Malware, "Weakware," and the Security of Software Supply Chains

C. Warren Axelrod, Ph.D., Delta Risk

Abstract. Increasing effort is being made to build security into software—but with mixed results. The need for security apparently exceeds the ability and will of software engineers to design secure software architectures, implement secure coding methods, perform functional security testing, and carefully manage the installation of software products on various platforms and in different environments. COTS/GOTS software often harbors numerous vulnerabilities (we will call such software "weakware") and such software occasionally contains "malware" (malicious software). The main difference between weakware and malware is that the weaknesses of the former are mostly unintentional or accidental, whereas the damaging characteristics of the latter are planned and intentional and usually require some measure of technical expertise to implement effectively. Nevertheless, from the security perspective the potential consequences are undesirable and damaging irrespective of how the weak or bad code got into the program in the first place. In this article, we examine how and where such damaging code or programs might be introduced throughout the software supply chain lifecycle and how such weakware and malware might be avoided, deterred, eliminated or mitigated.

Introduction

In general, software products are either purchased from software manufacturers or distributors, or they are built in-house and/or by third parties, such as contractors. Homegrown, off-the-shelf and open-source software modules are regularly combined to form overall working systems. Usually customers do not think about software as being produced via supply chains, although it often is since the manufacture of software involves a series of interrelated steps and is made up of components, many of which are acquired from subcontractors or other vendors. Custom software may be built in-house using contractors, farmed out to software consulting firms, which often further subcontract various phases of the software development lifecycle to other parties at home and abroad. System integrators incorporate off-the-shelf programs, particularly operating systems, system utilities, and the like, into working software systems. Software support and maintenance may be handled by software manufacturers, distributors or service companies. Support is now mostly provided over the Internet. Open-source software, developed and supported by communities and specialized service providers, incorporates its own version of supply chains, which is even more widely distributed over community members and geographies.

It is an enormous and complex task for customers to identify various channels through which software has passed and, for the most part, attempts to determine the structure and components of software supply chains have been thwarted by lack of knowledge and cooperation.[1] Yet this phase of the project is critical if risks are to be identified, assessed, mitigated and managed. When such channels are known, then customers will be confronted with trying to measure security risks for each component—a field where suitable metrics are sorely lacking and cooperation from suppliers is wanting. Even if one is able to assess these risks, one's ability to control them is hampered by the customer or end-user not having sufficient influence and control over supply chains that have been identified.

In this paper, we present some definitions to help us understand the various contexts in which software supply chain risks are experienced. We also develop a framework against which to identify and assess the risks. And finally, we point to ways in which the risk management process can be facilitated.

Supply-related Risks

One definition of supply chain risk as it relates to physical products is as follows:

Supply risk is … the probability of an incident associated with inbound supply from individual supplier [or market] failures, … in which its outcomes result in the inability of the purchasing firm to meet customer demand or cause threats to customer life and safety [1].

In addition to the above, we must consider specific software-related threats, which are far less tangible with respect to their impact. These include unauthorized and malicious access to intellectual property and sensitive data, and damage or destruction of applications or data. Also, piracy issues arise from phony software [2].

Supply Chain Risk Management generally addresses limiting the risk of disruptions, typically those that delay deliveries of an item to a manufacturer or consumer both for physical products, such as pharmaceuticals, luxury goods and entertainment media, and for software. However, Supply Chain Risk Management is also used to effect the reduction or elimination of counterfeit and/or malicious software during the supply chain lifecycle when insiders and others may have access to the software [3].

Software supply risk relates both to the impact of adverse events and the probability of those events occurring. Potential consequences include those that:

- Prevent the purchaser from meeting customer demand
- Prevent the supplier from providing contracted technical and operational support
- Compromise the supply chain management processes causing disruption and delays in meeting deadlines, diversion of products, theft of products, unauthorized copying and distribution, and the like

Types of Software

The following categories of software are of interest:

- COTS/GOTS software
- Open-source software
- Custom-built software (developed internally, externally or both)
- Hybrid—Combination of custom, open-source and off-the-shelf (OTS) software
- Embedded software—software or firmware built into physical products
- Supply chain management software—a specialized category of software that monitors supply chain processes and reports deviations from expected behavior

Software products are closed or open with respect to access to their source code by customers or other parties. Ordinarily, source code of off-the-shelf software is not available to customers. Users of both open-source and custom-built software usually have access to source code, which allows for code reviews and static testing. Hybrid software should generally be considered to be as weak as its weakest component, which is often thought to be a shrink-wrapped product, although studies have shown that open-source software can have as many and as severe security issues as COTS/GOTS software. Even though embedded software may not be top of mind for manufacturers and distributors of physical products, software-specific risk factors must be considered.

Software that is used to manage supply chains (whether the supply chains are for software, physical goods, or combinations of software and hardware) also needs to be considered since effective supply chain management can mitigate many risk factors normally encountered. However, such supply chain management software can also be a vector for cyber attackers. Few researchers appear to have considered the risk of compromise of supply chain management software, which might, for example, be made to report that everything is in order when it is not, and therefore could represent a significant risk.[2]

Another security category, occasionally considered in the literature, involves software used by computer chip foundries to produce complex integrated circuits. If hackers can gain access to such software, then they can change the circuit designs to perform nefarious functions with little chance of detection.

Risk Characteristics of Software

Software differs from manufactured products in several important ways, as follows:

- Software can be stolen without having to remove or otherwise change the original copy
- Physical transportation of software is not required since copies can be downloaded electronically
- Valid and lawful versions of software can be modified and still be made to appear valid even though they have been tampered with

For physical products, risks relating to manufacturing and distribution usually predominate. When it comes to software, more emphasis needs to be placed on the early phases of the product development process, such as the design and requirements phases, since manufacturing and distribution represent much smaller parts of the overall software supply chain than they do for physical products [4].

The following software supply chain attributes are at risk [5]:

- Confidentiality (intellectual property and personal and business data)
- Integrity (processes, products and data)
- Availability (flows, products and data)
- Authenticity (products and data)
- Trustworthiness (processes, products and people)

With respect to confidentiality, not only must one consider the potential risk of someone stealing intellectual property and trade secrets, but also one must be aware of the potential consequences of compromise of customer and employee personal data. When it comes to integrity, one can imagine the supply chain processes themselves being exploited by criminals, as well as the modification of software products and related data.

One must also be able to demonstrate that the mitigation efforts have been effective. The following properties enable one to have greater confidence that the risks have been adequately mitigated: transparency, quality, and accountability.

A report by the DoD Information Assurance Technology Analysis Center suggests that constituents are subjected to various supply chain threats [6]. Table 1 assigns threats to constituents.

An SEI (Software Engineering Institute) report points to similarities and differences between product suppliers and system development contractors [7]. The report notes that acquirers' assessments of software takes place after the product development is completed, whereas for custom-built systems, acquirers are able to "actively monitor both contractor and product supply chain risks during the development process."

Threats	Supply Chain Constituents					
	Products	Supply Chain Processes	Product Flows	Supply Chain Data Flows	Management Data	People
Sabotage	X	X		X	X	
Tampering	X			X	X	
Counterfeiting, piracy	X			X	X	X
Theft	X	X		X	X	
Destruction/deletion	X			X	X	
Disruption/delay		X	X	X	X	
Exfiltration—theft			X	X	X	
Exfiltration—disruption		X		X	X	
Infiltration, subversion		X		X		
Diversion			X	X	X	
Export control violations		X	X			X
Undesirable physical items	X		X			X
Corruption		X		X	X	X
Social engineering		X	X	X	X	X
Insider threat	X	X	X	X	X	X
Pseudo-insider threat	X	X	X	X	X	X
Foreign ownership, influence	X	X	X	X	X	X

Table 1: Assignment of Threats to Supply Chain Constituents

The report suggests that risk analysis include the following three components:

- Attack analysis, i.e., analysis of threats and exploits leading to successful attacks
- Ability to limit product vulnerabilities by supplier
- Identification of "attack enablers" and business risks by acquirer

Table 2 illustrates how software systems are combined and the characteristics of the combined systems.

The management of risk will vary with the various phases of the supply chain or acquisition lifecycle. The SEI report enumerates those activities as they relate specifically to security risks. Table 3 assigns such activities to the lifecycle phases. A number of activities have been added to the original list in the SEI report.

Forms of Combination	Characterizations
Embedded software	Many software products and systems contain software within the product or system about which acquirers might not be aware. [3]
Integrated systems	Software products are inserted into an existing environment and integrators ensure that the new software is compatible with the existing environment and validate that the combined functionality satisfies requirements.
Systems of systems	Disparate systems are combined to form systems of systems, which produce functionality that is greater than the sum of the individual systems.
Cyber-physical systems	Existing and/or new distributed information processing systems and networks and previously isolated industrial control systems are connected so that the control systems can be accessed over public and private networks and data from the control systems can be accessed over public and private networks [8].

Table 2: Combinations of Systems and Their Characterizations

SDLC Phase	Risk Management Activities
Requirements and design	• Perform a risk assessment. • Establish security requirements. • Develop auditing plans.
Manufacture (development)	• Monitor processes and product flows. • Inspect, test, verify and validate final products.
Distribution	• Monitor processes and product flows.
Warehousing	• Monitor processes and product flows. • Check that the product has not been removed, substituted or added.
Deployment	• Monitor processes and product flows. • Check that delivered products and systems are correct and authentic. • Provide user guidance to ensure that products and systems are not adulterated or otherwise compromised.
Operation	• Monitor operation for unusual behavior and damaging events. • Review operational readiness on a continuing basis. • Develop and implement a plan for responding to security incidents.
Maintenance and support	• Monitor suppliers of products and components for any adverse reports relating to the viability of supplier companies or any security or safety issues with products. • Develop contingency plans for potential disruptions in supply of parts or patches, for example, and support.
Disposal	• Monitor disposal of intellectual property and sensitive data, such as personal information and health data, and destruction of media containing such information.

Table 3: Supply Chain Security Risk Management by SDLC Phase

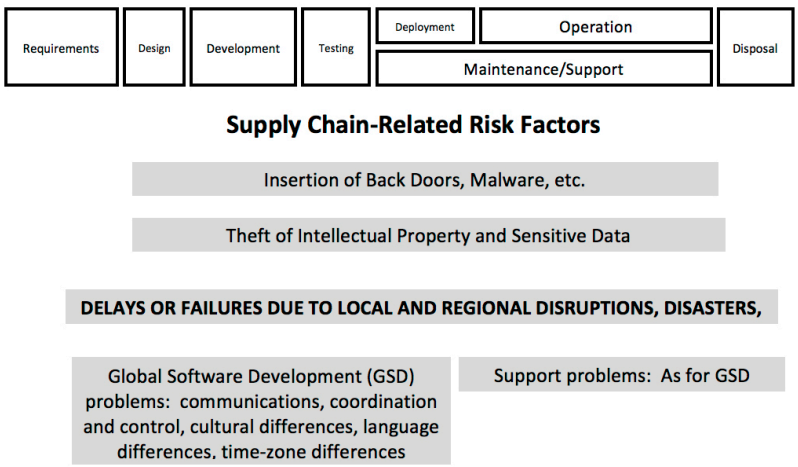

Figure 1: Supply Chain Risks by Phases of the SDLC

It is particularly difficult to get a full picture of all the complexities and nuances of global supply chains as they consist of so many constituents and components. Perhaps the only effective means of doing so is to build computer models of the processes, flows and controls, and use them in exercises to better understand the interaction of the components and the impact of various attacks and events [9].

Software Supply Chain Risks

A major differentiator, with respect to those risk factors related to software design and development, is the location of those efforts and the culture of those doing the work. Location can be defined in terms of whether the design and development is done internally or is outsourced, as well as by geographic location.

The level of risk varies greatly with factors such as loyalties and motivations of employees and contractors; legal, social and economic differences across countries and ethnic groups; and so on.

Figure 1, which is based on [10], illustrates factors affecting supply chain risk throughout the development lifecycle. As can be seen from the diagram, there are events, such as natural and man-made disasters, that can affect all supply chains including software supply chains. However, there are also a number of compromises, such as the insertion of malware, that are unique to software. Other incidents, such as the theft of intellectual property and personal data, are common across many products, but are facilitated for software by the ability to copy software and data without changing the original or having to be onsite to do the copying.

Simulation Models

There have been several research efforts relating to resolving issues relating to global software development and risk relating to software supply chains. Simulation is needed to optimize various characteristics of dispersed software development efforts due to the complexity and dynamic nature of such arrangements.

Researchers have developed models for the impact of cultural, communications and other factors on the distributed development of all types of software. For example, the U.S. financial services sector has worked on supply chain issues and surveyed industry members with respect to various aspects of supply chain risk mitigation. However, there does not appear to be much in the way of modeling the impact of adverse natural and human-invoked events on software supply chains. The need for such models is evident, but the effort to develop such models is substantial.

Software Development and Distribution

To make decisions with respect to any particular supply chain it is necessary to understand each phase as well as the interaction between phases. Figure 2 shows the lifecycles for three major aspects of software development, testing and deployment; namely, manufacturing, oversight, and assurance. Manufacturing is the "nuts and bolts" of developing and distributing software. Oversight consists of the independent oversight of the processes and products of the manufacturing lifecycle. Assurance includes separate evaluations of the quality, integrity and trustworthiness of the software being manufactured.

The shaded boxes in Figure 2 represent functions that are frequently given inadequate attention in the SDLC. Among these important areas are functional security testing, which is testing performed to ensure that the software does not do that which it is not supposed to do [11], and activities relating to the disposal of the software and any sensitive information that it might contain. Whereas verification and validation phases are common components in the development lifecycles followed by the DoD and other government agencies, they are often not fully developed in the private sector.

For the sake of comparison, one can consider highlights of the activities for the various processes and phases of the DoD's Integrated Defense Acquisition, Technology and Logistics Lifecycle Management System as presented by the Defense Acquisition University [12]. For the most part, many of the phases of the DoD model are similar to those shown in figure 2 and some of the processes are the same, although the scope of the DoD activities includes other important procedural areas, such as planning, contracting and financial management. The DoD model provides a more complete framework for complex processes, procedures and reporting.

The risks relating to the software supply chain largely depend on the nature and origin of the software. Table 4 shows the levels of risk that might be expected with respect to software from different sources and whether or not technical support is available.

Conclusions and Recommendations

The initial challenges addressed in this article are to identify and understand software-specific supply chain threats and vulnerabilities and protect against them. Risk was considered at each stage of the software development and software supply chain life cycles and activities suggested to mitigate the risk factors. It is also suggested that the only way to fully understand complex supply chains is to develop computer simulation models that represent those supply chains at the transaction level—i.e., from the process and product-flow perspectives.

Much has already been accomplished in various industries to gain a better understanding of software supply chains and their inherent risk. However, much remains to be done in the public and private sectors in order to achieve an acceptable level of understanding of related risks and their mitigation.

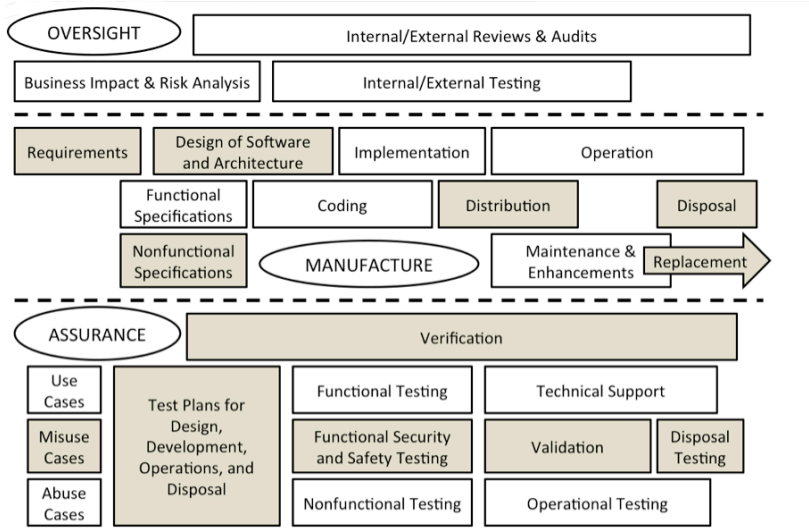

Figure 2: Oversight, Manufacture and Assurance in the SDLC
Source: C. Warren Axelrod; Engineering Safe and Secure Software Systems, Artech House, 2013. Reprinted with permission.

Risks	Sources			
	COTS/GOTS	Custom	Open Source	Unsupported
Risk that malware has been introduced during the development phase	Moderate	Low	Moderate	High
Risk that malware will be introduced during operation	Moderate to high	Low to moderate	Low	High
Risk that improper disposal will lead to compromise	Moderate to high	Moderate to high	Low	Low to moderate

Table 4: Supply Chain Risk by Origin of Software

ABOUT THE AUTHOR

C. Warren Axelrod, Ph.D., is a senior consultant with Delta Risk, a consultancy specializing in cyber defense, resiliency, and risk management. Previously, Axelrod was the chief privacy officer and business information security officer for US Trust.

He was a co-founder of the FS-ISAC (Financial Services Information Sharing and Analysis Center). He represented the financial services sector at the national command center over the Y2K weekend and testified before Congress about cyber security in 2001. He has participated in a number of initiatives at sector and national levels.

Dr. Axelrod was honored with the prestigious ISE (Information Security Executive) Luminary Leadership Award in 2007 and, in 2003, he received the Computerworld Premier 100 IT Leaders Award and Best in Class Award. His article "Accounting for Value and Uncertainty in Security Metrics" won ISACA's Michael P. Cangemi Best Book/Best Article Award in 2009.

Dr. Axelrod has published five books on various IT risk, outsourcing, cyber security, privacy and safety topics. His most recent book is Engineering Safe and Secure Software Systems, released in 2012 by Artech House. He has published three prior articles in CrossTalk magazine.

He holds a Ph.D. (managerial economics) from Cornell University and MA (economics and statistics) and B.Sc. (electrical engineering) honors degrees from the University of Glasgow. He is certified as a CISSP and CISM.

Phone 917-670-1720
E-mail: waxelrod@delta-risk.net

REFERENCES

1. Zsidisin, George A.; "A Grounded Definition of Supply Risk," Journal of Purchasing and Supply Management, vol. 9, no. 5-6, September/November 2003, p. 217-224
2. Goertzel, Karen Mercedes; "Protecting Software Intellectual Property Against Counterfeiting and Piracy," STSC CrossTalk: The Journal of Defense Software Engineering, vol. 24, no. 5, September/October 2011, p. 6-9
3. Ellison, Robert J., et al; Evaluating and Mitigating Software Supply Chain Security Risks, Technical Note CMU/SEI-2010-TN-016, Software Engineering Institute, May 2010, <www.sei.cmu.edu/library/abstracts/reports/10tn016.cfm>
4. Oehmen, Josef; Mohamed Ben-Daya; Omera Khan; "Integrating Supply VChain Risks in Product Development: A Conceptual Framework," in Samir Dani (ed.) Proceedings of the Tenth International Research Seminar on Supply Chain Risk Management, ISCRiM, September 2010, p. 56-61, <www.husdal.com/wp-content/uploads/2010/09/Proceedings-ISCRiM-2010.pdf>
5. Goertzel, Karen Mercedes; "Supply Chain Risk Management and the Software Supply Chain," OWASP AppSec DC, 2010, <https://www.owasp.org/images/7/77/BoozAllen-AppSecDC2010-sw_scrm.pdf>
6. Goertzel, Karen Mercedes, et al; State of the Art Report on Supply Chain Risk Management for the Off-the-Shelf (OTS) Information and Communications Technology (ICT) Supply Chain, Department of Defense Information Assurance Technology Analysis Center (IATAC), USA, 2010.
7. Ellison, Robert J., et al; Software Supply Chain Risk Management: From Products to Systems of Systems, Technical Note: CMU/SEI-2010-TN-026, Software Engineering Institute, December 2010, <www.sei.cmu.edu/library/abstracts/reports/10tn026.cfm>
8. Axelrod, C. Warren; "Mitigating the Risks of Cyber-Physical Systems," Proceedings of the 2013 IEEE LISAT (Long Island Systems, Applications and Technology) Conference, Farmingdale, NY, May 2013.
9. Axelrod, C. Warren; "Risks of Unrecognized Commonalities in the Information Technology Supply Chain," Proceedings of the 2010 IEEE International Conference – Technologies for Homeland Security, Waltham, MA, November 2010.
10. Raffo, David M.; Siri-on Setamanit; "A Simulation Model for Global Software Development Project," The International Workshop on Software Process Simulation and Modeling, USA, 2005, <www.sba.pdx.edu/faculty/davidr/draccess/WEB/publications/JOURNAL/ProSim'05-GSD.pdf>
11. Axelrod, C. Warren; "The Need for Functional Security Testing," STSC CrossTalk: The Journal of Defense Software Engineering, March/April 2011, p. 17-21.
12. Defense Acquisition University, Highlights of the Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management System, Version 5.4, 2010, <www.wired.com/images_blogs/dangerroom/2010/09/atl_wall_chart.jpg>

NOTES

1. In 2009, the U.S. Banking and Finance Sector, under the auspices of the FSSCC (Financial Services Sector Coordinating Council), initiated a project to determine IT (information technology) products used by the industry and evaluate relevant threats and risks. The report from Phase 1 of the effort, dubbed "Protecting the Resiliency of the Supply Chain," comprised the results of surveys about leading security practices as they related to purchased software, internally-developed software, and custom software developed by third parties, as well as computer and network hardware, firmware and appliances. Phase 2 was an attempt to identify the full range of IT resources used by banks and securities firms. Unfortunately, it proved difficult to gather even rudimentary information, much less specific data that would allow for a full analysis.
2. One of the characteristics of the infamous Stuxnet worm, which caused centrifuges in Iranian nuclear materials processing plants to self-destruct, was that it reported to operators that all was well even while bad things were happening.
3. One example of an embedded product about which the acquirer might not have been aware is SQL Server. When the SQL Slammer worm hit in January 2003, it surprised IT management by affecting applications that had silently loaded SQL Server. For a description of the worm and a list of affected applications, see F-Secure Corp., <www.f-secure.com/v-descs/mssqlm.shtml>