Mitigating Software Supply Chain Risk

Software supply chains differ from those of manufactured products in many ways. Consequently, a number of the risk factors of software supply chains are unique. This article defines supply risk factors as they pertain to various forms of software, examines their impact and offers suggestions for identification and mitigation.

Natural catastrophes, such as earthquakes, tsunamis, hurricanes, volcanic eruptions and floods, and man-made disasters, such as wars and terrorist attacks, or just plain human error, can have major impacts on supply chains. The level of damage and interruption depends on the location and nature of the event. While disruptions to manufacturing from catastrophic events can be highly visible, events that affect the software supply chain are usually less observable, but can be just as devastating. Software products are frequently affected by specific types of events, such as counterfeiting, piracy and the introduction of malware, that are less visible and often less dramatic, but can inflict just as much economic damage as physical disasters. Software supply chains are also subjected to a host of risk factors that are particular to the characteristics of specific unique software.

Looking at various types of software, software development and acquisition life cycles, and risk and exposure to and from software, vulnerabilities are shown to facilitate the disruption of the supply chain and provide those with bad intentions opportunities to steal and manipulate software and otherwise diminish its value.

To ensure that incidents are quickly recognized and their impact minimized, organizations must have a good understanding of the intricacies of particular software supply chains and the potential effects of disruptions. Organizations also need the ability to monitor supply chains, identify attacks and failures, and respond quickly to any aberrant behavior.

Furthermore, transaction-level simulation models of software supply chains can be effective in identifying weaknesses and formulating mitigating strategies. Such models can also be used for closed-loop exercises. These help participants understand the intricacies and interactions of components of software supply chains and how failure of any component affects the overall system.

SOFTWARE SUPPLY RISK

Most of the literature on supply chain risk management addresses the manufacture and distribution of physical products and further focuses on disruption of the supply chain and on counterfeit products, particularly in the pharmaceutical and luxury goods industries.

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

This definition covers the chance that an organization is able to satisfy customer demand adequately and can avoid substitution for the real products, such as counterfeit drugs or other products, that could cause physical harm to customers and financial loss for businesses. While these factors also affect software products, there are a number of additional factors specific to software that are practically invisible and much less tangible. For one, software-related threats encompass unauthorized and malicious access to intellectual property and sensitive data. Furthermore, there are piracy issues that arise from phony software. Software differs from physical products in a number of important ways and the various types of software affect supply chain risk in different ways.

It is recognized that supply chain risk management usually refers to limiting the risk of supply disruptions, typically disruptions that delay deliveries of an item to a manufacturer or
consumer. However, the term supply chain risk management can also refer to ensuring that inferior or counterfeit components are not introduced by suppliers.

Of course, inferior or counterfeit components, malware, or other items that might negatively affect the software system can be introduced by external parties and/or insiders without the knowledge of suppliers.

While there are many discussions about supply chain risk management as it pertains to software and software-intensive systems, there is no comprehensive definition of software supply risk. Software supply risk is a combination of factors that relate to adverse events affecting software throughout its supply chain life cycle and the probability of those events occurring. Such events and factors include those that:

- Prevent the purchasing firm from meeting customer demand
- Prevent the supplier from providing contracted technical and operational support to purchasing firms and end users
- Compromise the supply chain management processes causing disruption and delays in meeting deadlines, diversion of products, theft of products, unauthorized copying and distribution of software products, and the introduction of undesirable software (i.e., malware, viruses) and physical items (e.g., fake circuit boards)
- Pervert the behavior of software products through tampering, counterfeiting and the introduction of malware

This article addresses concerns that result from these adverse events and provides recommendations as to how related risk can be mitigated and/or avoided, even if it cannot be fully prevented.

**Types of Software**

The following categories of software are of interest:

- **Software products**:
  - Commercial/government off-the-shelf (COTS/GOTS) software
  - Open-source software
  - Custom-built software (developed internally, externally or both)
  - Hybrid, a combination of custom, open-source and off-the-shelf (OTS) software
- **Embedded software**—Software or firmware built into physical products
- **Supply chain management software**—Software products that monitor supply chain processes and flows and report deviations from expected behavior

Software products are either open or closed with respect to access to source code by customers. For the most part, source code of OTS software is not made available to customers.

End users of both open-source and custom-built software usually have access to the source code, which allows for code reviews and static testing. Hybrid software should generally be considered to be as weak as its weakest components, which are often shrink-wrapped products from vendors of varied levels of trustworthiness.

For the purposes of this article, risk factors relating to supply chains for embedded software are considered mostly the same as risk factors affecting physical products, to the extent that embedded software is considered part of an overall product. However, even though embedded software may not be top of mind for manufacturers and distributors of physical products, particular risk factors, to which software is generally subject, must be considered for embedded software.

Software that is used to manage supply chains also needs to be considered, since effective supply chain management can mitigate many of the risk factors normally encountered. Supply chain management software is available off the shelf or may be custom built. Either way, such supply chain management software can be a vector for attackers. Few researchers appear to even consider the risk of compromise of supply chain management software. However, such software may be compromised, even to the extent that it may be made to report that everything is in order when it is not, and, therefore, may represent a significant risk.

The effectiveness of managing supply chains depends on how comprehensive, accurate and timely such supply chain command-and-control systems are. Frequently, organizations are surprised to discover the interrelationships and common points of failure that exist within their supply chains—discovering them only when there are disrupting events, in which case it often takes days and weeks of investigation to get a full understanding of where interdependencies exist and how they might be reduced. It is highly desirable, therefore, to build a custom system or implement a commercial supply chain management system that is complete and up to date in its understanding of the complexities and relationships among supply chain components and that provides close to real-time, accurate information about disruptions and their impact.
SOFTWARE DEVELOPMENT AND SUPPLY CHAIN LIFE CYCLE

It is important to differentiate between the software (or system) development life cycle (SDLC) and the software supply chain life cycle. The former applies to the phases inherent in the manufacturing and deployment of all software, and the latter relates to processes that involve suppliers and customers of software products. There may be considerable overlap between the two life cycles depending upon the degree to which the organization acquires software from third parties vs. builds in-house. For example, there is no supply chain for software that is built totally in-house, using in-house engineers. On the other hand, for COTS software, the early stages of the SDLC, along with maintenance and technical support, are considered to be within the supply chain, while the operations phase is not part of the supply chain if the software system is operated in-house.

SPECIFIC CHARACTERISTICS OF SOFTWARE PRODUCTS

Software products or software components of broader systems or products differ from manufactured products in several important ways, as follows:

- Software can be stolen without having to remove or otherwise change the original copy.
- Illegal or unauthorized copies of software may be sold independently on the black market.
- Physical transportation of software is not required since copies can be downloaded.
- Valid or lawful versions of software can be modified and still appear to be valid.

Figure 1 compares the characteristics of software products to those of physical products throughout the supply chain life cycle.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Physical Product</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements and design</td>
<td>Determination of requirements and the design of products can be done globally based on the availability of educated, productive human resources and reliable, low-cost telecommunications.</td>
<td>Determination of requirements and the design of software can be done globally based on the availability of educated, productive human resources and reliable, low-cost telecommunications.</td>
</tr>
<tr>
<td>Manufacture (development)</td>
<td>Factories may be located anywhere based on proximity to raw materials, low-cost energy, cheap labor and efficient transportation, for example.</td>
<td>Development and testing of software can be done globally based on the availability of knowledgeable staff and reliable, low-cost telecommunications.</td>
</tr>
<tr>
<td>Distribution</td>
<td>Physical products must be transported via land, sea and/or air depending on size, weight, toxicity and flammability, for example.</td>
<td>Some physical distribution of optical or magnetic media exists, but software may be downloaded electronically via telecommunications lines for much lower cost than for physical distribution.</td>
</tr>
<tr>
<td>Warehousing</td>
<td>Storage is required for raw materials and intermediate and finished products, often at substantial cost.</td>
<td>Electronic copies can be stored on servers. Storage of media containing software has characteristics similar to those of physical products.</td>
</tr>
<tr>
<td>Deployment</td>
<td>Products are sold through distributors, wholesale and retail outlets, and the like.</td>
<td>Software is deployed via distributors, wholesale and retail outlets, and directly via downloads from vendors’ and/or distributors’ web sites.</td>
</tr>
<tr>
<td>Operation</td>
<td>Products are consumed and/or used in a variety of ways. As physical products wear out, they have to be repaired or replaced. At some point, the cost of operation and maintenance may become prohibitive in comparison to those of newer technologies.</td>
<td>Software may be run indefinitely on designated platforms and machines. However, upgrades may be needed as requirements, platforms, hardware and environments change.</td>
</tr>
<tr>
<td>Maintenance and support</td>
<td>Onsite third-party maintenance is usually required, or product may be shipped to a repair facility. Technical and user support can be onsite or remote.</td>
<td>Maintenance of software may include new versions or patches for installed software. Technical and user support is likely to be from remote locations.</td>
</tr>
<tr>
<td>Disposal</td>
<td>Products may require special handling if they contain valuable or dangerous materials.</td>
<td>Usually software can be deleted and/or upgraded. Media may need to be physically destroyed.</td>
</tr>
</tbody>
</table>
Although supply chain risk usually dominates the manufacturing and distribution phases, more emphasis should be placed on the early phases of the product development process. This is particularly relevant to software supply chains, in which development efforts are often greater than those for distribution, because the electronic distribution of software is relatively simple and inexpensive compared to distributing and selling physical products. That said, some software is still sold on physical media.

The following software supply chain elements and constituents (in parentheses) are at risk:

- **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)

For example, with respect to confidentiality, one must consider the potential risk of someone stealing intellectual property or trade secrets, for example, as well as the consequences of compromise of customer and employee personal data, particularly that which is considered by lawmakers and regulators not to be in the public domain, and of business data, such as planning strategies, intent to merge with or acquire other entities, and so on.

When it comes to integrity, one can imagine the supply chain processes, as well as the modification of software products and related data, being exploited by criminals. Similarly, one needs to consider the range of impact for the other risk elements listed here.

Not only must one aim to mitigate supply chain risk, one must also be able to demonstrate that the mitigation efforts have been effective. The following properties enable one to assure that the risk has been adequately mitigated or avoided:

- **Transparency**
- **Quality**
- **Accountability**

A report by the US Department of Defense Information Assurance Technology Analysis Center (IATAC) suggests that constituents are subjected to various supply chain threats. Figure 2 assigns threats to constituents.

<table>
<thead>
<tr>
<th>Threats</th>
<th>Supply Chain Constituents</th>
<th>Products</th>
<th>Supply Chain Processes</th>
<th>Product Flows</th>
<th>Supply Chain Data Flows</th>
<th>Management Data</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabotage</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tampering</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Counterfeiting, piracy</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Theft</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Destruction/deletion</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disruption/delay</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exfiltration—theft</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exfiltration—disruption</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infiltration, subversion</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diversion</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export control violations</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undesirable physical items</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corruption</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social engineering</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insider threat</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pseudo-insider threat</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign ownership, influence</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The potential effects of the supply chain threats of Figure 2 become much more complicated as systems are combined with other systems and products in various ways, as described in a Software Engineering Institute (SEI) report. The SEI report points to the similarities and differences between product suppliers and system development contractors and notes that acquirers’ assessments of products take place after the product development is completed. However, for custom-built systems, acquirers are able to “actively monitor both contractor and product supply chain risk during the development process.” Also, the report suggests that software supply chain risk analysis includes 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 risk by acquirer

The report suggests, however, that more work is needed to provide acquirers with useful tools.

Figure 3 illustrates four ways in which software systems may be combined and the characteristics of the combined systems.

<table>
<thead>
<tr>
<th>Forms of Combination</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded software</td>
<td>Many software products and systems contain software within the product or system about which acquirers might not be aware.</td>
</tr>
<tr>
<td>Integrated systems</td>
<td>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.</td>
</tr>
<tr>
<td>Systems of systems</td>
<td>Disparate systems are combined to form systems of systems, which produce functionality that is greater than the sum of the individual systems.</td>
</tr>
<tr>
<td>Cyberphysical systems</td>
<td>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.</td>
</tr>
</tbody>
</table>

Such combinations of systems likely contain subsystems from all of the categories (OTS, custom, open-source and hybrid). To evaluate supply chain risk factors of such combined systems, it is necessary to examine the origins and functionality of each component as well as the interactions among them.

Activities for the management of risk vary with the various phases of the supply chain or acquisition life cycle. The SEI report enumerates those activities as they relate specifically to security risk. Figure 4 assigns such activities to the phases of Figure 1. A number of activities have been added to the original list from the SEI report.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 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. |
LIFE CYCLE RISK

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.

For custom-built software, 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.

Risk is markedly different with respect to the degree of outsourcing and geographic dispersion. Further, the level of risk varies greatly with factors such as loyalties and motivations of employees and contractors and legal, social and economic differences across countries and ethnic groups.15

Figure 5 illustrates factors affecting supply chain risk throughout the development life cycle. As can be seen from the diagram, there are general types of events, such as natural and man-made disasters, that can affect all supply chains including the software supply chain. However, there are also a number of compromises, such as the insertion of malware, that are prevalent with or unique to software. Others, such as the theft of intellectual property and other sensitive data, are common across many products, but are facilitated by the ability to copy software and data without changing the original or having to be onsite.

As discussed previously, the SDLC and supply chain life cycle do not necessarily coincide. Figure 5 does not assign particular SDLC phases to supply chains since that assignment varies with the distribution of activities among onshore, offshore and in-house facilities and resources.

---

**Figure 5**—Supply Chain Risk by Phases of the SDLC

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design</th>
<th>Development</th>
<th>Testing</th>
<th>Deployment</th>
<th>Operation</th>
<th>Maintenance/Support</th>
<th>Disposal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Supply-chain-related Risk Factors**

- Insertion of Back Doors, Malware, etc.
- Theft of Intellectual Property and Sensitive Data

**DELAYS OR FAILURES DUE TO LOCAL AND REGIONAL DISRUPTIONS, DISASTERS OR CATASTROPHES**

- Global Software Development (GSD) Problems: Communications, Coordination and Control, Cultural Differences, Language Differences, and Time Zone Differences
- Support Problems (same as for GSD problems)

Source: Based on issues enumerated in D. Rafio and S. Setamanit, “A Simulation Model for Global Software Development Project,” The International Workshop on Software Process Simulation and Modeling, USA, 2005
THE NEED FOR MODELS
There have been several research efforts related to resolving issues around global software development (GSD) and risk of software supply chains. Simulation can be used to optimize various characteristics of dispersed software development efforts due to the complexity and flexibility of such arrangements.16

Researchers have developed models for the impact of cultural, communications and other factors on the distributed development of software—whether the software is custom-built, developed in-house, outsourced, or a combination of OTS and open-source software. The US 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 significant.

THE SUPPLY CHAIN FOR SOFTWARE DEVELOPMENT AND DISTRIBUTION
To make decisions with respect to any particular supply chain it is necessary to understand each phase and the interaction among phases. Figure 6 shows the life cycles for three major aspects of software development, testing and deployment: manufacturing, assurance and oversight. Each aspect may proceed along its own path, but, ultimately, it is the combination of all three that makes for a resilient, high-integrity software supply chain.

Manufacturing is the “nuts and bolts” of developing and distributing software. Oversight encompasses the independent oversight of the processes and products of the manufacturing life cycle. Assurance includes separate evaluations of the quality, integrity and trustworthiness of the software being manufactured.

When one refers to manufacturing, one means the full life cycle from conception (requirements) through design, development, distribution, operation and disposal (as well as replacement planning). Oversight can be performed by internal or external auditors, the project management office (PMO) and/or regulators, as long as the overseers are independent of the reporting structure responsible for the manufacturing process. Overseers review the status of the project, the existence of deliverables (and the like), and often perform their own testing of attributes such as security and integrity. Assurance is more closely integrated with the manufacturing process, but should not be under the direct control of manufacturing project management so that assurance staff can be objective and report with impunity on issues they discover for which correction may adversely affect the cost and timeliness of the end product.

The shaded boxes in figure 6 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,17 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 life cycles followed by the US Department of Defense and other government agencies, they are often not included in software development for the private sector. In the commercial world, it is common to release software that has not been adequately tested, often relying on users to identify errors or bugs and report them to the vendor.

As figure 6 suggests, there are actually three streams of activity in the typical manufacturing of software products; each stream representing a series of activities that need to be invoked to arrive at a high-integrity, secure and safe mission-critical software system. One set of issues arises simply from whether or not certain activities (such as functional security testing) have been included in the process. A second level of issues comes from whether the appropriate level of attention is paid to those activities that are included. The third series of issues relates to the supply chain, namely, where and by whom each of the activities is performed.18

For the sake of comparison, one can consider highlights of the activities for the various processes and phases of the US Department of Defense (DoD)’s Integrated Defense Acquisition, Technology and Logistics Life Cycle Management System as presented by the Defense Acquisition University.19 For the most part, many of the phases of the DoD model are similar to those shown in figure 6 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.
RISK AND THE SOFTWARE SUPPLY CHAIN

Most of the attention and concern about the supply chain relates to the manufacturing of software systems by third parties—either custom or OTS. However, it is important to include internally developed software when discussing the software supply chain, since, even when the manufacturing takes place in-house, there is usually significant reliance on outside services, and of course, virtually all custom-built application software relies on third-party software products, such as operating systems.

The risk relating to the software supply chain largely depends on the nature and origin of the software. Figure 7 shows the levels of risk that may be expected with respect to software from different sources and whether or not technical support is available.

CONCLUSION AND RECOMMENDATIONS

Examination of risk related to software supply chains demonstrates that in addition to factors, such as disruption due to natural disasters, that are common to all supply chains, software supply chains are subjected to risk that is specific to software, such as the insertion of malware.

The initial challenges that have been addressed in this article are to identify and understand software-specific supply chain threats and vulnerabilities and protect against them. Since software supply chains are so complex, risk was considered at each stage of the software development and software supply chain life cycles and activities were suggested to mitigate some of 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. The models should be developed using subject matter experts from the software supply chain field and refined and honed to realistic representations by means of closed-loop exercises—where experiences and lessons learned are ploughed back into model development in order to arrive at more realistic models.

A great deal has already been accomplished in various industries, such as financial services, to gain 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 risk and how to mitigate it, and then use that understanding to implement effective methods and approaches to improve the security and integrity of these supply chains.

ENDNOTES


2 In this case, risk is defined as the probability of an adverse event. The usual definition of risk includes both the probability of the event and some measure of the consequences or impact of the event if it were to occur. By multiplying the probability of an event by its impact, one arrives at the expected value of the event (usually an expected loss), which is generally accepted as a definition of risk. Further discussions of various approaches to risk can be obtained from many sources, such as: Axelrod, C. Warren; Jennifer L. Bayuk; Daniel Schutzer (editors), *Enterprise Information Security and Privacy*, Artech House, USA, 2009.


5 Ibid.

6 An interesting exception to this rule is the offer made by the huge Chinese network product provider Huawei to provide source code of its proprietary software to the Australian government so that the latter will lift a ban on its products. See Whittaker, Zack; “Huawei Offers Australia ‘Unrestricted’ Access to Hardware, Source Code,” CNET, 24 October 2012, http://news.cnet.com/8301-13578_3-57538843-38/huawei-offers-australia-unrestricted-access-to-hardware-source-code/

7 One of the characteristics of the famous Stuxnet worm, which caused centrifuges in Iranian nuclear material processing plants to self-destruct, was that it reported to operators that all was well even while bad things were happening.


An extensive treatment of supply chain risk as it applies to OTS software is given in: 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.


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.

In 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, the authors provide a good overview of the various factors that affect global software development (GSD) projects and recommend the use of simulation models to integrate and synthesize relevant factors.


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.

In 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, the authors provide a good overview of the various factors that affect global software development (GSD) projects and recommend the use of simulation models to integrate and synthesize relevant factors.


This article assumes that all activities have been included and concentrates on the supply chain aspects of location, expertise, management control and so on, that affect distributed manufacturing, oversight and assurance processes.