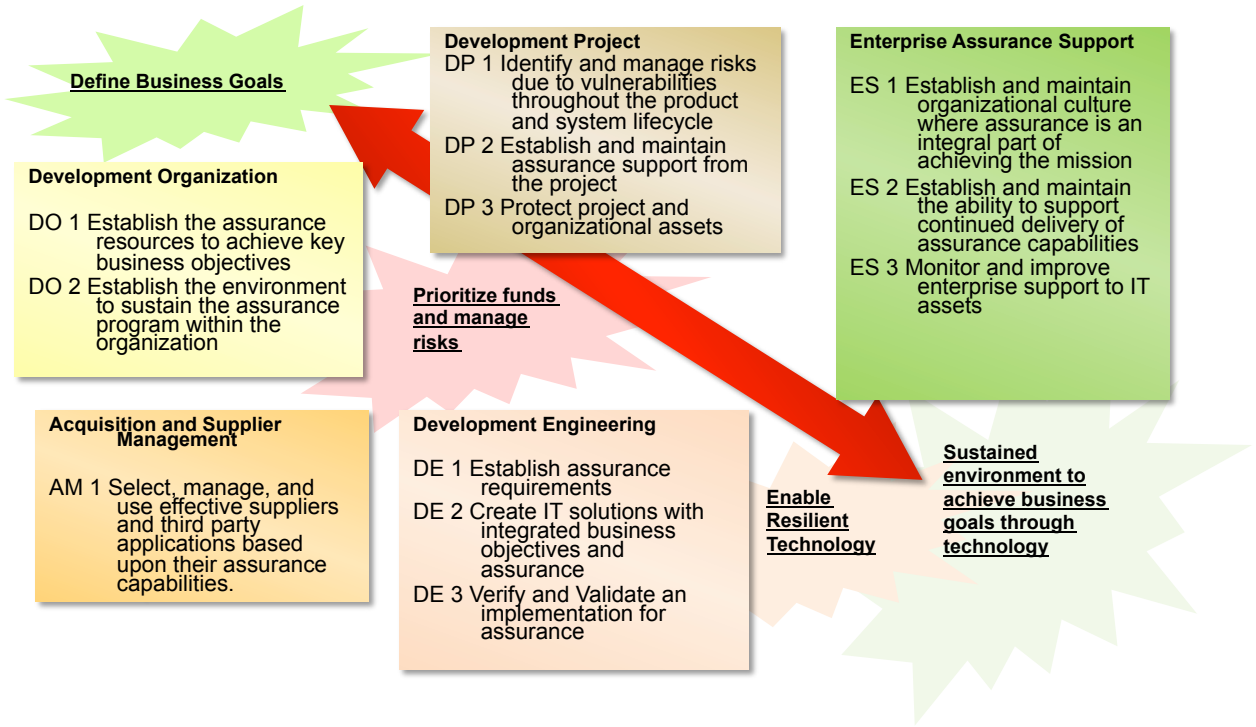


Figure 1. The Assurance Process Reference Model

Security (and sometimes safety) is explicitly covered in CMMI in the following:

- Example standards that are applicable to the organizational processes as listed in the Organizational Process Focus process area.
- A special section of the References that addresses Information Assurance/Information Security Related Sources.
- A discussion of information system vulnerabilities that appears in the Measurement and Analysis process area.
- Work environment standards in the Organizational Process Definition and Integrated Project Management process areas.
- Example quality attributes in the Engineering process areas.
- As a strategic consideration in the Project Planning and Work Planning process areas.
- Requirements and procedures that are considered as part of data management practices.

What Does This Imply?

CMMI addresses the critical broader (system) view of product and service scope, functionality, and quality attributes and how attention to all of these are necessary to make appropriate decisions and tradeoffs as the product or service is engineered and developed. However, CMMI does not provide specific guidance about individual quality attributes or information about their relationships. Safety and Security are addressed in an informative, not normative, manner. You must look elsewhere for explicit guidance about what to do in your organizational, team, or individual processes about safety and security because CMMI is relatively silent when your focus shifts from the overall balance of quality attributes to individual quality attributes.

What Additional Practices Help to Build Safe and Secure Products?

In recent years, multiple frameworks were developed that explicitly focus on practices and guidance for addressing safety and security. These frameworks can be grouped into three categories:

- Process capability frameworks.
- Secure lifecycle practices.
- Implementation approaches.

Process capability frameworks include:

- Resilience Management Model–a model that addresses converging security, business continuity, and IT operations in support of operational risk management [13].
- Assurance Process Reference Model–a model that synthesizes the contributions of leading government and industry experts into a set of high-level goals and practices to address software assurance (Figure 1) [9].
- Assurance for CMMI–a thread of assurance practices that can be overlaid on an existing CMMI implementation.[2]
- +Safe and +Secure[3]–white papers that extend CMMI models by providing a set of process areas specific to safety and security [14].

References for secure lifecycle practices include:

- Microsoft Security Development Lifecycle–a software development security assurance process consisting of security practices grouped into 7 phases: training, requirements, design, implementation, verification, release, and response [15].
- SAFECode–a global, industry-led effort to identify and promote best practices for developing and delivering more secure and reliable software, hardware, and services[4]

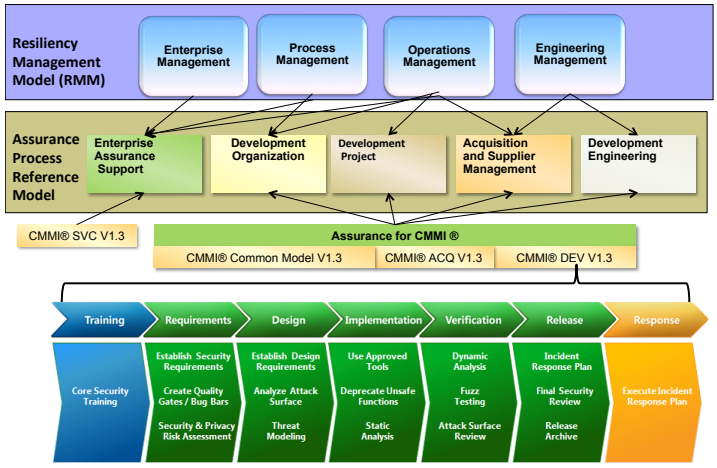


Figure 2. Configuration of Resources Addressing Process Improvement, Quality, and Security

Implementation approaches include.

- Open Software Assurance Maturity Model–an open framework that helps organizations to formulate and implement a strategy for software security that is tailored to the specific risks facing the organization [15].
- Building Security In Maturity Model–a descriptive model that describes the specific activities that organizations can engage in to improve and mature their software security posture [16].
- TSP Secure–an extension of the SEI Team Software Process (TSP) methodology that achieves the development of secure software systems by incorporating the planning, process, quality, measurement, and tracking frameworks of TSP and generating the practices and artifacts required to satisfy a maturity level 3 appraisal [17, 18].
- CERT Secure Coding Standards–a wiki-based website that supports a broad-based community of more than 500 contributors, including security researchers, language experts, and software developers [19].
- Open Web Application Security Project–an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted as well as to provide tools, documents, forums, and chapters free to anyone interested in improving application security [15].
- Build Security In–a collaborative effort that provides practices, tools, guidelines, rules, principles, and other resources that software developers, architects, and security practitioners can use to build security into software during every phase of its development [20].
- Strengthening Ties between Process and Security–a summary of key accomplishments in linking security, the SDLC, and process improvement, including an industry-led initiative to harmonize security practices with CMMI, the use of assurance cases, and NIST security considerations in the SDLC [21].
- Security Quality Requirements Engineering–a process model that provides a means for eliciting, categorizing, and prioritizing security requirements for information technology systems and applications [22].
- Correctness by Construction–a method of building software with demonstrable integrity for security- and safety-critical applications by combining formal methods and Agile development [23].

The challenge that many organizations face is defining their business goals and objectives with respect to safety and security. These goals and objectives, along with organizational policies and processes, are critical to identifying the regulations, standards, and best practices that are needed to enable organizations to build safe and secure products. It is important to tailor one or more frameworks, such as CMMI, to provide both a foundation and the flexibility for organizations to respond to internally or externally driven changes in business goals and objectives. Figure 2 illustrates how existing resources fit together to address process improvement, product quality, and security.

Summary

With the ever-increasing reliance on safety and security in products and services, CMMI provides a needed starting point, but it is not sufficient. A learning, knowledgeable, and resource-aware mindset is also required. A variety of approaches and tools are available to help you be successful. Very little of these approaches and tools are rocket science but their use requires a commitment by the organization. For a software product acquired through a supply chain, the acquisition mechanisms need to incorporate effective supply chain risk management to ensure the supplying organization is performing critical processes and practices.

As a starting point, identify the policies, standards, and business objectives that will drive excellence in your organization. CMMI and other lifecycle standards (e.g., ISO/IEEE 15288) provide the foundation and flexibility to build safety and security into an organization's lifecycle processes. You can then identify which of the available resources will best drive development of safe and secure products and services in your organization.

Process and product assessments are valuable to understanding potential vulnerabilities and risks. Organizations need to explicitly link their objectives to the assessments to use them effectively. The assessment results can help in the evaluation of resources that help in developing better products and services. There are many different approaches for addressing safety and security; no single approach addresses the needs of all audiences and organizations. The challenge for you is to pick the approaches that work best in your respective environments. This article provides an overview of some of the resources available because today most organizations are not taking advantage of the guidance that is freely available.

Acknowledgements:

We wish to thank Joe Jarzombek of the DHS; Don Davidson of the DoD; and Carol Woody of SEI, Carnegie Mellon University for encouraging us to write this article and for their helpful suggestions. Also, thanks to Winfried Russwurm and Peter Panholzer of Siemens AG for their work with +SECURE, and Stephanie Shankles of Booz Allen Hamilton for her review. Finally, thank you to Sandy Shrum, our editor, for improving our prose to say what we wanted more eloquently.

Disclaimer:

CMMI® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

NOTES

1. The term quality attributes is defined in CMMI models as "A property of a product or service by which its quality will be judged by relevant stakeholders. Quality attributes are characterizable by some appropriate measure. Quality attributes are non-functional, such as timeliness, throughput, responsiveness, security, modifiability, reliability, and usability. They have a significant influence on the architecture."
2. A pilot version of Assurance for CMMI was released in March 2009. Assurance for CMMI V1.3 has not yet been published.
3. Siemens AG, Corporate Technology released a draft report entitled +SECURE, V1.3, A Security Extension to CMMI-DEV, V1.3 in March 2012.
4. SAFECode, whose members include Adobe, EMC Corporation, Juniper Networks, Inc., Microsoft Corporation, Nokia, SAP AG, Siemens AG, and Symantec Corporation, displays its work on their website, <http://www.safecode.org>.

REFERENCES

1. Kahneman, Daniel. Thinking, Fast and Slow. New York: Farrar, Straus and Giroux 2011.
2. Forrester Research, Inc. Half of Companies Surveyed Report Web Application Security Problems. <http://www.networkworld.com/news/2012/091812-web-application-security-262520.html> (2012).
3. Office of the National Counterintelligence Executive (ONCIX). ONCIX Reports to Congress: Foreign Economic and Industrial Espionage.
4. Verizon, 2011 Data Breach Investigations Report, A study conducted by the Verizon RISK Team with cooperation from the U.S. Secret Service and the Dutch High Tech Crime Unit, <http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf> (2011)
5. Verizon, 2012 Data Breach Investigations Report, A study conducted by the Verizon RISK Team with cooperation from the Australian Federal Police, Dutch National High Tech Crime Unit, Irish Reporting and Information Security Service, Police Central e-Crime Unit, and United States Secret Service, <http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2012-ebk_en_xg.pdf> (2012)
6. Jones, Lawrence G. & Konrad, Michael D. Capability Maturity Model Integration V1.3 and Architecture-Centric Engineering.
7. Forrester Research, Inc. State of Application Security. <http://www.microsoft.com/en-us/download/details.aspx?id=2629> (2011).
8. Box, George E. P. & Draper, Norman Richard. Empirical Model-Building and Response Surfaces.
9. Department of Homeland Security, Assurance Process Reference Model (PRM), <https://buildsecurityin.us-cert.gov/swa/proself_assm.html> (2010)
10. Gallagher, Brian; Phillips, Mike; Richter, Karen; & Shrum, Sandy. CMMI-ACQ: Guidelines for Improving the Acquisition of Products and Services, 2nd Edition. Boston: Addison-Wesley, 2011.
11. Chrissis, Mary Beth; Konrad, Mike; & Shrum, Sandy. CMMI: Guidelines for Process Integration and Product Improvement, Third Edition. Boston: Addison-Wesley, 2011.
12. Forrester, Eileen; Buteau, Brandon; & Shrum, Sandy. CMMI for Services: Guidelines for Superior Service, 2nd Edition. Boston: Addison-Wesley, 2011.
13. Caralli, Richard A.; Allen, Julia H.; Curtis, Pamela D.; White, David W.; & Young, Lisa R. CERT Resilience Management Model, Version 1.0.
14. Defense Materiel Organization, Australian Department of Defense. +SAFE, V1.2: A Safety Extension to CMMI-DEV, V1.2
15. The Open Web Application Security Project. <https://www.owasp.org/> (2012).
16. McGraw, Gary et al. Building Security In Maturity Model. <http://www.bsimm.com/> (2012).
17. CERT. TSP-Secure. <http://www.cert.org/secure-coding/secure.html> (2010). [Davis 2009]
18. Davis, Noopur; Miller, Phillip L.; Nichols, William R.; & Seacord, Robert C. TSP Secure. <http://www.sei.cmu.edu/tspsymposium/2009/2009/DAY%203%20315%20PM%20TSP%20Secure.pdf> (2009).

ABOUT THE AUTHORS

Mary Beth Chrissis of the Software Engineering Institute (SEI), Carnegie Mellon University, is working with VA Health Systems to create a knowledge management system. She developed capability maturity models and training and co-authored multiple books and papers on process improvement. Chrissis chaired the CMMI Configuration Control Board, managed the CMMI Training Team, and instructs SEI courses. She received her BS from Carnegie Mellon University and pursued a MS in Computer Science from Johns Hopkins University.

SEI, Carnegie Mellon University
Phone: 412-268-5757
E-mail: mb@sei.cmu.edu

Dr. Mike Konrad of the Software Engineering Institute at Carnegie Mellon University leads two research efforts: one characterizing the economics of preventing vulnerabilities and the other orchestrating early software lifecycle activities across stakeholders. Previously, Mike served as the manager of SEI's CMMI Modeling Team (1994-2012) and as the CMMI chief architect. Mike has also worked with several software companies and universities. Mike obtained his Ph.D. in Mathematics in 1978 from Ohio University.

SEI, Carnegie Mellon University
Phone: 412-268-5813
E-mail: mb@sei.cmu.edu

Michele Moss of Booz Allen Hamilton, is a recognized thought leader in the integration and benchmarking of assurance practices. She is co-chair of the Department of Homeland Security (DHS) Software Assurance Working Group on Processes & Practices. She represents Booz Allen within the U.S. International Committee for Information Technology Standards Cyber Security 1 (CS1) technical committee and the U.S. Technical Advisory Group (TAG) for ISO/IEC JTC1/SC7. She is the liaison from SC7 TAG to CS1.

Booz Allen Hamilton
Phone: 703-377-1254
E-mail: moss_michele@bah.com

REFERENCES (continued)

19. CERT. CERT Secure Coding Standards. <https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards> (2012).
20. Department of Homeland Security. Build Security In: Setting a Higher Standard for Software Assurance. <https://buildsecurityin.us-cert.gov/bsi/home.html> (2012).
21. Woody, Carol. Strengthening Ties Between Process and Security. <https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/1049-BSI.html> (2008).
22. Mead, Nancy R.; Hough, Eric; & Stehney II, Ted. Security Quality Requirements Engineering. <http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm> (2005).
23. Amey, Peter. Correctness by Construction. <https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/613-BSI.html> (2006).

Software ID Tags Support Better Cyber Security

Steve Klos, TagVault.org
John Richardson, Symantec Corporation

Abstract. Would you fly on an airline that did not verify its passengers' identities, or that allowed high-risk passengers onto their flights? How about an airline that was unable to scan all checked luggage for known threats, or was unable to ensure that each piece of luggage was associated with a known, trusted passenger? This, unfortunately, is the position that IT organizations are placed into every day by their software applications. Software publishers do not typically provide secure, authoritative information that provides the critical information IT administrators require to validate the authenticity of their installed software applications and their executable files (program files, shared libraries, scripts, etc.). Typical computer systems such as a laptop with an operating system and a few software applications will have thousands of executable files installed on the system with no definitive way to authenticate these executable files, and to ensure that they have not been modified by any third party. To improve software supply chain security, IT organizations require standardized, authoritative software application information from software publishers that allows them to automate the following:

- Identifying all software applications, the operating system, their revision level, and all software updates and patches
- Associating all installed files to specific software applications, or to the operating system
- Validating that all installed software came from trusted software suppliers
- Validating the authenticity of each of the installed executable files

These capabilities are fundamental to securing computer systems. IT administrators must be able to automate these capabilities with a high level of confidence, and must be able to trust that their security tools can identify threats or known vulnerabilities quickly and definitively. However, without standardized, authoritative information provided by software publishers at the time software applications are released, it is remains very difficult for IT administrators to fully secure their computer systems.

The ISO/IEC 19770-2:2009 [1] Software Identification Tagging standard is a cross platform (Windows, UNIX, Linux, Mac) software identification data standard that provides the means for authoritative identification of software applications, operating systems, software updates, and patches. Tags also provide the means for:

- Validating the authenticity of install media
- Automating the authenticity validation of installed application executable files
- Automating the identification of installed applications with known vulnerabilities per the NIST National Vulnerability Database.

This article provides high-level information that outlines how software identification tags provide the fundamental building blocks required for building a resilient and automated IT cyber security ecosystem based on information that is very easily provided by software publishers.

Standardizing Software Identification in an IT Environment

Today's software and computing environments are large and complex. Software applications are difficult to identify and track properly. Non-standard techniques employed by software publisher for software updates and patches make software identification even more difficult. In particular, it can be extremely difficult to determine the software revision level and whether or not a software update has been properly applied to a system. These complexities make it extremely difficult for IT organizations to ensure the most fundamental aspects of cyber security—mainly:

- Ensuring that all of the software deployed in the IT environment is authentic, unmodified, and from a trusted supplier
- Ensuring that all software is patched and that all known software vulnerabilities have been mitigated

This article covers the following four critical software supply chain security areas, and compares these areas to the security requirements of airline travel:

1) Authoritative Identification:

Only authorized applications from trusted suppliers are installed (verified passengers).

2) Application Associations:

Correct versions, patches and third party components are installed and related to their parent applications (all bags can be associated with passengers).

3) No Corruptions Allowed:

Installed files or third party components have not been modified (no unknown passengers).

4) No Known Threats:

Known application vulnerabilities have been resolved (passengers and bags do not have any known threats).

Software ID Tags–What Are They?

The missing link in today's computing environment is standardized, authoritative information provided by software publishers that allow IT organizations to confidently automate a process that allows only trusted applications to be deployed in their environment, ensure that all application executable are authentic, and validate that the correct software updates have been applied. The International Organization for Standards (ISO) published the ISO/IEC 19770-2:2009 Software Identifica-

tion (SWID) Tagging standard enabling software publishers to provide standardized, authoritative, and secure application identification information in a consistent and secure format enabling security and asset management tools to authoritatively identify installed applications, components, and patches, and associate these items with files installed on a computing device.

This article does not provide detailed explanations of SWID tags—instead, it focuses on how authoritative software identification provided by SWID tags should be used to support other security standards and improve overall cyber security processes. For more detailed information on SWID tags, refer to the TagVault.org website [2].

SWID tags are not silver bullets that solve all cyber security concerns. Rather, SWID tags provide the necessary building blocks upon which the IT community can build a more secure infrastructure. Each of the following sections covers a different aspect of security capabilities supported by SWID tags. Providing SWID tags with the necessary data to provide all four critical functions ensures a much higher degree of authoritative and security related IT capabilities. Additionally, material covered in this article is focused on security of known software titles from trusted publishers. If a computing device includes unknown software installations, SWID tags can identify that fact, but will not provide more than the identification of unknown elements.

1) Authoritative Identification

When flying, a passenger must carry and present government issued identification documents. This provides a level of validation that the person carrying the document is who they say they are. This type of security relies on a trust model of a third party validating the identity of the passenger and some method to present that validation information to an unknown third party (a security screener validating a driver's license for example).

When it comes to software, an organization that publishes software must validate who they are and that they have the necessary trust to "digitally sign" SWID tags. This is done through a certificate authority that validates that an organization is real and that it represents (i.e. owns) the organizational name.

Once a certificate authority certifies a digital certificate for a software publisher, the publisher can sign data with a private key that can be validated using the corresponding trusted public key as shown in Figure 1. SWID tags build on this trust model by allowing publishers to digitally sign a SWID tag that is associated with their software.

Signed SWID tags must also include a timestamp from a trusted timestamp server. This ensures two things—first, that the signed data cannot be modified by anyone (even the publisher) after the timestamp is applied without a third party being able to identify that data has been modified. Second, it allows a third party to identify that the

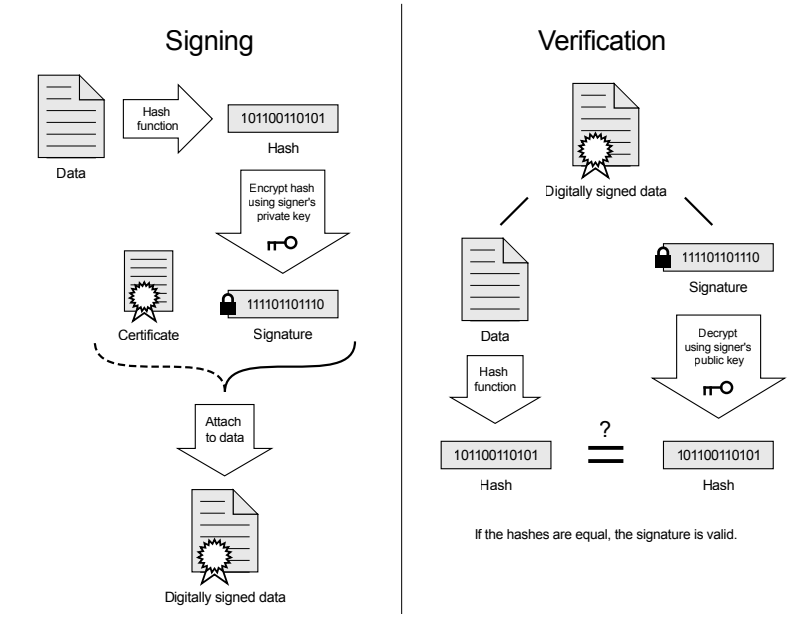


Figure 1: Digital Signatures and Digital Signature Verification [3]

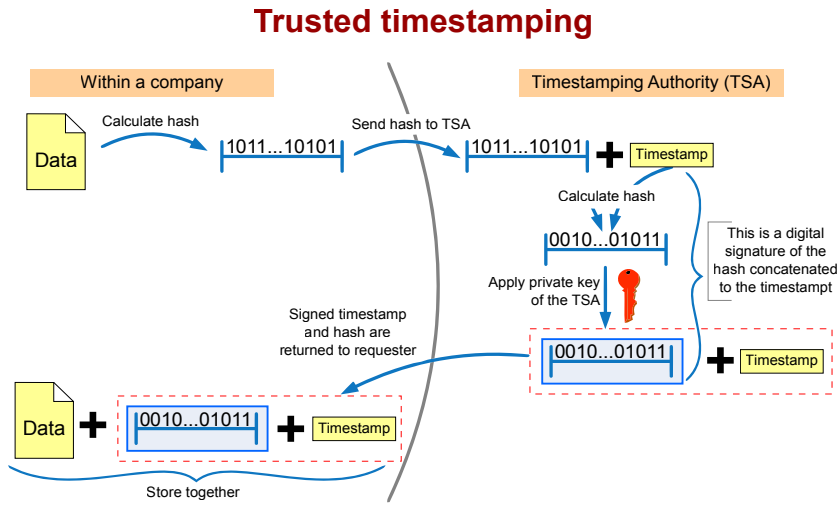


Figure 2: Image of Trusted Timestamping process [4]

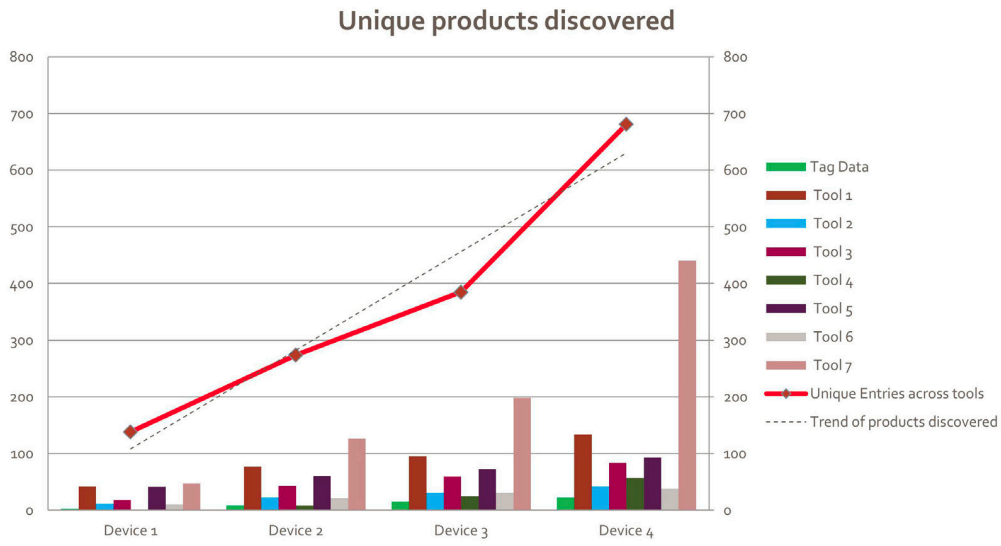


Figure 3: Unique Products Discovered in Discovery Tool Analysis [5]

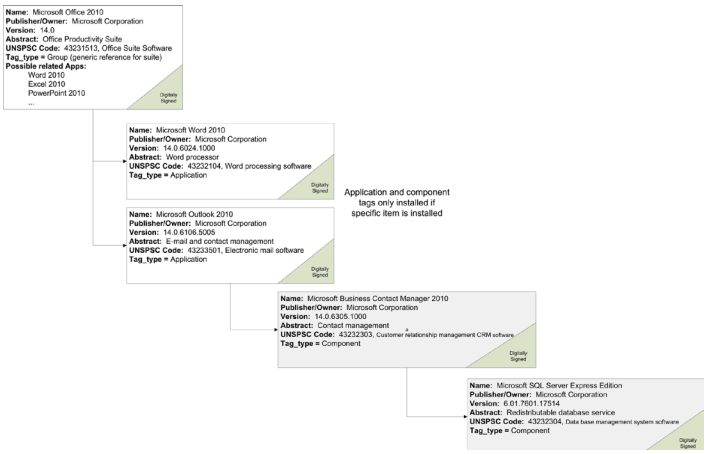


Figure 4: Diagram representing relationship of suite, applications, components and redistributable components.

signature was made during a time period when the certificate authority certifies the validity of the digital signature.

After a SWID tag has been digitally signed and timestamped, a third party can validate that the SWID tag was provided by an authoritative source during a time when the certificate authority indicated that the digital certificate was valid.

This level of trusted identification is required by any organization that needs authoritative data about the name of a software product, who published the product and which files the product installed. In short, this is the trusted identification information organizations must have to manage their IT environments securely.

Summary

Similar to passengers on an airline being required to identify themselves using government issued identification documents, software products require a similar level of identification. SWID tags that are digitally signed by the software publisher and include validation from a certification authority provide this trusted data.

Requiring a digitally signed SWID tags provides the ability for end-user organizations to validate that the software they have installed in their organization are the titles they expected and that they came from the publishers they expected and not some unknown publisher.

2) Application Associations

When an individual flies on an airline today, the airline must ensure that the owner of checked luggage travels on the same flight as the luggage. Providing the association of luggage to passengers is important to airline security as well as to the passenger who expects to have their luggage returned to them at the end of the flight.

A typical Microsoft Windows computer will have thousands of executable files (*.exe), and tens of thousands of shared libraries (*.dll, *.ocx, etc.) with a single application potentially responsible for the installation of hundreds of these executable files. Some of these files may be created and owned by the publisher, some may be redistributed versions of software from another publisher. Today, it is nearly impossible to associate every executable file with its parent application. Additionally, if files from another publisher are redistributed, it is impossible for the publisher of the software to validate that the files in the redistribution package are authentic without support from SWID tags.

TagVault.org did an analysis of software identification tools using a very simple test case. The test utilized different installations of 22 currently supported, separately licensable products from nine different vendors. The test pulled results from six different software discovery tools. The graphical results in Figure 3 provide a clear indication why application associations are so critical to get right when dealing with software discovery data.

Basically the number of unique products discovered exploded from what would be a reasonable expectation of 22 different products to a total of 700 unique names across the various discovery tools. A large part of this problem has to do with the fact that there is no consistent method to identify software and the range of options produces wildly different collections of product names.

The keys from a security perspective require that a security operations manager needs to know:

- Which applications are related to a particular bundle or suite (for example, Word, Excel and PowerPoint are applications that could be stand-alone, or part of a Microsoft Office suite).
- Which components are related to a particular application (for example, Microsoft SQL Server Express is used by Microsoft Business Contact Manager which is a component of Outlook which is an application included in the Microsoft Office Suite).
- Validation that a redistributable component came from a known and trusted 3rd party and not some other, unknown, or unexpected entity (for example, was a C++ runtime library actually a redistributable library provided by Microsoft).

SWID tags provide these capabilities and more through the fact that each publisher provides their own digitally signed SWID tags. When a publisher such as Microsoft provides C++ runtime libraries in a redistributable package, they provide a digitally signed SWID tag indicating that the C++ runtime is owned by Microsoft and the vendor that is redistributing the runtime can reference this SWID tag as a "child" product providing the association. By using this method, security operations can readily identify parent/child relationships and be able to automatically group application and component data that would otherwise show up as independent applications.

Obviously, in highly secure environments, there will be a need to validate these relationships and the security of each item, but once validated the security policy rules can be applied to the inventory data collected from all devices in the organization.

Patches have a similar problem. When a patch is released by a publisher and the organization determines that it makes sense to install the patch, the system inventory can be used


SUBSCRIBE TODAY!
To subscribe to CrossTalk, visit www.crosstalkonline.org and click on the subscribe button.
CrossTalk
The Journal of Defense Software Engineering
Vol. 24 No. 2
Vol. 26 No. 1
Jan / Feb 2013
SOFTWARE PROJECT MANAGEMENT
LESSONS LEARNED
Enjoy!
Jan / Feb 2012
July / August 2011
Vol. 24 No. 4
Vol. 25 No. 5
March / April 2013
Vol. 24 No. 1
Nov / Dec 2010
Vol. 23 No. 6

to immediately identify every device that requires the patch. This is due to the fact that the patch's SWID tag includes details for which software products it applies to. Obviously, validating that a patch is installed is as easy as checking that the SWID tag for that patch is installed. In highly secure environments, additional validations will be applied, such as ensuring that the publisher's name for the patch matches the publishers name for the application being patched.

Summary

Just like airlines have the ability to associate a passenger's checked bags with the passenger, software product and component relationships must be detailed by the publisher. These relationships need to be included even if software from a third party is included in an application's installation routine. This is not difficult, nor is it expensive for publishers to provide, it simply has not been a requirement that software purchasers have made to their vendors.

Requiring the relationships between applications or components to be specified in SWID tags (including those that may be redistributed from a third party) ensures that security operations can validate the items are related to each other and that any redistributable components were, in fact, provided by the publisher indicated.

3) No Corruptions Allowed

Airlines must ensure that only passengers who have purchased tickets and who have gone through security screening are allowed on a flight. It is particularly important to validate that there are no unknown or high-risk passengers allowed on a flight.

This requirement extends to the files that are installed by a software product on a computing device. Security operations must ensure that the files installed on a device are not corrupted (i.e. unknown) or malicious (i.e. high risk) before installing and should be able to validate those details in real time.

There are two areas where file corruptions (regardless if they are an accidental or malicious) can occur and must be identified. The first is on the supply side—being able to validate that the software publisher's distribution of a software installation package is exactly what the publisher shipped, and has not been modified by a man in the middle attack, is critical to secure systems. In these cases, a SWID tag with the complete installation file manifest that is digitally signed by the publisher and that includes secure hash values for every file is required. This allows an organization to validate that the files shipped by the publisher are exactly the same as the files received by the organization with no modifications.

The second issue dealing with file corruptions has to do with ensuring files that are installed on a device for a software product are known files that are not corrupted. In this case, the SWID tag must include a digitally signed file manifest that includes a secure file hash as part of the file list. Obviously, some files that are installed for a software title will be modified once installed (configuration options, data files, etc.)—and these files cannot include a trusted secure hash, however, at a minimum, the executable files for an application must be included in the manifest and must include a hash that can be validated.

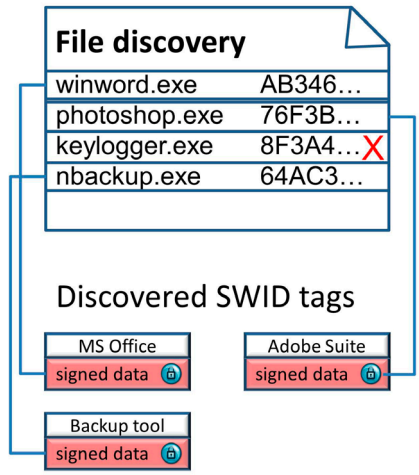


Figure 5: Conceptual graphic of file manifest used to identify malicious executable files.

Some software components (for example, Windows device drivers) need to be digitally signed for the operating system to trust them. Unfortunately, as was seen with the Stuxnet malware, simply requiring a digital signature for the operating system to trust a driver is not sufficient. In the case of Stuxnet, the drivers that were part of the malware payload were digitally signed, but were signed by a publisher other than the publisher that provided the software [6]. If the software had included a digitally signed manifest from the publisher, it would have been significantly more difficult for the Stuxnet malware to avoid detection as the changes it made to the application by replacing a core shared library with a malicious one, would have been detectable by ISO 19770-2 compliant scanning tools.

Summary

Just as airlines and security requirements would not allow unknown people or unscreened baggage onto a commercial flight, IT Security Managers should not allow unknown applications or executable files to be installed on their network. The details required to securely provide file manifests for software installation media as well as executable files that are installed on a device are not difficult or expensive to provide if the process to create the manifest is integrated into a product build cycle. The only organization that can do this in any cost effective manager is the publisher themselves.

Requiring SWID tags with secure file manifests to be included as part of the installation media (to ensure supply side security of the distributed files) as well as for the installed software allows security operations to validate applications, components and patches to any level of detail required.

4) No Known Threats

Airlines would not allow passengers to board with carry on or

checked luggage that contained dangerous materials. All baggage is screened to validate that potential risks to any flight are minimized.

The software environment in most IT operations is much more dynamic than is the case for a piece of luggage that can be screened once and assumed to remain safe as long as it remains in a secure environment. With higher complexity levels inherent in today's software, publishers are regularly providing patches and updated configuration guidance to minimize the potential for security breaches. Unfortunately, the process by which patches and/or configuration changes can be identified as being required is often a very manual and time intensive process.

Patches to software products must have the same type of secure file manifest provided so that as the patch changes an executable file, security monitoring systems can identify that a patch made the change and it is not due to a potentially malicious change to a file.

The requirement to include secure file manifests with applications, components and patches provides a very positive side-effect that can be easily implemented for IT environments— organizations have enough data that IT processes can validate that any of these items are, in fact, installed on a device. If the SWID tag is installed on a device and it contains a secure manifest, a process can validate if the files are actually installed.

Summary

Just as all baggage destined to fly on a commercial airplane must be screened and declared safe, software must be validated to ensure it is up-to-date with no outstanding patches or configuration changes required. This can be done by ensuring that patches identify the software products they are targeted at as well as by providing secure file manifests to identify that a specified publisher provided the patch and that the files have not been modified.

Requiring patches to include SWID tags will not add to the complexity or cost of software development efforts if the procedures are integrated into the proper process of a patch build environment. The benefits to end-user organizations and tools that manage IT operations are very significant and can allow a much higher degree of security that simply is not possible today.

Secure Software Requires Secure SWID Tags

IT environments today are getting more and more complex. With this complexity, cyber security issues related to software installations, patches and configurations are skyrocketing. With commercial organizations, national infrastructure systems and national security related systems becoming increasingly connected, it is clear that software must include a trusted and authoritative identification capability in a standardized, normalized format. The cross platform (Windows, UNIX, Linux, Mac) and cross vendor capabilities enabled by the ISO/IEC 19770-2:2009 Software Identification Tagging standard must be embraced by the software engineering community as a software assurance best practice for commercial and internally developed applications. Until the community makes these requirements, IT environments will continue to struggle with effectively managing and securing their software applications.

ABOUT THE AUTHORS

Steve Klos is the executive director of TagVault.org, a program of IEEE-ISTO. He is also the convener of ISO/IEC 19770-2:2009 and a member of the ISO/IEC JTC1/SC7 US Technical Advisory Group (TAG) and Work Group 21 (WG21 – targeting SAM Standards). Steve is also the recipient of multiple industry awards and certifications in Software Asset Management including being an IAITAM Fellow and a Microsoft Certified Professional with a Software Asset Management Competency.

Phone: 735-562-6031
E-mail: stevek@tagvault.org

John Richardson has a background in software engineering, software engineering management, and software licensing. He has been involved with ISO Software ID Tagging since 2008 as the Symantec representative, helping to establish industry implementation standards and certification requirements, working with U.S. government agencies on approaches for integrating SWID tags and SCAP, and providing tools and process support to Symantec engineering teams as they integrate SWID tagging into their products.

Phone: 919-213-4027
E-mail: john_richardson@symantec.com

REFERENCES

1. ISO/IEC 19770-2:2009 Information technology -- Software asset management -- Part 2: Software identification tag, published by ISO - <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53670>.
2. <http://www.tagvault.org>
3. Image Source – Acdx - <http://en.wikipedia.org/wiki/File:Digital_Signature_diagram.svg>
4. Image Source - Bart Van den Bosch - <http://en.wikipedia.org/wiki/File:Trusted_time stamping.gif#file>
5. Image Source – TagVault.org White paper - <http://www.tagvault.org/discovery_tools>
6. W32.Stuxnet Dossier - <http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf>

Upcoming Events

Visit <http://www.crosstalkonline.org/events> for an up-to-date list of events.

Software Assurance Forum - March 2013
5-7 March 2013
Gaithersburg, MD
https://buildsecurityin.us-cert.gov/bsi/events/1417-BSI.html

Government Contracting
12-13 March 2013
Washington, DC
http://publiccontractinginstitute.com/events

Conference on Systems Engineering Research
19-22 March 2013
Atlanta, GA
http://cser13.gatech.edu

7th International Symposium on Service Oriented System Engineering
25-28 March 2013
San Francisco, CA
http://sei.pku.edu.cn/conference/sose2013

Symposium of Mobile Cloud, Computing, and Service Engineering
25-28 March 2013
Redwood, CA
http://www.engr.sjsu.edu/gaojerry/IEEEMobileCloud2013/index.htm

Software Technology Conference
8-11 April 2013
Salt Lake City, UT
http://www.sstc-online.org

7th Annual IEEE Systems Conference
15-18 April 2013
Orlando, FL
http://ieeesyscon.org

Cloud Computing and Assurance for Critical DoD Initiatives
23-25 April 2013
Washington, DC
http://www.marcusevans-conferences-northamerican.com/cloud_2013

Systems Engineering, Test and Evaluation Conference
29 April – 1 May 2013
Canberra, Australia
http://sapmea.asn.au/conventions/sete2013

IBM Edge 2013
10-14 Jun 2013
Las Vegas, NV
http://www.ibm.com/edge

23rd Annual INCOSE International Symposium
24-27 Jun 2013
Philadelphia, PA
http://www.incose.org/symp2013

Software Assurance Working Group Sessions
25-27 Jun 2013
McLean, VA
https://buildsecurityin.us-cert.gov/bsi/events.html

Risk Reduction Through Inactivity

Once again, I find myself writing a BackTalk column sitting in an airplane. I live about two hours from an airport so between the joys of getting up early on a Saturday, traffic, airport parking, crowded check-in lines, totally full flight, and only a one-hour delay in taking off—I am just in a great mood.

Back in the day, this column was not called "BackTalk." It was called "The Curmudgeons' Corner." A curmudgeon is defined as a killjoy, wet blanket, or a grouch. This column was a place to air gripes and general displeasure about software, bureaucracy, and the general process of producing software. There used to be several authors—and all of us were grouchy. Over time, several of the authors decided to pursue other areas of literary achievement. You know what I think? They ran out of things to gripe about. Luckily, the publishers still have me around. I can always find something to complain about—such as change.

The story goes that a new assistant at a grocery store noticed that hardly anybody was buying rutabagas. It was the lowest-selling item in the produce department. He decided to remove the entire display of rutabagas. When the general manager noticed that the rutabagas were missing, he spoke to the assistant

and asked why. The assistant explained, and expected the general manager to compliment him. Instead, the general manager, with an irritated look, asked, "Well, why stop there? After all, now something else is the lowest-selling item!"

With that epiphany, the assistant realized that he could always remove the lowest-selling vegetable, until nothing was left but apples and unhappy customers. With this realization, rutabagas were restored, and the system was left the way it was. Leaving things the way they are does not imply you do not care—it might also mean that the current system works well enough.

Back in 1998, I became a Personal Software Process (PSP) instructor. I had to take the class (and pass!) before I could be certified to teach. Overall, I learned a lot, and my coding quality and speed really improved. Part of the process was submitting a Process Improvement Proposal (PIP) every time as I completed the exercises. The PIP was to force me to come up with an improvement on my personal process for developing software. And I rebelled. I submitted several, but eventually I reached the point where I was pretty happy with my own process, and did not really see a critical need to improve. Nevertheless, my instructor demanded that I submit a suggested improvement for each remaining program. As I remember, my last few improvement proposals consisted of things like, "Keep a pencil sharpener closer to my desk," "Use higher-wattage bulbs during design," and my all-time favorite, "Stock up on beverages, chips and popcorn prior to coding." I eventually passed—and am still pretty pleased with my coding process.

I am not saying PSP did not help me—it did. What I am saying is that eventually I reached a point where things worked for me. Why change what works? When you modify a process, it involves risk. The risk of changing things is that somebody might be less satisfied. Here are two similar "risk rules" I have learned over the years:

1. You cannot make all the customers happy. In fact, you really probably cannot make most of them happy. What you can do is try not to make too many of them very unhappy. Sometimes, nobody is happy. But perhaps few are miserably unhappy.

2. For customers, it is much easier to wait until a change occurs and then complain, rather than being proactive about what they need up front. There are lots of reasons for this—but the main issue is that customers do not really know what they want—but they are quick to realize what they do not want.

Sometimes the best thing to do is to change nothing! Those of us who work for the government or any large organization know the golden rule of change, "Change is often substituted for progress." Need to look occupied? For heaven's sake, change something. Cannot find something to change? Then it is probably time to reorganize. To the outsider, it looks like progress.

Do not wake a sleeping baby. Do not change a process that minimizes the number of unhappy customers. Sometimes the least damage you can do is not change a single thing. Before making changes, weigh the advantages of leaving things the way they are. It might make some customers unhappy, but it also might make less of them very unhappy than any other action. You are not being complacent—you are being passively proactive!

And I plan on being a curmudgeon for a long, long time.

David A. Cook
Stephen F. Austin State University
cookda@sfasu.edu