CSIS REPORT

FOREIGN INFLUENCE ON SOFTWARE

Risks and Recourse

A Report of the
Technology and Public Policy Program
Center for Strategic and International Studies

Author
James A. Lewis

Foreword
Phillip J. Bond

March 2007

CSIS

FOREIGN INFLUENCE ON SOFTWARE

Risks and Recourse

A Report of the
Technology and Public Policy Program
Center for Strategic and International Studies

Author

James A. Lewis

Foreword

Phillip J. Bond

March 2007

About CSIS

The Center for Strategic and International Studies (CSIS) seeks to advance global security and prosperity in an era of economic and political transformation by providing strategic insights and practical policy solutions to decisionmakers. CSIS serves as a strategic planning partner for the government by conducting research and analysis and developing policy initiatives that look into the future and anticipate change. Our more than 25 programs are organized around three themes:

- *Defense and Security Policy*—With one of the most comprehensive programs on U.S. defense policy and international security, CSIS proposes reforms to U.S. defense organization, defense policy, and the defense industrial and technology base. Other CSIS programs offer solutions to the challenges of proliferation, transnational terrorism, homeland security, and post-conflict reconstruction.
- *Global Challenges*—With programs on demographics and population, energy security, global health, technology, and the international financial and economic system, CSIS addresses the new drivers of risk and opportunity on the world stage.
- *Regional Transformation*—CSIS is the only institution of its kind with resident experts studying the transformation of all of the world's major geographic regions. CSIS specialists seek to anticipate changes in key countries and regions—from Africa to Asia, from Europe to Latin America, and from the Middle East to North America.

Founded in 1962 by David M. Abshire and Admiral Arleigh Burke, CSIS is a bipartisan, nonprofit organization headquartered in Washington, D.C., with more than 220 full-time staff and a large network of affiliated experts. Former U.S. senator Sam Nunn became chairman of the CSIS Board of Trustees in 1999, and John J. Hamre has led CSIS as its president and chief executive officer since 2000.

CSIS does not take specific policy positions; accordingly, all views expressed herein should be understood to be solely those of the author(s).

Cover photograph: Circuit board © Royalty-Free/CORBIS

Library of Congress Cataloging-in-Publication Data
CIP information available upon request
ISBN-13: 978-0-89206-497-7

The CSIS Press
Center for Strategic and International Studies
1800 K Street, N.W., Washington, D.C. 20006
Telephone: (202) 887-0200
Fax: (202) 775-3199
E-mail: books@csis.org
Web: www.csis.org/

Contents

Foreword

Recent months have served to underscore the fact that cyberspace threats remain: direct cyber attacks on Internet root servers, government Web sites, and countless commercial IT infrastructures. Suspected attackers include both intelligence agencies of organized states as well as a variety of non-state bad actors that harbor no shortage of ill will for the United States.

In light of this harsh reality, national policymakers are voicing concerns over foreign involvement in the development of software and other technologies employed to uphold our national security. Concerns over security in the development of these critical technologies are certainly justified. We need a comprehensive effort to ensure that IT solutions meant to uphold national security are not turned into weapons against us.

Done well, such an undertaking will leave us stronger and sharpen the technological edge we hold over our adversaries. Done poorly, good intentions will give way to erosion of our national security and economic competitiveness.

A security scheme that would impose restrictions on software developed anywhere outside the United States, or more narrowly by blacklisting certain nation states, would harm national security in two ways. First and foremost, it would give Americans and their leaders false comfort. Second, such a scheme would deny our defense and intelligence agencies access to critical, innovative technologies.

The information technology industry is global. Corporations based at home and abroad depend on a worldwide supply chain in order to develop and deliver the very best products to the U.S. government. Our government, and our people, expect and deserve nothing less than the latest in IT to serve and protect the United States. An industry restricted to Americans working in America would be unable to keep up. Our competitors in the global economy and our adversaries jockeying for position on the battlefields of cyberspace would have the advantage.

As a sponsor of the Center for Strategic and International Studies (CSIS), we were pleased to see that CSIS has developed a body of recommendations that follow the better course. The key ingredients to success are here: a full assessment of related risks, shared in a responsible way with industry; a focus on assurance, rather than location; leveraging and accelerating existing certification and assurance efforts; and support for, rather than restrictions on, our nation's ability to innovate. These are necessary elements of any effective assurance effort.

Right or wrong, critics of the war in Iraq have made much of the perception that our troops were sent into battle without the right equipment. Don't let them make the same critiques about our readiness in cyberspace. Let us address these concerns in a way that solves the problem and preserves our technological advantage, rather than one that does more harm than good.

—Phillip J. Bond, President and CEO of the Information Technology Association of America

Acknowledgments

A number of people contributed to this report. John Hillery and Erik Hembre researched industry trends. Ashley Rasmussen did heroic work in editing drafts. Experts from both the private sector and the government, including IBM; SAP; Jeff Lande, Trey Hodgkins, and others from ITAA; Microsoft; Guy Copeland and others from CSC; Tiffany Jones and others from Symantec; Bill Sweeney from EDS; and Shannon Kellogg and Kem Clawson from EMC, made valuable suggestions for improvement. I thank them for their help.

Executive Summary

Globalization drives change. The immense economic transition that comes with globalization has brought an unprecedented prosperity to the world. The United States is among the chief beneficiaries. However, America and other countries have learned that with the benefits come new risks. Nations face different and unexpected threats to their safety. Opponents will look to the immense global economic machine created for commerce to find new ways to attack. Creating policies that can maintain economic opportunity while managing new risks is one of the most complex challenges that governments face today. This report looks at one new set of risks created by changes in how companies write software and considers how best to mitigate that risk.

Companies are more efficient and more competitive when they take advantage of global supply chains. However, depending on a foreign and perhaps anonymous supply chain for sensitive technologies creates reasonable concern. The problem is not that an opponent will blockade the United States or cut off crucial sources of supply. Instead, a sophisticated opponent could look to exploit the global supply chain for malicious purposes. The broad adoption of commercial off-the-shelf (COTS) software for sensitive applications complicates matters, because companies make this software for a mass market, not for a specific or sensitive application.

Although companies in the United States lead the information technology industry, they use complex networks of suppliers, subcontractors, and integrators located around the world to make their products. The industry is highly competitive, and that competition drives companies to pursue talent and lower costs wherever they are found. The growth of a skilled workforce outside the United States works to move design and production functions overseas. Foreign programmers are cheaper, but more importantly, business strategies must be global to succeed. Key trends in markets and business practices encourage companies to move software production outside the United States, but in recent years, the global nature of this industry has created concern over the security of networks and information.

Software is of particular interest from a security perspective. Some intelligence analysts believe that software offers one of the best mechanisms for technical intelligence collection by a range of adversaries. A foreign intelligence service or a hostile group could infiltrate the global supply chain and surreptitiously introduce “malicious code” that would make it easier to gain remote access to networks and information or remotely trigger disruptive events during a crisis after the product is sold and installed.

Many programs involve millions of lines of code, and malicious code hidden in a large program is difficult to detect. Both U.S. and foreign-developed software are vulnerable, although the opportunity may be somewhat greater in foreign processes, where an opponent may face fewer obstacles and less scrutiny. Manual scanning for malicious code is prohibitively expensive and not effective—a

programmer may not detect a problem even if he or she is looking directly at it. Existing software development practices focus on reducing defects and improving performance. Automated quality-assurance programs created to detect coding errors or vulnerabilities can miss unintentional errors. Open architectures, COTS software, and the use of legacy code keep costs down, but they may also create opportunities for mischief.

Concern over the insertion of malicious code into foreign-made software appeared at least as early as 1999, in a Defense Science Board report on globalization and security. A stream of reports since then—from the Government Accountability Office, the Department of Defense, and the Intelligence Community—have reiterated that potential opponents, whom we know are looking at the possibility of asymmetric attack and information warfare, may use the opportunities created by a growing dependence on global supply to gain advantage over the United States.

This is a kind of cyber sabotage. It does not exploit existing vulnerabilities. Instead, it creates its own vulnerability. It is one weapon in the arsenal of those who wish to attack in cyberspace. The crux of the matter is whether software (along with other IT products) made outside the United States is less trustworthy than software made domestically by Americans. This question reflects a larger set of concerns, including fears about the effect of globalization on U.S. technological leadership and competitiveness; the seemingly intractable problem of information security; the very real threat of espionage; and the need to secure critical infrastructure that depends on IT for its operations.

However, in the arsenal of information warfare, the surreptitious insertion of malicious code during production may not be the most powerful weapon available. An attacker has many different options for gaining access to networks and information. Exploiting a global supply chain is one of them, but it is not the best option as compared to the alternatives: hacking, insider attacks, and traditional methods of espionage and crime. As in business, low cost, low risk, and high return are the preferred criteria, and these make other modes of attack more attractive. Given this, a malicious code insertion does not compare well with other attacks. Finding and exploiting existing vulnerabilities in networks remains the most attractive mode of attack. Recruiting insiders to gain access is a time-honored and successful strategy.

Despite this, it is still safe to assume that a well-resourced opponent will consider all three modes of attack—hacking, insiders, and malicious modifications during coding. Malicious code insertions, where an agent or criminal surreptitiously inserts lines of code into a program, are possible, although they are not the most probable mode of attack on information systems and networks. Although there may be specific instances where the United States has knowledge that certain foreign companies require extra security, focusing on location—the place where software is written—is less useful for improving security than are the processes by which software is produced and tested. An approach to software assurance based on locational restrictions for software acquisition will increase costs for government and potentially reduce the pace of

innovation. It could affect the ability of agencies to obtain cutting-edge software and take advantage of new Internet technologies. Locational requirements could make U.S. firms less competitive by cutting them off from markets, the global talent pool, and the ability to pursue lower costs in design and production. As technological innovation spreads around the world, some of the best technologies will be produced outside the United States, and it would be harmful to cut ourselves off from these products based simply on geographic considerations.

These issues are difficult to address, and there are no simple solutions, but there are several steps that can improve the situation. These steps can build on the work done by the Department of Homeland Security, the Department of Defense, and other agencies to improve software assurance and network security. Government can play an essential coordinating and guiding function to strengthen initiatives and processes for increased assurance and to take advantage of commercial best practices. These practices include: security training for programmers; designing software from the start with security considerations in mind; strong management procedures for oversight and transparency, including management of modifications or additions to code; an independent review process for security (including the use of automated software review tools); "red teaming"[1] to test for vulnerabilities; and penetration efforts by specialist contractors. Finding new ways to identify and, where appropriate, require best practices will reduce the risk of foreign involvement in software production.

The United States should place any effort to reduce risk from foreign involvement in software in the larger context of cyber security and software assurance. Improvements in cyber security and software assurance will mitigate the risk of malicious code insertions, whether by foreign or U.S. programmers. The long-term, strategic objective is to take those steps that will improve security and maintain U.S strength in information technology. In light of this, we recommend eight measures:

1. **Assess the risk (and share the assessment).** Inserting malicious code into software during the production process (whether overseas or in the United States) is only one of several attack options available to opponents. Responsibility for collecting information about opponents who are considering such attacks and the form these attacks might take should be assigned to the Intelligence Community and the information shared among agencies and with appropriately cleared company representatives. Government and industry can develop formal processes to improve the exchange of information about threats and vulnerabilities to inform and coordinate their risk assessments.

2. **Focus on assurance, not location.** In the past, it was safe to assume that technology produced in the United States by a U.S. firm did not contain intentional vulnerabilities. This assumption no longer holds. Even if the technology is manufactured in the United States, the global nature of business means that this alone does not guarantee trustworthiness. An American

[1] A group formed to simulate an opponent and attack a product or system.

company is likely to have employees from a broad range of countries. Foreign intelligence agencies could take advantage of the increasing internationalization of business to insert or recruit insiders, including U.S. citizens, with access to software production in the United States.

The place where companies make software is not the key variable. Since 2000, many companies have made security a central element of their design and production processes for software. A strategy that takes advantage of the best procedures adopted by leading software manufacturers to make their products more secure has a better chance of succeeding than a strategy that attempts to determine security by looking at location.

3. **Avoid one-size-fits-all solutions.** The government already has processes for producing software with high assurance levels for very sensitive applications, such as command-and-control or intelligence. Cleared personnel working in secure facilities and following strict guidelines write this software. This provides software that is more trustworthy, but it is too expensive and too limiting to scale across government. Building on existing efforts, an effective strategy will map software assurance levels and requirements to the sensitivity of the function and networks they support. Federal requirements could scale progressively from routine applications to the most sensitive, with requirements increasing to match sensitivity.

4. **Refocus and reform existing certification processes.** There are already several security certification processes for software products, such as the Common Criteria, but these processes do not ensure that certified software products are capable of resisting hostile attack. The United States can lead an effort to streamline these certification processes, reduce their cost, and buttress them with best practices and software assurance tools.

5. **Identify commercial-sector best practices and tools and expand their use.** Many companies already have extensive software assurance procedures as part of their production processes. The processes include a sequence of internal reviews for performance and security, internal testing, external testing and red teaming, and the use of software review tools (some commercial, some proprietary and developed by the software company itself) to find vulnerabilities or errors. These practices offer the building blocks for an

approach that is most likely to succeed in reducing the risk of distributed production. Extending these best practices would improve software assurance and security overall and reduce any risk from hidden malicious code.

As part of this effort, the government could provide incentives and support for building better software assurance tools. As software programs continue to grow in size, investment in R&D for better tools will become more important for preliminary checks of the millions of lines of code found in many products.

6. **Create a governance structure (or structures) for assurance.** Companies may be taking extensive steps to improve software assurance, but if these steps are unknown or unmeasured, they cannot increase trust. Finding ways to overcome this is a crucial step for increasing trust in software products used for national security and critical infrastructure applications. It is essentially a governance problem. Traditional approaches to governance—command-and-control or regulation—do not work as well as they once did, or they may increase assurance at an unacceptable cost. An alternative solution is to create public-private partnerships to improve assurance. Whether this structure is formal or informal (and there are a number of existing groups that could be consolidated to serve this purpose), the objective would be to identify and share the best practices developed by software companies and shape requirements and procedures for better software assurance.
7. **Accelerate information assurance efforts.** Even if there were no foreign participation in IT production, networks would still be insecure. Networks involve thousands of different devices, some running older legacy code, others running unpatched programs, and all facing the possibility that they are vulnerable because of a configuration error found in a separate network to which they connect but do not control. In this environment, knowing who has accessed information, and whether they have changed it, copied it, or transferred it offers a more efficient way to improve security. Greater attention to accountability and transparency in information use—monitoring and safeguarding data at rest—can help manage risk. Emerging technologies for information assurance, use control, and better authentication and authorization can counterbalance network and software vulnerabilities by allowing networks to control who can access information and what they can see and do with it.
8. **Promote leadership in IT innovation.** Globalization and distributed production are unavoidable, but the United States can take steps to keep itself at the forefront of technology. Technological innovation is good for the economy and for national security. Innovation makes life more difficult for opponents. All of an opponent's work to "rig" one technology is wasted if a new technology appears and supplants it. Innovation can improve assurance processes, tools, and overall network and information security. Measures that improve the climate for innovation in the United States (such as increased funding for IT-related R&D) also help build a skilled domestic workforce, so that the United States does not find itself relegated to low-end functions or working off some other nation's designs.

Conclusion

We cannot dismiss concerns over foreign involvement in the software production process, but at the same time, we must frame any response to avoid damage to innovation and economic growth. There are measures that could improve information security and software assurance, but the issue of U.S. or foreign production is largely irrelevant to them. Globalization presents each nation with a mix of opportunity and risk. The best way for the United States to mitigate these risks may be to embrace globalization rather than attempt to block it. The good news is that the federal government is already focused on these issues and has begun to address them. We offer these recommendations, therefore, as part of a larger and ongoing process to make America more secure.

Foreign Influence on Software

Risks and Recourse

James A. Lewis

Introduction

We live in a period of intense economic change. The effects of this change go well beyond business. The motor of change is the desire of individuals to take advantage of the opportunities provided by international economic integration. This integration provides bigger markets and lower costs. The result is greater wealth for those who participate. A range of enabling technologies supports economic integration and is the basis of globe-spanning networks for communications, travel, and finance. Labor, technology, and capital are now highly mobile.

This is globalization. Some argue that globalization presents an overarching menace and that the country will be hollowed out or turned into a giant rust belt. These fears are unrealistic. By any concrete measure, globalization works to America's advantage. Rapid economic change always brings discomfort as markets, economies, companies, and workers adjust to the new environment. The result is faster economic growth in more countries. However, globalization creates new kinds of risks for national security and critical infrastructure.

One set of risks involves the potential for the United States to lose its technological edge. The increased international mobility of highly skilled labor and the diffusion of technological expertise mean that many countries can now compete with the United States in producing cutting-edge research and innovation. Technological superiority is crucial to U.S. leadership, and its loss would be very damaging.

The second set of risks comes from the dilution of national control over economic activity. This dilution applies to both ownership and production. In the past, security concerns over foreign ownership, influence, or control of a critical domestic producer or service provider involved the risk of illicit technology transfer—a foreign government would buy a U.S. company in order to gain access to its advanced technology. This is still a valid concern, but there is also a new risk—that a potential opponent would buy a U.S. firm to gain access to the services it provides, perhaps to monitor and collect information, or to disrupt them in time of crisis and, in the case of telecommunications, in peacetime.

A third set of risks comes from the effect of global economic integration on production. Globalization has eroded the idea of a national industrial base, in the autarkic sense of a nation being able to supply all the material, components, ideas, plants, and labor needed to create a particular product. Businesses call this a global supply chain. In looking at the many steps in the process that takes an idea and turns it into a product, we find that companies and workers in widely separated locations are much more closely integrated than in the past.

The global supply chain conflicts with old notions of trust and security. The United States could "trust" a product that came from U.S. factories and workers. Many would say that we cannot extend the same degree of trust to a product that comes from a foreign and perhaps anonymous source. In this case, there is a risk that an opponent will have deliberately and secretly tampered with components or products supplied to the United States. This mode of attack sounds unlikely, but the combination of globalization and changes in warfare suggest that while the risk may be exaggerated, it cannot be dismissed.

These changes involve asymmetric attack and information warfare. Infiltrating the software production process to secretly modify a program to provide later access or to damage it could be an attractive option to a potential opponent looking for "asymmetric advantage." A conventional force-against-force attack against the U.S. military is almost suicidal. It would be better to find a means to attack an area that is unsuspected and where U.S. defenses are weak. Gaining information superiority, whether through knowing more than an opponent or from disrupting his ability to know, has become one of the keys to success in conflict.

Information technologies are a primary target for asymmetric attack. Information—an array of intangible goods that include technological know-how, data, statistics, and news—and the networks and processing technologies that aggregate, process, and distribute it have become an integral part of national power. Creating and using new information is crucial for economic success. Information superiority provides an immense advantage for military action. As information has become more important, it and its associated technologies have become a key objective for both theft and disruption.

The issues surrounding foreign involvement in software production are made more complicated by larger fears about the effect of globalization on U.S. technological leadership, the seemingly intractable problem of information security, the very real threat of espionage, and the need to secure critical

infrastructure that depends on IT for its operations. All of these are difficult issues, and it is tempting for Congress or the federal government to look for simple solutions. Before we fix the problem of foreign involvement, however, we must first answer a fundamental question: is software written with foreign participation a principle source of this new risk, or does the source of that risk, and therefore the means to address it, lie somewhere else? If we do not answer this question, the result could be policies that damage our economic strength without benefit to national security.

Restrictions on where software is made are among the most dangerous of these damaging policies. Restrictions on foreign development (such as a country "black list") would not be in the U.S. interest for several reasons. In a global industry, attempting to block some locations will do little or nothing to improve security. It would likely trigger retaliation. The United States would be among the biggest losers in this scenario. Other nations, such as France and China, have considered the idea of restrictions on IT acquisitions—these are very often nontariff barriers to trade disguised as security measures—and the United States has opposed them. It would be a step back for the global economy if software production fractured into separate national fiefdoms.

Attempting to restrict where companies write or acquire code would also complicate the U.S. transition to new informational and network technologies, such as Web 2.0 or "distributed computing." These new, improved systems will incorporate and integrate many different software programs, written in many different companies and countries, into an interconnected global network. Attempting to restrict the "provenance" (e.g., the place of origin) of this software might slow U.S. adoption of these new technologies.

Overly strict restrictions could also backfire and damage larger U.S. security interests. The United States has been encouraging allies like Japan and potential new partners like India to open their defense sector to foreign firms. A decision to restrict software acquisitions, even if only for the U.S. defense market, would undercut these efforts. Alternatively, if the United States took the lead in creating processes for producing secure and reliable code, it would reinforce the market position of U.S. companies. If U.S. allies used this more secure code, it could improve their own security and performance and their ability to work with the United States.

The internationalization of the IT industry and the use of distributed coding processes create opportunities for the insertion of malicious code that may increase the risk to critical functions or infrastructure. Foreign intelligence agencies are opportunistic and likely to be attracted to exploring the prospects provided by malicious code. Assurance processes focused on ensuring that the code works are not optimal for detecting malicious insert. We do not want to overstate risk, but at the same time, we should not ignore it; and while the software production process is porous, the risks that this porousness can create are manageable.

The report that follows considers the nature and source of risk and potential steps to mitigate it. It recommends two sets of measures. The immediate concern is to improve software assurance processes and networks security to defend against malicious code insertions. The long-term or strategic concern is to take those steps that will improve security and maintain U.S strength in information technology.

Global Production of Software

Changes in the IT industry and the way it develops and makes products are the source of the new concern over foreign involvement. We have gone from national industries writing proprietary code for specific products to a global industry focused on interoperability. Research, production, and markets are dispersed around the world. The industry is highly competitive, and that competition drives companies to pursue talent and opportunities to lower costs wherever they are found. This global industry delivers immense economic and security benefits to the United States and the world and promises to continue to do so in the future. However, in recent years, its global nature has raised concern over the implications for security. The crux of these concerns is that the increasing production of IT outside the United States creates new vulnerabilities and new risks for security.

Some of the root causes of an increase in "foreign influence" in software products include cost, competitiveness, and capabilities. Labor is the largest component of software development costs. Foreign programmers generally cost less to hire than U.S. programmers—estimates suggest that, on average, the cost of programmers in Asia is 30 to 40 percent lower. These savings may be only temporary. Foreign programmers are becoming more expensive, and their cost advantage is shrinking, but at the same time, they are also becoming more skilled. Estimates on the number of software programmers or engineers in the world are imprecise, but most are still found in the United States. There are probably 15 million software programmers in the world. Half of these programmers live in the United States. Over the next few years, however, the number of programmers in China, India, Brazil, and Russia will increase at a faster rate.[1] This means that the involvement of foreign programmers in high-end programming and software design will increase.

Company strategies to increase their own competitiveness also increase foreign production. Many companies focus on core competencies and outsource functions that are less valuable or in which they have less of an advantage. Some companies emphasize their ability to manage complex projects and to integrate components from a global supply chain as a source of competitive advantage. All of these

[1] Government Accountability Office, *Offshoring: U.S. Semiconductor and Software Industries Increasingly Produce in China and India*, GAO-06-423 (Washington, D.C.: GAO, September 2006); ZDNet Research, "Number of software developers in China and India to grow at 25.6% and 24.5% a year," April 19, 2005, http://blogs.zdnet.com/ITFacts/?p=7667.

business strategies favor a distributed, international production process. There are other benefits for U.S. companies that develop software overseas. A "localized" strategy can increase sales. Local programmers are essential for developing or customizing products for the local market or for local needs. This lets foreign companies compete with local products that would otherwise have an advantage. Hiring developers or setting up a facility gives a company a presence in a country that can help to open the market.

An equally important set of factors that leads companies to produce overseas involves human capital (i.e., a skilled and well-educated work force) and research. These are two important elements for innovation. America's skill at innovation and its rich sources of human capital and research have made it a powerful economy. Other countries have taken notice of the U.S. "model" and are putting in place policies and investments to copy it. At the same time, changes in funding for education and research in the United States have led many people to ask whether the United States is losing the competitive advantages it once had. Whether this is correct or not, there is no doubt that nations in Asia are increasing their stocks of human capital. China, for example, has oriented its educational system to producing engineers and technologists. Foreign companies are attracted to this new supply of human capital and the global pursuit of talent has become another element of economic competition. The result is that an interrelated set of business practices and larger economic trends has moved and will continue to move software production outside the United States.

Changes in the processes for writing software could increase risk, or at least the potential, for malicious activity. Teams—often groups of teams—write software. Software products are designed or "architected" in a modular fashion. Sometimes they reuse older code originally written for some other program or purpose. Software programs can now run into the hundreds of thousands or millions of lines, creating complex management and integration challenges. These factors—and others—explain some of the concern over software assurance. It is difficult to know and assess the trustworthiness of large, complex programs written by geographically distributed teams, and these conditions can create opportunities for surreptitious intrusion into the production process.

As programs have become longer and more complex, the number of people involved in writing software has increased. Software is now often written in a distributed process, with teams in different centers and different countries contributing code to the final product. Project managers lay out the architecture, requirements, and design of a program; the work is then divided among several teams; the chunks of code that come back from these teams are then integrated into the final product by a team directed by the project manager. Testing for quality assurance and security may occur at several different stages, both on the code produced by a single team and on the entire integrated product. Without the appropriate safeguards in place, a distributed process could increase risk.

A company may have design cells or production groups located in several countries. They are tasked with creating or building parts of a new product, which the company will then assemble, test, and market. Other companies may contract

out parts of the work to a third party, another company, or an independent contractor who will contribute some element of the larger product. Many companies blend the two approaches, using distributed teams of their own employees for some tasks and contractors for others. Sensitive work may often be reserved for internal company teams while lower-value activities are contracted out. Each company has its own decisionmaking process on how to organize coding and production, and this provides part of its competitive advantage. Companies that do not take advantage of the distributed global supply chain, particularly high-tech companies, are at a competitive disadvantage.

Some companies engage in offshore activity through acquisitions. When an American company buys a foreign software firm, it acquires the foreign firm's intellectual property. These acquisitions are driven by a desire to add innovative products to a company's portfolio of products or to expand markets. Access to the foreign company's talent may also be a motive. The result is that an American company may incorporate foreign-made software in toto into its product. Acquisitions also lead companies to internationalize their workforce—an executive of one large company that has moved aggressively into the global market told CSIS that in 1990, 80 percent of its employees were American, while today, 57 percent are foreigners.

Many companies have taken advantage of the lower costs and large talent pool available on the global market. The results have been better products, faster development, and lower costs. Sending work offshore, either to company-owned facilities or to third parties working under contract, provides a competitive advantage that few companies can ignore. Some of the benefits of this approach are obvious—several companies use a process where teams in the United States, Asia, and Europe each write code for the same project, allowing work to be passed from team to team at the end of the work day. The U.S. team can hand off to the Asian team, the Asian team can then hand off to the European team, which can then again hand off to the United States, the result being 24 hours of work in a single day.

While much of the public discussion has fixated on cost—and foreign programmers tend to be cheaper—there is also a strong "human capital" incentive for companies. Countries like China, Korea, and India have invested in science and engineering education and R&D programs, and this has created a strong technical labor force in Asia. Although these countries face other obstacles to competitiveness and their educational systems are not as productive as the U.S. educational system, government programs like China's 863 Program have created a technologically skilled workforce that attracts foreign companies who want to tap into these new talent pools. Trends in U.S. investment in R&D and science and engineering education reinforce this—as fewer Americans choose to go into science and engineering or are paid to do so, the foreign workforce becomes a resource over which global companies will compete to employ.

One survey conducted in 2005 found that offshore software development of some code had become standard practice in the software industry and that the number of companies using offshore developers had increased by 25 percent since

2003. Although a distributed process creates management and integration challenges, these are outweighed by the cost advantages. A distributed process, however, also increases the opportunity for mischief. Open architectures for software products, which enable a modular approach to production and increase opportunities for interoperability, also lower costs and improve functionality, but this approach to project management can also provide potential avenues for malicious code insertion.

Companies outside of the IT industry are particularly attracted to outsourcing. Writing code is not their specialty. Subcontracting this sort of function to an outsider is increasingly a normal business practice. In some cases, a company will hire one firm to advise and assemble its IT systems, and this firm will hire a third company to write the required software. Some industries, such as the financial industry, have extensive safeguards in place to minimize the risk of outsourcing. In other sectors, security may not be a primary concern or even be considered at all. An assessment produced by the National Intelligence Council (NIC) in 2004 noted that technology now allows companies to have their most sensitive computer code written overseas, but that this can create a significant vulnerability. The NIC's director stated, "Outsourcing cuts costs but also brings potential security risks—particularly when it involves entities owned and operated abroad."[2]

Powerful motives drive the growth of outsourcing, but the increase in outsourcing does not automatically translate into an increase in risk. Companies use several strategies to maintain control and security, and some of these strategies reduce the risks of foreign involvement. For example, most companies are deeply concerned about protecting their intellectual property. Companies remain cautious when it comes to outsourcing high-level activities or research and development, and this limits the potential for disruption, but as offshoring becomes more prevalent and as foreign IT workers become more skilled, there will be pressures to move more work outside the United States. This trend is already apparent when we count the number of companies that have gone to Asia and parts of the former Soviet bloc to set up their own research centers or to contract out R&D work to foreign institutions.[3]

Managing these multinational teams and supply chains, with their mix of different technologies, standards, regulations, and processes, is a complex challenge. For security, the challenge can be increased by the use of "legacy" code—code written at an earlier time, often years in the past, for other programs. The problem for security is with the provenance of the legacy code. The code will function, but it may be inadequate for today's threat environment, or the programmers and processes used to develop it may be unclear or unknown.

---

[2] Robert L. Hutchings, "Terrorism and Economic Security" (speech by National Intelligence Council chairman to International Security Management Association, January 14, 2004, http://www.dni.gov/nic/speeches_terror_and_econ_sec.html).

[3] Mike Ricciuti and Mike Yamamoto, "Companies Determined to Retain 'Secret Sauce,'" CNET News.com, May 5, 2004, http://news.com.com/Offshoring+Companies+guarding+secret+sauce/2009-1022_3-5198605.html.

Company programmers, subcontractors, or some third party may have written it. Some companies also blend proprietary and open development processes. This means that they will create new programs that incorporate open source software in some of their modules. Again, the issue is provenance—the programmer who coded the open source software may not be known to all who use it.

Collaborative software production models continue to evolve as companies seek to make themselves more efficient. Competition drives them to this, and the results of this are very positive for consumers. Attempting to reverse this trend would only slow innovation and increase costs, if only by cutting off access to supplies of talent. However, the software production process intersects with and has implications for security for two groups of customers: the federal community and critical infrastructure industries.

The Department of Defense (DOD), the intelligence community, and very often the leading infrastructure companies once wrote almost all of the code they used, but today, only the most highly classified code is written at places such as the National Security Agency. Changes in defense industrial capabilities and the rapid pace and broad scope of technological change means that the military and other government agencies are increasingly dependent on commercial off-the-shelf (COTS) software. The defense industrial base, although still composed at the top of a few large contractors capable of assembling and integrating weapons systems, is merging with civilian industries that are part of a global commercial market. The use of COTS software reduces both cost and the time needed to acquire software, but the result is that code for some of the military's most sophisticated weapons—fighter aircraft and missile defense systems, for example—and even some secure systems—contain codes that have been written in other countries.[4]

The same trend is found in many industrial sectors besides defense. Code that was once proprietary and written in-house is now procured from the market. However, for national security and critical infrastructure protection, the increased use of COTS software may create new sources of risk. The ultimate source of the COTS product may be unknown to the purchaser or to a prime contractor, particularly if the COTS product is an assembly of software modules produced by different subcontractors. In some cases, it is not unusual to find several tiers of contractors—a prime and then one or two layers of subs. The customer, at best, may know only the first layer of contractors. COTS software can include large chunks of legacy code, which can come from unknown sources and, because it has worked successfully in the past, may not be subject to scrutiny. Mitigation and assurance efforts in both the private and public sectors have been largely carried out on an uncoordinated and at times ad hoc basis.

The move to greater use of COTS software by government provides benefits for cost, interoperability, and innovation. However, all agencies recognize that the increased use of COTS software changes how we must think about security.

---

[4] General Accounting Office, *Defense Acquisitions: Knowledge of Software Suppliers Needed to Manage Risks*, GAO-04-678 (Washington, D.C.: GAO, May 2004).

Greater use of COTS software and the changes in how companies make software mean that we can no longer make the same assumptions about security and trust. Planning for security must now take into account a very different model for software production. The goal should be to find a way to continue to reap the benefits of new models for software production while at the same time putting in place measures that increase the levels of trust in the products. The remedy lies in better practices and processes for assurance and security and better knowledge of those practices and processes for both producers and consumers.

The Potential for Risk

Attempting to exploit the global supply chain for advantage against the United States could be ambitious and risky, but in some circumstances, it will be attractive to potential opponents. One key question, however, is to ask just how attractive this is when compared to other modes of attack.

To answer this, we must put the threat of a malicious code insertion into the context of espionage, cyber security, and asymmetric warfare. All three inhabit a murky world where cause, effect, and attribution are never very clear. However, they are all very real threats. The United States is probably the world's leading target for espionage, including economic espionage. The cyber environment is fabulously insecure, as increasingly professional and skilled cyber criminals seek to exploit the Internet for gain. And while asymmetric attack remains more in the field of theory rather than practice, we know that many potential opponents are exploring various asymmetric attacks for a way to strike at the United States. While the most likely attackers are national governments and their intelligence agencies or militaries, advanced criminal gangs, terrorists, or even large global corporations could also attempt to use the global supply chain for software to gain advantage.

The primary threat comes from espionage. Counterintelligence officials testifying before Congress have stated that at least 41 countries engage in espionage against the United States. Foreign intelligence agencies are most likely to have the resources, skills, and inclination to take advantage of this kind of attack. "Fixing" technology to provide access is a standard trick, as is recruiting insiders, so a malicious code insertion would be only an extension of existing operational techniques. Communications intelligence has grown in importance in the last decade (as the volume of accessible and valuable data has increased), focusing intelligence agency attention on finding ways to access communications and information technologies, including the Internet.

Foreign intelligence agencies are accustomed to the use of cyber tools for collecting data from their targets and to placing individuals inside companies or commercial development centers to gain access to sensitive information. Public information on these sorts of activities is very limited. Foreign intelligence agencies could exploit opportunities provided by economic integration to insert or recruit personnel with access to critical functions in the United States. Production

of hardware and software, especially production at facilities in which the government has an ownership stake—not an unusual occurrence in some Asian and European economies—could provide new opportunities to foreign intelligence agencies. Foreign intelligence agencies could also take advantage of the increasing internationalization of business by inserting or recruiting personnel with access to code or to critical hardware.

Espionage is only one source of risk. Cyber attack is another, related danger. Cyber security is a broad topic area that can mean everything from consumer awareness of virus threats to state-sponsored attacks against critical cyber infrastructures, and there are indications that crime syndicates and nation-states have run well-planned tests on how to exploit cyber vulnerabilities to attack critical infrastructure.

We already know that the cyber environment is inherently risky. The events of the past few months illustrate the complex challenge that the United States faces in securing federal systems from cyber attack. In December 2006, computer networks at the Naval War College and other DOD facilities had to be taken off-line after being infected by spyware that was probably introduced by another nation. The National Aeronautics and Space Administration (NASA) put limits on e-mail attachments to avoid outsider attempts to gain access. The Department of Commerce had to take an entire bureau's computers off-line after they were hacked and infected with spyware, and this attack was likely part of a larger series of attacks launched against many federal agencies, affecting both sensitive and routine information.

The U.S. government is a primary target for hackers around the world. Politically motivated groups target U.S. networks to make statements about U.S. policies or actions. Foreign governments see hacking into federal networks as a crucial intelligence tool. The primary purpose of this hacking is the surreptitious theft of information, but there is also the real risk that opponents could infiltrate networks, implant "malicious code," and then wait to launch an attack to disrupt data and services during a crisis.

Businesses and consumers are also targets. Businesses can be targeted by foreign intelligence agencies or competitors who seek sensitive information or technical data. One troubling development of the last decade has been the growth of cyber crime, with skilled and specialized professionals targeting networks and information and developing the automated tools needed for their crimes. One principle collection of tools involves software programs that provide illicit access or control of a target computer. These programs are known as malicious code.

Malicious code is a term used for programs that, when executed, cause undesired results on a system. There is a wide range of names for malicious code, including spyware, trojans, viruses, keyloggers, bots, and root kits. Users are usually not aware of the code until they discover the damage. Attackers hide malicious code in legitimate programs that they have altered. These sorts of programs can lead to serious data loss or denial of services. The preferred method for introducing malicious code is delivery over the Internet, using programs that

exploit vulnerabilities in existing software products. However, there is an alternative method. An opponent with the right access could insert malicious code during the software development process, as the code is written.

Most programs now involve hundreds of thousands or millions of lines of code, making malicious code difficult to detect. Manual scanning is prohibitively expensive and not always effective—malicious code may not be detected even if a programmer is looking directly at it. Existing software development practices focus on reducing defects and improving performance. Automated quality assurance programs created to detect coding errors or vulnerabilities still miss unintentional errors, so they are even less likely to detect intentional changes in places where some thought has been given to hiding the lines in question.

The surreptitious installation of malicious code on a computer is the core of many cyber attacks. Viruses, worms, and spyware reach the target computer through the Internet and either implant a small new program or modify existing programs to give the attacker access to and control of the target computer without its user's knowledge or consent. Much of the effort in the cyber-crime community (and in intelligence agencies) is to review software products, looking for vulnerabilities they can exploit with this kind of "backdoor" attack. A malicious code insertion has the same objectives, except that the malicious act occurs during the production of the software, not after it has been deployed. In considering whether to introduce malicious code during production or after, an attacker would likely ask which approach is most likely to succeed, which approach costs the most, and which approach has the greatest likelihood of detection.

Existing standards for software evaluation do not adequately address the use of foreign-produced software or the potential risk of the insertion of malicious code. A product could be validated and still contain malicious code. It is this uncertainty that helps to create the concerns over foreign participation in software production. These concerns must be placed in the larger context of information security defined by several crucial reports published in the last decade.

In October 1997, the Presidential Commission on Critical Infrastructure Protection, chaired by Gen. Robert Marsh (USAF, ret.), predicted that the increasing reliance of the United States on critical infrastructures and cyber-based information systems would also create "a new dimension of vulnerability." The commission predicted that unconventional attacks on infrastructure and information systems by a range of new opponents could be capable of significantly harming both the U.S. economy and U.S. military power.[5]

Malicious code inserted surreptitiously by an insider during the software production process could be an effective element of a larger intelligence collection program or information warfare strategy. A 2004 study commissioned by the Defense Department concluded, "Software provides the greatest mechanism for technical exploitation by a wide range of adversaries." The study identified foreign-produced software as one area (among several) of potential risk.

---

[5] President's Commission on Critical Infrastructure Protection, *Critical Foundations: Protecting America's Infrastructures* (Washington, D.C.: PCCIP, October 1997).

One reason that this may be attractive is that the chances of discovery are small, particularly if the malicious code is to function as a "time bomb," to sit quietly and be triggered only in the event of need or conflict. Even if the code is discovered, it may be impossible to attribute responsibility to a particular opponent.

We can find testimony to the strength of this concern in an unexpected quarter. In 2004, in an editorial explaining why China needed its own IT industry, *People's Daily* (Beijing) wrote, "Strategists reveal that in peacetime, the United States sells virus-carrying chips as ordinary commodities to…other countries. When needed in war-time, the United States can remote control and activate the virus at anytime, making ineffective or paralyzing the enemy's commanding and weaponry systems."

The risk from the insertion of malicious code during development is not a hypothetical threat. In 2003, for example, there was an attempt to implant malicious code in the Linux operating system code. The malicious code itself was small in size and relatively simple, but it would have allowed backdoor access to computers running Linux based on that kernel. We know of this case only because the Linux community detected and thwarted it. Judging from other detected examples, malicious code differs from other forms of attack in that it does not look for or exploit existing vulnerabilities. Instead, it creates a vulnerability that is accessible only to those who know of the addition, making the product appear otherwise secure.[6]

There are few publicly known examples of malicious distortion of code. While it is not (usually) a malicious distortion, most people are familiar with software "Easter eggs." These are segments of software code hidden in larger programs. There are well-known examples of Easter eggs where games have been inserted into another program. Easter eggs are usually harmless and entertaining, but they are hard for customers to discover and often are only found after programmers leak information on how to access the egg. Suppose, however, that a programmer decided to hide an Easter egg in a program for criminal purposes. In fact, the code needed for some malicious purposes, such as unauthorized access to data or networks or remotely triggered disruption of data or functions, could be much smaller than an Easter egg, perhaps only a few lines.

One publicly known case involving interference with the production process was the Farewell project, a U.S. intelligence operation in the 1980s designed to take advantage of Soviet efforts to illicitly acquire Western high-technology, including computers and microprocessors. Under the Farewell project, U.S. companies deliberately introduced modifications into technologies (such as computer chips) that the Soviets planned to steal. Farewell is not a meaningful precedent, as the markets and technologies of the 1980s were so different from conditions found today. The economic, technological, and political conditions

[6] Matthew Broersma, "Server Breach Raises Linux Code Worries," CNET News.com, August 14, 2003; Robin Miller and Joe Barr, "Linux Kernel Development Process Thwarts Subversion Attempt," *NewsForge*, November 6, 2003.

under which the Farewell operations took place were, of course, very different. First, a member of the Soviet KGB (whose code name was Farewell) was willing to provide a list of the specific technologies the Soviets were trying to steal. This allowed the United States to focus its efforts. Second, the high-tech sector was much smaller and its products were much more "proprietary"—using code that tended to be written in-house rather than outsourced. Finally, U.S. laws were particularly clear that the exports in question were illegal. This more constricted business and legal environment made it easier to use a malicious code insertion.[7]

Another example involved the Bell Labs programmer who developed the widely used UNIX operating system. He described in a speech how it was possible to hide a backdoor in the code that would give him access to every computer running UNIX. One of his concluding remarks was, "You can't trust code that you did not totally create yourself."[8]

In the 1990s, when the United States still tightly controlled access to encryption software, an unknown foreign entity posted a powerful and restricted encryption program as freeware on a European server. This encryption product was unbreakable—with one exception. There was a hidden vulnerability that the unknown foreign entity could exploit. If we assume that the foreign entity collected a broad range of communications and automatically processed them, it could automatically decode a communication encrypted with the "fixed encryption."

A series of studies produced by government agencies have concluded that the United States faces a dilemma created by global production. The common conclusion of these studies is that the United States has little choice over participation in globalization, and while the benefits of this participation can be great, there can be at the same time new and different challenges for security that current policies do not adequately address.

Concern about the potential for the insertion of malicious code into foreign-made software appeared almost a decade ago. A seminal Defense Science Board (DSB) report on globalization and security (published in 1999)[9] took a broad look at the challenges created by international economic integration and its effect on the defense industrial base and on national security. Its central theme was that the United States must adjust to an environment where, in the future, the sources of supply for the U.S. military would be global.

The globalization report called on DOD to "take full advantage of the commercial sector," but it also warned, "The Department must act aggressively to ensure the integrity of critical software-intensive systems." The authors concluded

---

[7] Gus Weiss, "Duping the Soviets: The Farewell Dossier," *Studies in Intelligence*, March 2001, https://www.cia.gov/csi/kent_csi/docs/v39i5a14p.htm.
[8] Ken Thompson, "Reflections on Trusting Trust," *Communications of the ACM* 27, no. 8 (August 1984): 761–763.
[9] Defense Science Board, *Final Report of the Defense Science Board Task Force on Globalization and Security* (Washington, D.C.: Office of the Undersecretary of Defense for Acquisition and Technology, December 1999), 115.

in 1999, "It is not feasible to vet the software for malicious code, nor is it a simple matter to prevent grave damage from trusted insiders." The report urged DOD to adopt a series of measures to help secure "computer-based command-and-control activities," including a "performance-based trustworthiness regime," development of "trusted factories" (for hardware), and a greater use of technologies like field-programmable gate arrays (which are more secure, because the code they use is not irreversibly embedded when the chip is made). The report's authors believed that "the need to take appropriate risk-mitigating measures is urgent."

Several of the annual reports (from 2002 to 2005) to Congress by the National Counterintelligence Executive identify malicious code insertions as a growing area of vulnerability. The 2002 report states the following:

> Although we have no evidence to date that foreign countries have attempted to insert malicious code into products sold to the United States, remote and malicious code insertions are major threats for the future. Foreign firms are becoming dominant in the production of key IT hardware components and software, and it is possible in the future that such products could even end up in highly classified closed systems. In that event, there would be opportunities to introduce difficult-to-detect code capable of corrupting those systems or, perhaps more importantly, of covertly sending sensitive information back to foreign providers.[10]

Johns Hopkins University's Applied Physics Lab (APL) produced another study that looked at the implications of foreign involvement in IT for the Department of Defense. The focus of the study was on how the security of one critical infrastructure, that of the U.S. telecommunications infrastructure, would be affected by foreign ownerships and investment.[11]

The APL study began with the question of how the United States could continue to reap the benefits of open investment and free trade in the telecommunications sector while managing the national security risk. The study enumerated a variety of possible risks that the telecommunications infrastructure faced, including the risk from foreign ownership.

The fundamental problem created by foreign ownership, APL concluded, was that it could give the foreign owner the ability to make significant technology decisions for crucial U.S. services. One set of risks identified early in the report included the possibility that a foreign owner could compel its U.S. subsidiary to purchase foreign-made technology and, if the foreign owner had malicious intent, perhaps guided by its home government, it could embed features in the systems that allow control over key aspects of network operations. The APL study identified software as the greatest mechanism for technical exploitation by a range of adversaries.

---

[10] National Counterintelligence Executive, *Annual Report to Congress on Foreign Economic Collection and Industrial Espionage—2002* (Washington, D.C.: NCIX, February 2003); see also the 2005 report, p. 5.

[11] Applied Physics Lab, *Protecting the United States Telecommunications Infrastructure: The Way Forward* (Baltimore: Johns Hopkins University, spring 2005).

APL's study considered a number of recommendations to improve security. These included having the government conduct regular security assessments, based on estimates of potential attacks and corresponding mitigation measures; developing a two-tiered security policy, with one tier being the development of security standards and strategies by the government and the second being the establishment of a specialized secure network for critical government and private-sector needs; improving government review of foreign transactions; and improving procedures for risk assessment and information sharing. The general theme of these recommendations is that an informed and consistent process should replace an ad hoc and irregular approach for identifying and mitigating threats.

APL's conclusions about the risks of foreign ownership of the telecom infrastructure have major implications for the issues created by foreign involvement in software production. Foreign ownership or production is a source of risk, but not the only source. Closing off foreign ownership and production would do real economic damage to the United States, but would not close off all of the avenues for a determined opponent to gain access to (and perhaps control of) information systems and networks. Implicit in the study's conclusion is the idea that foreign ownership and influence is a symptom of the larger problem. This problem is how to reconstruct trust when the old assumptions about national production no longer apply.

The Government Accountability Office (GAO) wrote several reports on the problems posed by foreign software. GAO's May 2004 report, *Defense Acquisitions: Knowledge of Software Suppliers Needed to Manage Risks*,[12] looked at the risks posed by potentially malicious software. GAO found that DOD's reliance on software and information systems is significant and increasing, but at the same time, traditional DOD prime contractors are subcontracting more of their software development to lower-tier and sometimes nontraditional defense suppliers. Those suppliers, the GAO added, use "offshore locations and foreign companies" for some software development.

GAO found that two-thirds of the program managers in the 14 major weapons programs it reviewed had little or no knowledge of the foreign suppliers involved in their programs. Eighty-one percent had no knowledge of who developed the COTS software used in their programs. In some cases, the major contractors also had no insight into the source of the software they were using, and most contractors did not identify foreign involvement in software as a risk to be addressed in meeting performance requirements.

The GAO report concluded that DOD acquisitions and software security policies did not require program managers to identify or manage the risk of using foreign suppliers to develop weapon-system software despite what GAO called the "inherent risks." DOD security initiatives tended (at the time GAO conducted

[12] General Accounting Office, *Defense Acquisitions: Knowledge of Software Suppliers Needed to Manage Risks*, GAO-04-678 (Washington, D.C.: GAO, May 2004).

its review) to focus on external threats and not on the possibility of "the insertion of malicious code by software developers."

The report noted that, given the increasing number of lines of code involved, testing had become more difficult and costly and tended to focus on functionality (whether the program worked as intended) rather than on security. It found that by themselves, existing software assurance practices did not address the development of malicious software. GAO also noted that DOD could not monitor worldwide software development or provide clearances for all software developers, but neither could it afford to exclude all foreign software producers.

GAO's recommendations were to make security an integral part of the software acquisitions process; identify potential risks early in a program; establish requirements for program managers to exercise oversight of software suppliers; and, with the help of other DOD organizations, collect and maintain information on software suppliers. In response to the report, DOD agreed "that malicious code is a threat not adequately addressed in current acquisitions policy and software security procedures." DOD said that it would strengthen risk management to focus on comprehensive testing to discover malicious code, develop process evaluations for suppliers, and make greater use of counterintelligence threat assessments.

The most recent report to consider the problems created by globalization is the 2005 Defense Science Board report on high performance microchip supply.[13] It concludes that the rapid migration of semiconductor manufacturing plants to locations outside the United States is an "alarming" trend for security and economic well-being. The Task Force on High Performance Microchip Supply says that the U.S. semiconductor industry cannot change the competitive dynamics that shift production and markets away from the United States. DOD must take the lead, because it needs a "trusted" supply of chips. "There is no longer a diverse base of U.S. IC fabricators capable of meeting trusted and classified chip needs," says the DSB task force. The study calls for the United States to reexamine its industrial policies.

Trustworthiness and supply assurance for components used in critical military and infrastructure applications are casualties of the migration. The shift from American to foreign IC manufacturers could endanger the security of classified information embedded in chip design; additionally, it opens the possibility that "Trojan horses" and other unauthorized design inclusions may appear in unclassified integrated circuits used in military applications. The task force says that the federal government and the DOD have no strategy to maintain technological leadership.

The task force also called for strategies to assure and maintain U.S. technological superiority in microelectronics. It says that the Defense Advanced Research Projects Agency (DARPA) should consider creating a new Sematech-

[13] Defense Science Board, *Defense Science Board Task Force on High Performance Microchip Supply* (Washington, D.C.: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, February 2005).

type of research consortium. It calls for doubling the budget for the National Science Foundation and for funding the National Institute of Standards and Technology's Office of Microelectronics Programs.

Several scenarios illustrate potential methods for a malicious code insertion. An opponent could take a mass market or broadcast approach to get a maliciously altered product into the hands of a target. A broadcast approach would build vulnerabilities into widely used software or hardware products. While most of the people who bought the product would be of no interest, an attacker could hope that a target of interest would acquire it. Given the range of things that people attach to networks—from digital cameras and MP3 players to printers and computers—a product sold in the millions has a good chance of reaching a target of interest. The attacker would then probe networks of interest to see if the device had been attached, or the device could automatically report to its creators once it was installed. This sort of approach is common in spyware.

For some critical components (servers, switches or routers for major telecommunications service providers, and the software that runs them) that come from a foreign supplier, a government could induce the supplier to build in a vulnerability for it to use later, after the items are sold. Concerns over this sort of attack sometimes come to light as part of the Committee on Foreign Investment in the United States (CFIUS) process, when agencies move to block or restrict the foreign acquisition of a U.S. company to prevent a potential opponent from gaining insider access to products or services.[14]

The problem for malicious code insertion may be greater for embedded software (e.g., software used for instructions to microprocessors or other microelectronic devices), as the code is unavailable for inspection and the developer of the code for the device is usually unknown to the ultimate purchaser. This software is "burned" into the chip and usually impossible to inspect. Production of microelectronic devices has shifted to the Pacific Rim, and Asian countries would like to see their microelectronics industries move from low-value activities to high-value tasks like design and software writing for microelectronics. This growing dependence is a source of concern for the security community. The infrastructure vulnerability risks posed by embedded code will increase over time as sources of supply diversify and are no longer national.

Assessing Risks

One way to assess the risk from the insertion of malicious code during production is to look at the portfolio of potential attacks available to an opponent. An attacker has many different options for gaining access to networks and information. Exploiting a global supply chain is one of them, but it is not the best. An attacker will ask which mode of attack is most likely to succeed, how much risk there is in mounting the attack, and how much it will cost. As in business, low cost, low risk,

[14] A recent case involved Sourcefire, a U.S. company that made software used by many government agencies.

and high return are the preferred outcomes. A malicious code insertion does not compare well with other modes. Finding and exploiting existing vulnerabilities in networks remains more attractive. Recruiting insiders to provide access is also a time-honored and successful strategy. A well-resourced opponent may choose to pursue all three modes of attack—hacking, insiders, and supply chain.

Intelligence agencies have the wherewithal to tap into the skills and technologies developed by cyber criminals and the hacker community. This means that even a small country could use hacking to supplement its global intelligence collection activities. It also means that the cost of hacking, if governments could buy tools and services from cyber criminals, could be less than the cost of corrupting the production process to insert malicious code. Again, this is not an imaginary scenario—this occurred as early as 1989, when a foreign intelligence agency hired private hackers to penetrate U.S. government networks. The differences in cost and risk between infiltrating production to insert malicious code and hiring hackers to launch an attack suggest that malicious code insertion is unlikely to be the first choice of an opponent.[15]

Even if software were flawless, it would not guarantee security. Using highly secure software in an insecure network environment would bring no benefit. The security of a network depends on many other factors besides the software it is running. Improperly configured networks are vulnerable. Networks with many different kinds of devices attached to them or running a range of software products (including legacy code not designed for network use) may be vulnerable. Networks that rely on passwords for controlling access are vulnerable. Social engineering—the use of fraud to manipulate network users into revealing sensitive information that allows access to a network—remains effective (and an area where intelligence agencies have some skill), no matter how secure the technology.

This review of the alternate modes of attack available to opponents to use against information or networks suggests that the risk of using a foreign supplier versus domestic supplier should not be overstated. Yes, a hostile group or government could have a freer hand in putting pressure on an employee to insert malicious code, or could have a greater chance of getting one of its agents hired by the foreign firm. But there is nothing to stop them from recruiting in domestic firms. To begin with, the concept of domestic firm no longer makes sense in some industry sectors. High-tech industries have a global work force, in both management and in research and development. A Chinese company will have Americans and Europeans on its staff, and an American company will have Asians in many positions. When we compare foreign and domestic firms, we can say that neither is risk-free. Malicious code insertions are a kind of insider threat, and the insider recruited by a foreign intelligence agency can be, as experience has shown, an American with security clearances, recruited in the United States. The distinction between foreign and domestic really does not make sense. Making

[15] Cliff Stoll, *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage* (New York: Simon and Schuster, 1989).

the location of software production the focal point for mitigation will not result in more secure code.

The focus of security and assurance efforts have been primarily on the later stages of the software life cycle, after the product has been developed and installed, and on the unintended vulnerabilities in the product that an attacker could exploit. But there is a risk that malicious code could be inserted into software during development, before the software is compiled into a product and before it is purchased or installed. Risk lies in the software production process, not location.

The common themes in the federal reports discussed above are that reliance on a global supply chain for IT products creates potential and growing vulnerabilities; that software is a particular area for concern; and that the United States needs new processes to deal with the risk created by a global supply chain. The reports offer a range of recommendations, including greater transparency into who actually writes codes, development of trusted sources of supply, and greater attention to processes for assurance and security. This last recommendation suggests that an emphasis on taking advantage of and harmonizing company best practices found at leading software firms offers the best chance for mitigating the risk of a malicious code insertion.

Company Practices

At this point, we can draw two conclusions. First, the insertion of malicious code during production is possible, although it is not the most probable mode of attack on information systems and networks. Second, the location where software is written is less important for determining security than are other factors. These other factors grow out of changes in how companies write software, and this set of changes offers one possibility for mitigating risk. Companies have responded to market pressures for improved security by putting systems and processes in place for assurance. Assessing the strength of these internal processes by which software is written and tested provides a better response to risk than an emphasis on location or nationality of production.

Security has become a central concern for most software companies because of market pressure. Customers want more secure software, and the last few years have seen companies reorient their processes to make security an integral part of the product. Having the checks and procedures in place to detect inadvertent errors or malicious code inserted by an insider addresses the real concerns with foreign involvement. Some companies also provide security training for their programmers to alert them to the kinds of coding errors that can create vulnerabilities that hackers can later exploit.

We can debate whether this has gone in far enough or fast enough in remedying security defects, but the trend is encouraging. Software today is more secure than it was five years ago. In looking at what companies do to increase the security of their products, there are three sets of activities. The first is in the design phase, to

avoid architectural errors that create vulnerabilities and to consider the threat profile a product might face. The second is in the assurance phase, where it is standard practice to use checks, code reviews, and assurance tools to check the new product. Finally, some companies use a variety of external tests—such as red teaming and penetration testing—once the software product is almost ready for release, to discover vulnerabilities. The measures used by companies in all three phases of production can increase assurance and mitigate the risk posed by the development process.

The existing company practices offer the building blocks for an approach that is most likely to succeed in reducing the risk of distributed production. The processes include a sequence of internal reviews for both performance and security, testing by companies or individuals external to the company, and the use of software review tools (some commercial, some proprietary and developed by the software company itself) to find vulnerabilities or errors.

Companies can use distributed production processes to reinforce these measures. The principle means by which to achieve this is to make sure that the team that writes the code is not the team that checks it. Having a dedicated security and assurance team (as is the case with some major software producers) that can provide advice during the development process and perform an independent check once the product is completed also increases the likelihood that vulnerabilities will be detected and corrected.

It is worth noting that while a distributed approach to software production can create new risks, it may also reduce them. In the distributed approach, groups in different countries (company employees and contractors) work on chunks of code for a product, but the overview of how these pieces fit together remains with the project manager, the quality assurance team, and a few others. Companies retain the integration of the different pieces of code into a final product for themselves. Companies also limit what is shared with any external or offshore team in order to reduce the chance that intellectual property will be stolen. A member of one of the distributed teams or a contractor may not know if later changes made by the project manager, quality control team, or security reviewers will affect any insertion. Risk is reduced if there is transparency and tight control over the integrating process, where many sections of code are melded into a product. Integration provides an opportunity to discover vulnerabilities.

Most companies audit any change made to code during development and track who made that change. The primary purpose of this auditing software is to provide better management and control of the distributed coding process. Companies augment audit trails with authorization software that limits the ability to change or add code to those who have been granted access, so that programmers cannot freely access areas where they are not assigned work. Auditing software, combined with robust management practices, greatly reduces the risk that a rogue coder will surreptitiously slip malicious code into a product.

Automated software review tools appear to be an attractive solution, but in fact, they are not the most important measure for mitigating risk. This does not mean

they should not be used, but a risk-mitigation effort cannot rely on software tools alone. These tools are special software programs that review the code and test its functionality. Many companies use some sort of software assurance tools. Some are available on the commercial market; others are developed in-house. These software tools review code and look for errors. In an ideal world, the entire quality assurance and security review could be automated (because the review entails dealing with millions of lines of code), but the current state of the art means that no automated tool or set of automated tools is sufficient. Automated tools are good at discovering coding errors—bugs—where a programmer has made a mistake, but they are less likely to detect more complex, "design level" vulnerabilities. Over time, as automated tools improve, their value will increase and federal support for research to improve software-checking tools might be one way to accelerate this improvement for the coding process.

Those companies that have independent security review teams augment the quality assurance process and the use of automated tools by adding a step to the production process where code is reviewed for vulnerabilities. These reviewers have specialized knowledge of potential threats and vulnerabilities that allows them to identify mitigating strategies that can be embedded in the software to reduce risks.

Finally, some companies conduct red-teaming and penetration attacks against their own products before they launch them onto the market. Some hire contract hackers to attack the product as it runs in a test setting. A few companies even allow these contractors to see the source code of the products they are attacking. Others conduct their own penetration testing using company employees.

The practices and procedures at each company vary, but an appropriately constructed process could let the government use best practices to create metrics for assurance and use these metrics to guide acquisitions and policies. The key to these new metrics should be to identify and build on what is already done for assurance within the private sector. Software producers realize the importance their customers place on assurance and security and are adjusting their internal procedures to meet this market demand. While there is much commonality and overlap in what companies do, each company approaches the issues of assurance and security somewhat differently. From these differences, we can extract best practices and requirements that will address, as part of a larger solution set, the risks posed by foreign involvement in software production.

There are already standards and processes in place to review practices and evaluate products or processes for security. They include the Common Criteria, the Carnegie Mellon Software Institute's Capability Maturity Model (CMM), the ISO 9000 series, ISO 19779, SAS 70, FISMA and NSTISSP 11, and the National Information Assurance Program (a U.S. government initiative launched by the National Institute of Standards and Technology and the National Security Agency). The Department of Homeland Security is developing another evaluation process. All of these efforts have strengths, but the common industry view is that the evaluation processes are inadequate for the task of increasing security and assurance.

The Common Criteria are the most important of the existing evaluation processes. The Common Criteria provide a framework for security requirements and software production that can be evaluated and certified by independent testing. This testing is done by labs in both the United States and overseas, and one of the strengths of the Common Criteria is that it is an international standard adopted by 23 nations (most of whom are U.S. allies). There are 43 laboratories around the world that test software products against the criteria—9 of them in the United States.[16] The evaluation process also establishes the level of confidence that users can assign to a product. The Common Criteria provide assurance that producers have followed rigorous processes for the specification, implementation, and evaluation of a software product. Most Common Criteria evaluations have been for IT components (such as operating systems or firewalls).

While the Common Criteria apply to products, Carnegie Mellon's Capability Maturity Model (CMM) applies to processes. The CMM provides standards for software production that grade companies on quality control and schedule. The Carnegie Mellon Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense, develops these standards. The Capability Maturity Model looks at the software development process, project scope, software design, development, and testing. It uses different levels to rank companies' quality control processes. The highest level, level 5, means that a company can ensure predictable, consistent results when it undertakes a software project. The Capability Maturity Model provides a standard that allows companies to assess risks in outsourcing software development.

Other standards include Federal Information Processing Standard (FIPS) 140, issued by NIST to set requirements and standards for cryptographic modules for both hardware and software components used by the federal government. The requirements cover the cryptographic modules and their documentation. However, FIPS 140 was not intended to guarantee that a module conforming to its requirements is secure, or that a system built using such modules is secure.

Each of these procedures is beneficial, but there are several problems with them. The review process is cumbersome and time-consuming—a review can take more than a year to complete and some industry sources report reviews lasting more than two years. Many reviews involve a kind of paper exercise—companies file thousands of pages of documentation on the software to satisfy auditors. Criteria used for evaluation can be outmoded and slow to change. The idea of some external set of best practices against which companies' production processes can be evaluated remains worthwhile, but the existing implementation needs to be improved. Reviews can cost tens of thousands or even millions of dollars. This is a significant impediment for smaller and more innovative companies. Most importantly, it is possible to meet or surpass the existing standards for certification and yet still be insecure.

A more robust assurance process could use both a private sector–driven best practices process combined with the existing product reviews like Common

[16] See http://www.commoncriteriaportal.org/public/consumer/index.php?menu=7.

Criteria. One solution is for industry to develop the rules and for government to enforce them, perhaps by linking the certification of processes to the acquisition process. This could reinforce Common Criteria certification. By setting acquisition requirements, government can drive software assurance without prescribing how companies produce software. In this acquisitions-driven process, there could be cases involving less sensitive applications, where a company could self-certify that it has followed best practices. In other cases, some kind of external review might be necessary in addition to self-certification.

One major problem with self-certification is the lack of enforcement mechanisms. There are a number of ways to address this. Attaching liability to self-certification as part of the acquisitions process builds in an enforcement mechanism—a company that certifies it has done something is liable if it turns out later not to have actually taken the necessary steps. A blended approach that combines self-certification of development and coding processes and Common Criteria could help to reduce this problem. Process and product certification requirements should be scaled to different levels of assurance—functions requiring high assurance software would need more robust review and certification procedures; routine functions would need to meet a smaller set of requirements.

One way to achieve this would be to use partnerships between industry and government to develop, refine, and share best practices and tools. There are a number of existing groups under government leadership that could be used to accomplish this, but one important change in the role of government should be considered. Most of these groups confine the private sector to an advisory role. Agencies make the decisions and set standards. It would be better to copy private-sector standards bodies—such as the Payment Card Industry Standard. In this sort of group, government would be a participant but not a ruler. This might provide greater flexibility in the development and modification of best practices. Government could still drive the process by using its power over acquisitions to establish requirements or assurance levels, and a government presence is essential to avoid having any set of best practices settle on a “least common denominator” set of requirements for assurance.

The Common Criteria and the National Information Assurance Partnership (NIAP) process are reinforced by the Committee on National Security Systems (CNSS), a DOD-chaired forum with representatives from 20 federal agencies and with extensive private-sector participation. CNSS sets policy for securing national security systems. One CNSS requirement is that only COTS software that has been certified under the Common Criteria can be used on national security IT systems. CNSS may offer a vehicle for developing better approaches to software assurance.

The use of criteria and evaluation, embedded in a more flexible and responsive system, would mitigate the risk of foreign involvement in software. The core processes to be drawn from company best practices are as follows: security training for programmers; a design or software architecture that includes security considerations from the first; strong management procedures that provide

oversight and transparency and track who has modified or added code; an independent review of code for security issues (including the use of software assurance tools) that occurs throughout the production, not just at the end; red teaming; and penetration efforts by specialist contractors. While competitive pressures will move companies to adopt these practices, the government can accelerate this process, guide and require it for sensitive applications, and play an essential role in coordinating private-sector efforts to increase assurance.

Innovation Reduces Risk

Concern over foreign involvement in software reflects a larger set of worries involving the effect of globalization. As globalization reshapes the U.S. economy, many fear that it will erode America's strength. The most serious problem created by globalization is not outsourcing, offshoring, or the loss of manufacturing; it is the effect on America's ability to innovate. Innovation is the development of new products, services, or methods of production through the application of scientific research or invention to commercial or military activities. Innovation provides the flow of ideas that benefits both the commercial market and a military that increasingly relies on commercial technology. Innovation is the key to economic growth, and it provides crucial security benefits. The central question for a strategic view of the security implications of foreign software production is whether the United States is losing its ability to innovate.

The evidence for this loss is mixed. Declining enrollment in science and engineering courses and flat budgets for research suggest a weaker, or at least different, innovation system in the United States. In the past, the United States has been able to compensate for some these weaknesses through its ability to attract foreign talent. Other countries now compete for this talent, and the attractiveness of living in suburban America is less than it used to be. Current U.S. visa policy also works against bringing foreign talent to this country, and this policy and the more onerous regulations put in place after September 11 have encouraged companies to open research and developments centers overseas.

There has been a shortfall in key basic research areas for many years. The United States has underfunded key areas of basic scientific research since 1975 (measured by share of GDP). The effect on national security is concentrated in mathematics, computer science, and engineering. In relative terms, these areas have been the most seriously underfunded. One result of this underfunding is that the number of undergraduates getting degrees in computer science is falling in response to student perception of job opportunities. While 20,000 bachelor's degrees in computer science were awarded in the 2003–2004 academic year, new enrollment in computer science programs has dropped for a fourth year in a row, falling 10 percent in the 2003–2004 school year and 39 percent since 2000.

Recruitment to the IT workforce is still recovering from the dot-com boom, yet the demand for high-level skills continues to grow. The shrinking technology workforce could become a problem in the next few years if computer graduates

decline while the United States tech industry grows. The number of undergraduates getting degrees in computer science is falling in response to student perception of job opportunities. New enrollment in computer science programs has dropped by almost 40 percent since 2000. While the demand for high-level IT skills continues to grow, the United States is producing fewer IT high-skilled workers. The shrinking technology workforce contributes to the risk of a global software supply chain.[17] Better-funded programs will attract more American students into information technology–related fields and help to build the human capital needed for secure software.

Some degree of change is inevitable as other countries develop their economies and their scientific establishments expand. Globalization will produce these outcomes irrespective of any U.S. policy. This means that China, India, and other developing nations will join the United States in the ranks of economic and innovation powerhouses. However, the United States can mitigate many problems created by globalization if it protects and expands its ability to innovate.

The strategic issue for security is whether the United States has the right policies in place to minimize risk and maintain technological leadership in the face of globalization. The simple answer to this question is that the United States could do better than it is currently doing, and a failure to strengthen its innovation capabilities will make reducing the risk of foreign involvement in software more difficult.

While the United States spends more on research than any other nation, changes in the composition of spending and increasing budgets overseas will work against national security unless the United States adjusts its policies. The reasons for this reflect a number of factors. First, the decision in the 1990s to focus federal support for R&D on the life sciences while keeping funding flat for the physical sciences and engineering, including computer science, has led to a slowing of research in these areas. Given the length of time between when a scientist makes a discovery in the lab and when this discovery appears on the market as a product, the result of this slowing may not appear for some years.

Second, changing research investment priorities reduce the supply of Americans with expertise in programming. A college degree has become more expensive, and a smaller percentage of high school graduates can afford to attend college than in the past. Science and engineering are less attractive career paths—if there is no funding for research, there are fewer positions and fewer interesting projects to work on; people do something else. Students remain cautious about entering an IT career track, given concerns over the dot-com bust that still linger on campuses—a perception that IT careers do not offer long-term opportunities when compared to other fields.

Third, changing U.S. policies and growing foreign competition reduce our ability to "import" talented new technologists. Since the 1930s, when leading European scientists fled Europe, the United States has gained from an influx of

[17] Michelle Kessler, "Fewer Students Major in Computer," *USA Today*, May 23, 2005; "IT Staff Shortage Looming," *CIO Today*, August 15, 2005.

foreign talent (the "brain drain" of which so many countries complain). This brain drain was the result of better living conditions in the United States, better access to funding for research, and a dynamic technological base that offered jobs and opportunities. Other countries now compete vigorously for scientists and engineers—China offers awards to technologists who return, Singapore has scholarship programs to recruit foreign science and engineering students, Britain even placed billboards in some countries with the message, "Can't get a visa to study in the United States?—Come to the UK." The flow of talent that once came automatically to the United States and stayed here could eventually dry up.

The effect on security is not as simple as having to rely on foreign workers or even foreign companies, rather than American workers or companies. Broad distinctions between "foreign" and "American" increasingly obscure the multinational nature of science and business. More important issues involve the ability of Americans to participate in and understand the products and networks on which they will depend. We do not want a situation where, as one federal official put it, "We know how to drive the car, but we've forgotten how to build it."

This shortfall will damage national security. Leadership in advanced technology derived from scientific research is critical to expanding U.S. military capabilities and to maintaining an advantage over potential opponents. Funding for basic research determines the scope and pace of discovery and the size of the technological workforce from which the United States can draw for its most important software security needs. But, it can be hard to win political support for these investments. Some see them as corporate welfare, and others are deterred by the fact that the benefits may not appear for years, lessening the political returns of voting for increased funding.

Policies that reinforce the innovative capabilities of the United States can mitigate the risks of globalization, including risks from foreign-produced software (or other IT products). Exactly the opposite is true. If the United States does not strengthen its innovative capabilities, its position will continue to erode, no matter how much effort it puts into increasing software assurance or building new procedures for evaluation. Less innovation in the United States means that more products will come from unknown sources and fewer Americans will be able to understand those products. Any strategy to improve security must include policies to strengthen innovation.

Recommendations

This section lays out ideas for addressing the central problem: increasing trust in software products used for national security and critical infrastructure applications. Any response should place the problem of foreign involvement in software in the context of cyber security and software assurance. Progress in addressing these issues will mitigate the risk of malicious code insertions involving malicious software. While market forces will continue to move to create

trustworthy software and networks, the process is slow and complicated. Our question is how to accelerate the move to software that is more trustworthy and quickly increase assurance for software used in critical functions.

A government response to the problem of malicious code insertions is more likely to succeed if it works with the private sector to identify and strengthen best practices rather than trying to develop a prescriptive approach. The norm in most high-tech industries is to use some sort of evaluation process, usually based on a standard, to achieve trust. A standard is simply a description of some threshold the product must pass in order to be deemed trustworthy (trustworthy in performance, reliability, or security). A product that is evaluated—either by the producer itself or by an outside party—to meet an industry-agreed standard has a known degree of reliability.

Underpinning the standards process is a governance structure that allows standards to be created, modified, and enforced. The system is usually voluntary—no one is "forced" to follow a standard. Instead, a combination of customer demand and enforcement through civil litigation produces compliance—if a nonstandard product does less well in the market, there is a powerful incentive for producers to follow the standard, and inaccurate assertions that a product meets a standard create liability and can incur civil penalties. In some industry sectors—aviation, computers, or automobiles—the process is well advanced. In others, it is still being developed.

Companies came to this approach as products became more complex and as a single manufacturer did not make every component or part that went into the final product. Aircraft are one of the best examples of this. A large aircraft will have thousands of parts produced by hundreds of suppliers. These suppliers are scattered across the world, as all large aircraft projects depend on a global supply chain. Aircraft manufacturers cannot afford to say, "I can only trust parts (or code) that I make myself." A supplier may have to have its production processes evaluated before contributing, and its products may have to meet certain performance standards before they will be accepted and used. Automobiles, computers, and other complex, high-tech products follow a similar model, which can be described as evaluating production processes for parts and components before they are integrated into the final product. These sectors have developed industry-led processes to manage distributed production and global supply chains for many years and are a source of precedent for the software industry.

The basis for any effort to reduce the risks posed by malicious code should include a reliance on market forces, the use of standards, and additional regulation. The preferred approach would combine greater transparency into production, identification, and sharing of company best practices; improvements in standards; and a degree of government oversight for software used in sensitive applications. Recommendations 1 through 6 below lay out some of the elements of this best-practices-based approach. Recommendations 7 and 8 emphasize that in addition to this tactical response, we also need a broader strategic vision of how to improve security.

1. Assess the risk (and share the assessment).

The first task for improving security is informational. Knowledge increases trust. To increase trust for software, software producers should share information on their practices and procedures for writing code, and the federal government should provide information on potential risks to producers. For software used in more sensitive applications (such as weapons systems), information about the provenance of the software (how much is legacy code, how much comes from external sources) should also be collected and shared with the federal customer.

Information sharing now is done on an ad hoc basis and suffers from a lack of formal processes and from releasability and classification problems. Finding the best vehicle for information sharing is a separate problem, discussed in the sixth recommendation.

Inserting malicious code into software during the production process (whether overseas or in the United States) is only one of several attack options available to opponents. Surreptitious insertion of malicious code is, in essence, a particular kind of insider threat. The first line of defense against this must be strong practices and procedures within companies. However, knowledge about foreign organizations interested in this sort of insider attack is not something that most companies have much access to—it lies outside of most companies' operations. Assessment will rely on traditional counterintelligence techniques to identify and understand operations by potential opponents and could take the form of sharing threat profiles of likely attack or attacks that pose the greatest concern to government services or critical infrastructure.

Responsibility for the software assurance problem falls across several agencies, but the responsibility for collecting information about which opponents might be considering malicious code insertions and the form these attacks might take should be assigned to the intelligence community. The designated agency should then share this information with federal customers and with appropriately cleared representatives of software companies and critical infrastructure operators. Government and industry can develop formal processes to improve the exchange of information about threats and vulnerabilities to inform and coordinate their risk assessments.

2. Focus on assurance, not location.

In the past, it was safe to assume that technology produced in the United States by an American firm was unlikely to contain intentional vulnerabilities. This assumption no longer holds. Even if the technology is manufactured in the United States, the global nature of business means that this alone does not guarantee trustworthiness. A U.S. company is likely to have employees from a broad range of countries. Foreign intelligence agencies could take advantage of the increasing internationalization of business to insert or recruit insiders, including U.S. citizens, with access to software production in the United States.

Market pressure has itself led to an increased emphasis on security within the software industry. Security was not a primary concern in the early days of the commercial Internet. Since 2000, it has become a central element in the processes many companies use to create software. A strategy that takes advantage of the best procedures adopted by leading software manufacturers to make their products more secure has a better chance of succeeding than a strategy that attempts to correlate security and location.

3. Avoid one-size-fits-all solutions.

The government already has processes for producing software with high assurance levels for very sensitive applications, such as command-and-control or intelligence. This software is written by cleared personnel working in secure facilities and following strict guidelines. This provides more secure software, but these products are more expensive. Pushing this set of secure processes down to apply to less sensitive functions would increase costs and slow development. For most of its software requirements, the United States cannot go back to an "arsenal" approach where everything is produced in-house.

Two major problems prevent the United States from scaling this high-security approach across government as a solution. First, software would become much more expensive, driving the cost of many programs to prohibitive levels. Second, government networks that connect to the Internet may, in the absence of other security policies, still be insecure, even if they use high-assurance code. The government cannot afford (and industry will not spend) the funds to require that all code used should meet strict security and assurance requirements. The solution requires deciding how far to extend these highly secure processes and what alternative processes can provide adequate assurance. This requires a differentiation between critical and noncritical applications.

Assurance can be based on different levels of rigor for review and certification. Existing levels include the Evaluation Assurance Levels (EAL) used in the Common Criteria, or the levels of maturity for software development established by Carnegie Mellon's Software Engineering Institute. They offer useful precedents on which to build. Critical systems requiring the highest levels of trustworthiness will necessitate very costly development processes. As the level of required trustworthiness declines, greater reliance can be placed on commercial software, development tools, and operating systems. Building on existing programs, an effective strategy will map software assurance levels and requirements to the sensitivity of the function and networks they support. Requirements could be progressively scaled from the most sensitive to the routine.

4. Refocus and reform existing certification processes for software products.

There are already several certification processes for software products, such as the Common Criteria. Currently, a company submits an existing product for

certification. Certification can include testing and a review of the processes used by the company in developing the software. The United States can lead an effort to streamline these certification processes, reduce their cost, and buttress them with best practices and software assurance tools.

The Common Criteria process is currently the primary vehicle for certifying that software meet standards for assurance. Industry views on the usefulness of the Common Criteria process are mixed, but the Common Criteria provide an agreed international framework for software assurance and for certification of products. Finding ways to make the Common Criteria review more effective, faster, and less expensive would form an important part of building software assurance.

The certification process would also benefit if it became more flexible and could adjust more rapidly to changes in software development processes, in programming, and in modes of attack. A standard that is adequate now for software assurance may no longer be adequate a year to two years later. The problem is the lack of a process by which standards can be revised in a timely fashion to fit developments and practices for programming and the ability to resist new kinds of attack.

5. Identify commercial-sector best practices and software assurance tools and expand their use.

Many software companies have already begun to put in place extensive assurance procedures as part of their production processes. The processes include a sequence of internal reviews for performance and security, internal testing, external testing and red teaming, and the use of software-review tools (some commercial, others proprietary and developed by the software company itself) to find vulnerabilities or errors. These practices offer the building blocks for an approach that is most likely to succeed in reducing the risk of distributed production. Extending some generalized set of these best practices from the IT companies to other software providers in the defense community would improve software assurance and security overall and reduce any risk from malicious code.

Some of the procedures developed by the financial sector (for offshoring software and IT services) could provide useful precedents for government agencies in developing schema for greater assurance. These procedures are designed to let financial institutions accurately assess risk and develop mitigating strategies by understanding the processes and controls that offshore producers use to address risks. Strategic Development Partnerships (where companies and government work together to develop software) are another useful precedent.

As part of this effort, the government could provide incentives for companies to build and share better software assurance tools. Automated tools are not the most important measures for mitigating risk. This does not mean they should not be used, but a risk-mitigation effort cannot rely on software tools alone. These tools are special software programs that review the code and test its functionality. These can catch many problems but will not find all problems. There are

commercial software assurance tools. These commercial tools continue to improve. Some companies also develop their own proprietary software review tools that are optimized for their products. They share some (but usually not all) of these proprietary tools with their partners or subcontractors.

Having one program check another program is not a panacea. The current generation of software tools works only as part of a larger assurance effort. As programs continue to grow in size, however, investment in R&D for better tools will become more important for preliminary checks of the millions of lines of code that will be found in many software products. One way to achieve this would be to increase funding for the Cyber Security Research and Development Act of 2003, which has never been fully funded. R&D of better assurance tools, along with other research-driven improvements in information assurance, is crucial for mitigating risk.

A set of best practices or a "due diligence" checklist for software developed by the private sector (with government participation) could reinforce certification processes and help provide assurance that code was trustworthy. An emphasis on best practices is the best way to reduce the risk of foreign interference (or other problems) in software. This set of processes would be less onerous than the high-end, very secure development requirements used for the most sensitive functions but sufficient to reduce risks for commercial software. Due diligence could entail ensuring that developers and coders are trained in writing secure code; taking steps to reduce the potential for an "insider threat" among company employees or contractors, ensuring that knowledge management processes and programs are in place to track additions and modifications to software as it is written; threat modeling to help guide coding and testing; a formal review of a software product by an independent security team, including the use of software assurance tools after a program is completed; and penetration testing conducted by external contractors.

6. Create a governance structure (or structures) for assurance.

Companies may be taking extensive steps to improve software assurance, but if these steps are unknown or unmeasured, they cannot increase trust. Finding ways to overcome this perception is a crucial step for addressing the central problem of increasing trust in software products used for national security and critical infrastructure applications. This is essentially a governance problem. Traditional approaches to governance—command-and-control or heavy regulation—do not work as well as they once did or may increase assurance, but at an unacceptable cost. An alternative solution is to create public-private partnerships to improve assurance. Whether this structure is formal or informal (and there are a number of existing groups that could be adapted to this purpose), the objective would be to identify and share the best practices developed by software companies, to shape requirements and procedures for improved software assurance, and to adjust these requirements and procedures in a timely fashion.

What is needed to remedy this can be described by the term "governance." Governance systems provide the answer to three questions: Who makes the rules? What is the rule-making process? How are the rules enforced? Part of the solution to the question of software assurance and security is to devise an adequate governance structure. An appropriate governance structure can create trust for software no matter where it is made. Efforts by private-sector groups in setting practices and standards for privacy, for smart-card use, and for the financial sector offer useful precedents.

The task of this governance structure will be to let government and the private sector jointly define secure development practices—best practices and due diligence requirements for software production. A well-designed governance structure would be more responsive in making necessary changes, as the practices and requirements for software assurance are not static. One approach would leave this entirely to the private sector—but government agencies are unlikely to accept this as adequate to meet their security concerns. Another approach would be for government to devise standards and then require industry to meet them. This approach is likely to be both inflexible and inadequate.

Both approaches have strengths. An industry-led process will be more flexible and able to react more quickly to changes in technology, threats, or business practices. It will have access to a larger pool of technical talent with greater expertise. A government approach is more likely to gain greater compliance. One alternative, however, is to blend industry and government processes into a new form of governance suited to the evolving modes of production in the software industry. This would be a partnership for software assurance involving stakeholders from federal agencies and the private sector. The group would identify policies and practices for software assurance to facilitate trust and evaluation. Any new effort should build on and complement existing mechanisms. While membership would be voluntary, products that meet the criteria established by the group could be given preference in procurements.

The United States now has an opportunity to devise new processes for making progress toward achieving the goal of greater assurance. One key to these new processes is to build on and expand what the private sector already does. Software producers realize the importance their customers place on assurance and security and have adjusted their internal procedures to meet this market demand. While there is much commonality and overlap in what companies do, each company approaches the issues of assurance and security somewhat differently. From these differences, we can extract best practices and requirements that will address, as part of a larger solution set, the risks posed by foreign involvement in software production.

This is not a new approach. There are already many different partnerships, committees, and groups dedicated to improving trusted software and information security. This multitude of groups already provides real benefits for security, but they could be made more effective. One approach is to create a different kind of stakeholders group. The stakeholders would need to include both industry and government agencies. Government would not play its traditional role of rule

issuer, but instead set agendas and ensure compliance (perhaps by making the use of best practices a requirement for some acquisitions). A stakeholders group could develop best practices and requirements for trusted software development processes and map these best practices to assurance levels set by the government. This process could include procedures for evaluation, accreditation, and compliance—a self assessment in some cases and external review in others—that would demonstrate that these best practices were actually being used.

One crucial element for better governance is a more concentrated federal policy process to provide unity of effort. There may be as many as a dozen different federal groups pursuing software assurance and security. Consolidation and harmonization would help. The best way to achieve this unity of effort is for the new Policy Coordinating Committee for Cyber Security to provide a strategy and architecture for software assurance, and for one of the existing bodies, such as the Committee on National Security Systems, to provide oversight and coordination with an industry-led effort.

7. Accelerate improvements in information assurance.

Even if there were no foreign participation in IT production, networks would still be insecure. They involve thousands of different devices, some running older legacy code, others running unpatched programs, and all facing the possibility that they are vulnerable because of a configuration error found in a separate network to which they connect but do not control. In this environment, knowing who has accessed information, and whether they have changed it, copied it, or transferred it offers a different way to improve security. Greater attention to accountability and transparency in information use—monitoring and safeguarding data at rest—can help manage risk. Emerging technologies for self-monitoring networks, data protection, and better authentication and authorization will counterbalance network and software vulnerabilities, because they will allow networks to control who can access information and what they can do with it.

Information assurance and secure networks are the ultimate goals of increased attention to software reliability. Any effective defense will involve multiple layers that address the different aspects and challenges of the information security problem. This will not be put in place quickly or all at once, but improving software assurance will only improve information security if it is part of a larger security strategy.

8. Promote leadership in information technology innovation.

Globalization, distributed production, and strong foreign competitors are unavoidable, but the United States can take steps to keep itself at the forefront of technology. Technological innovation has powerful economic returns, but there are returns for security as well. Innovation makes life more difficult for opponents. All of an opponent's work to "rig" a technology is lost if a new technology appears and supplants it. Innovation can improve assurance processes, create better tools, and increase overall network and information security.

Measures that improve the climate for innovation in the United States (such as increased funding for IT-related R&D) also help build a skilled domestic workforce, so that the United States does not find itself relegated to low-end functions or working off another nation's designs.

The last year has seen a number of political statements and reaffirmations of the importance of strengthening U.S. innovative capabilities, beginning with the American Competitiveness Initiative announced in the State of the Union address, but these reaffirmations have so far produced little additional money or changes in regulation. Any fears Americans have about dependence on foreign technology, whether justified or not, will only grow if the United States does not work harder to expand the sources of innovation.

One specific step would be for Congress to increase funding for the Cyber Security Research and Development Act. Signed into law a little more than five years ago, this program could provide the resources for work on new security tools, architectures, and coding procedures that could improve information security across the board.

Conclusion

It should be no surprise that one result of immense economic and technological change is that old assumptions about security and the policies based on those assumptions do not work as well as they did in the past. The process of adjusting the policies to the new global environment is a major challenge for all governments. Each country must in some way respond to a world where the lines between government or the commercial sector and domestic production or foreign production are blurred.

There is doubt, of course, over the value and scope of global economic integration. The dreaded words "outsourcing" and "offshoring" can provoke fierce political debate. But this debate is neither new nor accurate. Just as English workers who smashed factory machines in an effort to block industrialization were unknowingly working against their own interests, those who object to an integrated global market fail to realize the cost and damage that any steps back to separate, less connected economies would create. For the United States in particular, the cost could be very high and could even include the loss of economic and technological leadership.

To maintain national security in the new economic environment, we must recognize that risk and threats will take new forms, and in devising measures to mitigate that risk, we must choose those that interfere the least with the global IT market, as this market drives companies to innovate and lower costs. To devise these measures, we must approach the question of how to build trust from a different perspective. We cannot go back to a national supply base or to working only with code written in the most secure environments. There is no magic technological bullet that will solve the problem (at least not yet and not for the foreseeable future). An answer will require better processes for assurance, greater

transparency, and a reorientation of how we think about security and will be able to shape a holistic strategy for information assurance.

Asking about foreign influence on software production is in some ways the wrong question. Location is less important for determining trust than is the strength of the processes used to write and test software. There will be greater improvement in security if a government response to the problem of malicious code insertion is to work with the private sector to identify and strengthen best practices rather than to try to develop prescriptive approaches. The software production process is becoming more mature, and companies have responded to their customers' demand for more secure software by creating processes for writing and reviewing code that increase assurance.

New policies should place the problem of foreign involvement in software in the context of cyber security and software assurance. A malicious code insertion is possible, although not the most probable mode of attack on information systems and networks. Developing a new model for trust that is not based on a reliance on national industries will require new ways of thinking and new organizations. The old model for trusted technology rested on the assumption that technology produced by a national firm was unlikely to contain intentional vulnerabilities. In an interconnected global economy, it is no longer safe to make this assumption.

About the Author

James A. Lewis is a senior fellow and director of the CSIS Technology and Public Policy Program. Lewis, a former member of the U.S. Foreign Service and the Senior Executive Service, worked on foreign policy, national security, and technology-related issues at the Departments of State and Commerce. Since coming to CSIS, he has authored numerous publications, including *China as a Military Space Competitor* (2005), *Globalization and National Security* (2004), *Spectrum Management for the 21st Century* (2003), *Perils and Prospects for Internet Self-Regulation* (2002), *Assessing the Risk of Cyber Terrorism, Cyber War, and Other Cyber Threats* (2002), *Strengthening Law Enforcement Capabilities for Counterterrorism* (2001), and *Preserving America's Strength in Satellite Technology* (2001). His current research involves innovation, military space, and the global information technology industry. He received his Ph.D. from the University of Chicago in 1984.