Software Security in a Flat World

Mano Paul, CSSLP, CISSP, AMBCI, MCAD, MCSD, Network+, ECSA

Introduction

The developer anthem entitled "My epitaph" that was specially commissioned by Saltmarch Media and performed by India's leading rock band Motherjane during the Great Indian Developers Summit of 2010 highlights how the shrinking (flattening) of the world has impacted the life of developer. This "geek god" is considered to be "a human architect in the digital crucible…one of the zillion unsung heroes" putting in place "the ones and zeroes." But he's unknown, "behind the a screen". The song ends with these words: "You see the world is so much smaller these days. How could something so small take away my face?"

Personal computers, Internet, workflow software, outsourcing, offshoring, collaboration communities, supply chain streamlining, insourcing, search engines, and ubiquitous computing via digitization and virtualization – the ten flatteners that are discussed in the renowned book The World is Flat, by Thomas Friedman – all seem to have one thing in common: they are all sourced in technology. The prevalence of technology is, in fact, what makes this flattening possible.

Powering the technologies is software. Unfortunately, however, as is evident from bug tracking databases and full disclosure lists, it cannot be assumed that the software that influences the various flattening trends is necessarily secure. In most cases, software is merely sprinkled with some security protection mechanisms, rather than being developed with a software security assurance mindset.

This whitepaper covers the flattening forces at work that impact people, processes, and technologies, with focus given to software development in outsourced and offshored engagements. We'll look at software security concerns and the solutions one needs to consider in order to be secure in our rapidly flattening world.

Software engineering … From its Genesis to Now

Software is ubiquitous and a crucial and essential component of our society. The engineering of it has evolved dramatically. What began as a time-intensive and cumbersome methodology requiring "machine-rooms" and "punched cards" for implementing functional logic is now an efficient and prominent discipline, involving model-driven and agile development methodologies.

In 2006, Money magazine and Salary.com rated software engineering as the most attractive career in America in terms of growth, pay, stress levels, environment, flexibility in working hours, creativity, and ease of entry and advancement. This held true for not just the United States, but for almost any geographical region with a highly-educated workforce with analytical and computer programming skills. Five years later, the results of a survey conducted by CareerCast show that software engineering continues to be ranked #1 on the Ten Best Jobs of 2011.

But it must be recognized that without due consideration given to the security aspects, the appeal of software engineering today can be diminished quickly in this technology-empowered flat world.

Software security in a flat world

Software outsourcing splits services and development activities into components that are subcontracted and performed in the most efficient and cost-effective way, while offshoring moves those services and development activities to a foreign land, often to gain the benefits of lower labor costs and tax incentives, besides access to intellectual capital. Flattening as a result of outsourcing and offshoring has had an impact on the way business is conducted, how software development happens, and on security.

Software security assurance must now be viewed from a new and different perspective. It requires a holistic viewpoint. Security programs must be more than just technological controls. They must incorporate the people and process aspects of flattening as the flattening impacts the resiliency of software. Only when security is holistically considered in a given software development project can one be assured that the software developed is intrinsically and truly secure in our flat world.

Dude, where is my perimeter?

Offshoring has warranted the need to move certain processes, and in some cases even the data, outside the boundaries of the company. This has made the perimeter that once defined an organization's boundary thin, and in some cases non-existent, resulting in serious security concerns. The security professional's charter, which was to protect the organization from intruders that were outside the organization's perimeter, has changed to now having to defend organizational assets without the benefit of a definitive boundary.

Dude, where are my personnel?

Not only is the perimeter disappearing, but the location of where the work is performed has changed as well, thus producing important security concerns that require attention.

Before the year 2000, India experienced what came to be known as a "brain drain" as world leaders, primarily the United States and the United Kingdom, sought software engineers to help address the Y2K problem. This phenomenon prompted software engineers to physically

relocate to where the employers were, often on the other side of the globe. Now, just a little over a decade later, the prevalence of offshoring has resulted in a reversal in the brain drain phenomenon. Instead of workers moving to where the work is, the work has moved to where the workers are.

Software developed outside the purview of one's control warrants inspection. Inspection of code for implanted insider issues, such as logic bombs and Trojan horses, is critical for the purchaser of the software. The institution of standardized systems development lifecycle (SDLC) processes and organizational capability maturity models (CMM) with software assurance ingrained in the SDLC, are necessary for the publisher of the software.

"Software developed outside the purview of one's control warrants inspection."

The move toward standardization

A decade ago, every organization largely had software development processes in place that were proprietary, and interoperability between heterogeneous systems was more or less impossible. Now, in a flat world that fosters global commerce, organizations have created efficiencies with standardized software development processes. What was often an ad hoc process is now of a pattern, repeatable, managed, and defined. But having structured software development processes does not necessarily mean that the developed software is intrinsically secure, unless security activities and tasks are made a part of the overall software development process.

From requirements to retirement

From requirements to release and to eventual retirement of systems and data, security activities need to be interjected into the SDLC. Governance and policy decomposition activities must be undertaken to glean out security requirements in the requirements phase of the SDLC. Data classification, threat modeling, use/misuse case modeling, subject-object modeling, and role matrix determination are bare minimum, must-have, security activities during the design phase of the SDLC. Implementing software assurance controls in code and code reviews (inspections) are not to be ignored during the development phase. Testing for the presence and effectiveness of the software assurance controls must be undertaken in the testing phase of the SDLC. Regression testing, simulation testing, and user acceptance testing must factor in as well, for both the functionality aspects of the software as well as the security aspects. Secure installation and secure boot-strapping are essential components of secure software deployment activities. Software Quality Assurance (SQA), continuous monitoring, and post-deployment verification and validation activities such as penetration testing and vulnerabilities assessments are crucial. Eventually, the software system and associated data must be securely disposed of, or archived offline for later use.

Ideally, it would be most beneficial for the software development team to have dedicated security personnel tied to a software development project, instead of leveraging a shared service resource for security. Practically, however, often due to resource constraints, such allocation may not be feasible. In such situations, leveraging automation of security processes as well as the use of tools may need to be considered.

More than meets the eye

When operating on an outsourced software development project, it is important to recognize that, when it comes to implementing the program successfully, there is more than meets the eye. Culture, as it pertains to communication, position and pay, and motivational factors, is an important consideration. Taking into account the cultural context when implementing any given security program is vital for the success of that program.

Communication

It is not always clear-cut when it comes to the understanding of requirements, particularly security requirements. Anyone who has engaged in an outsourced software development project would agree that there are communication challenges to face for both the outsourcer of the software (producer/service provider) and the outsourcee (purchaser/client). This might be due in part to unsolidified requirements but, more often than not, it's because of a lack of understanding of the respective cultures.

In addition to verbal and written expressions of the project requirements, non-verbal expressions need to be paid attention to, as well. A simple nod in one culture could mean one thing in one culture and something entirely different in another. "Saving face" (not losing one's honor) is very important in certain Asian cultures, for instance. Answers such as "I don't know" or negative or correcting comments are considered unacceptable in some cultures, even though the nature of software security is such that answers are not always readily available. How, therefore, a question or concern is communicated becomes crucial. In most cases, partnering together to find the correct and secure solution(s) is the best approach.

Position and Pay

It is also important to recognize the position and pay of developers in the outsourcer's organization. Hierarchical organization culture can hamper the creativity of a software engineer in a lower position, potentially causing the process to suffer. Not only does the position a developer might have within an organization impact the creation and implementation of security solutions, but their pay does as well. The dark side can be alluring; cybercrime not only pays, it pays very well. Underpaid software engineers can be more easily swayed to participate in fraudulent and nefarious activities and are often the targets of more organized cyber criminals.

Software engineers add a lot of value to the business, and they must feel as though their importance is recognized. A developer can be security's best friend or its most dangerous foe. Since developers have access to source code, they can be powerful friends in implementing controls in code. But only when they are educated and trained to implement requirements without compromising the need to protect the confidentiality, integrity, and availability of data and systems. For at the same time, they can also be very dangerous foes, as they can just as easily implement malware software such as logic bombs and Trojan horses.

"A developer can be security's best friend or its most dangerous foe."

Motivational factors

In some cultures, positive reinforcement programs are effective while in others, negative reinforcement programs prove handy when it comes to implementing software security programs.

Some examples of positive reinforcement include recognizing teams that produce code with no known security vulnerabilities ("Zero-Bug Code") and/or providing an incentive to the individual who writes the most hack-resilient code by placing him or her in a "Hall of Fame", or some such honor. Leveraging the competitive interests of people by pitting teams or individuals against each other in the interest of improving the state of security in software, proves very useful in implementing such positive security programs. It is also evident, however, that in some cultures, negative reinforcement is relatively effective, though not often welcomed or advised. Some examples of negative reinforcement include penalizing the teams with the maximum number of security vulnerabilities by, say, withholding their bonus for that season ("Zero-Bonus Code"), and/or placing the individual with the most insecure code in a "Hall of Shame".

Ancillary considerations

Thomas Friedman summarizes that, "Technology cannot protect us; we must harness that technology and decide how it will be used." The same idea is aptly expanded by Peiter C. Zatko, program manager in the Defense Advanced Research Projects Agency (DARPA) Strategic Technology Office (STO), who, in his keynote address, "Analytic Framework for Cyber Security" during the Shmoocon 2011 conference, said "Technology is not the only culprit; nor the only answer!"

There are a few other things that need to be taken into account to holistically secure the software being developed. These include legalities in software development, and fraud control.

Legalities in software development

Legal and regulatory protection mechanisms and instruments such as contracts, service level agreements (SLAs), copyrights, non-disclosure agreements (NDAs), and non-compete agreements become important and necessary when dealing with software development in a flat world. When opportunities for software developers are many and attrition is high, NDAs to control the flow of competitive information and trade secrets are a must and need to be enforced, so long as the enforcement does not violate any privacy or regulatory requirements. Contracts development must not be just a pen-and-paper exercise, but an exercise that produces an in-depth analysis from a security perspective, involving legal teams, security personnel, and the development organization. This is particularly applicable when it comes to the correct "legalese" that needs to be specified in End User Licensing Agreements (EULAs) and disclaimers. Additionally, for published code that is signed, it must be ensured that the correct organizational certificates are used to sign the code, which the legal teams can help determine. It is also very important to verify the ability to enforce contractual obligations when dealing with jurisdiction across borders. In some cases, software escrowing may be necessary to protect both the licensee, assuring them of business continuity, and the licensor from any breach of intellectual property rights.

Fraud control

To augment legal protection measures in software development, administrative controls such as background checks of employees and collusion control efforts are critically important to minimize the incidence of fraud. The fraud that fleeced customers of Citibank to the tune of nearly 300,000 USD caught international press, and investigation revealed that it was orchestrated and conducted by the call-center employees of an outsourcer. This further accentuates the need for people-, process- and technology-based assurance controls, particularly as they apply to protection against data theft and misuse.

What Next?

The publication of software from the traditional original equipment manufacturer (OEC) license model is shifting toward a service-oriented model as business applications and software, infrastructure, and/or platforms are becoming virtualized. We now live in a service-oriented flat world that is also virtual. Computing by anybody, anywhere, is a reality today. Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), mere concepts just a decade ago, are very much the reality in today's computing environment.

The benefits of virtualization include leveraging physical resources, server consolidation, and reduced cost. From a software development perspective, production environments can be more easily simulated by way of virtualized development environments, which reduces the likelihood of software deployment issues. However, it must be recognized that in a virtualized world, security cannot be virtual. In addition to the inherent isolation between virtual machines by most virtualization technologies, additional sandboxing of applications operating in virtualized settings is necessary.

With ubiquitous computing made possible by digitization and virtualization technologies, personal and mobile computing devices are

"Security in a virtualized world, cannot be virtual"

on the rise within organizational networks. "SMiShing" – Short Message Service (SMS) phishing – is a growing social engineering technique that uses mobile devices such as personal digital assistants (PDAs) and cell phones to deliver the "bait" aimed at conducting disclosure attacks. Other threats such as Bluesnarfing and Bluejacking have been evident against the Bluetooth protocol. The Voice over Internet Protocol (VoIP) telephony implementations that are the backbone of communications in the majority of outsourced and offshored engagements is susceptible to vishing (voice over IP phishing) attacks. User education against electronic social engineering, node-to-node authentication of devices, and cryptographic protection of data in transit and at rest becomes mandatory in such computing environments.

The importance of training and education

Awareness, training, and education programs on software security provide a significant return on investment because development team members become not only aware of what needs to be done to produce hack-resilient software, but also become trained and skilled in implementing the necessary controls. And such software security training can be valuable to a developer who may feel that he or she is one in a zillion by providing a differentiating edge. In other words, as the Saltmarch Media developer anthem might have it, developers who are adequately trained in software security can be said to have "a face."

Cross-cultural training is also critical. Such training will help ensure that there are no ethnocentric overtones but, more importantly, will help ensure that the understanding, design, and implementation of security requirements do not collectively take a mere spectator role.

The Certified Secure Software Lifecycle (CSSLP®) credential offered by the International Information Systems Security Certification Consortium (ISC)[2] is recommended as a starting point to educate the development community with software assurance concepts and principles. It is a holistic credential that is focused on educating all stakeholders that are involved in a software development project.

Conclusion

While the flattening trend has brought with it several advantages, it has also brought with it a fair share of security concerns. Outsourcing and offshoring of software development projects throw a new light on how one needs to consider the security of the software that is produced. The cost savings that is realized from outsourcing and offshoring must be balanced against the estimated cost of a security breach, and only then can appropriate decisions be made.

In a flat world, security can become a differentiator. The company that is recognized as having structured people-, process- and technology-centric "secure" software development methodologies will stand out against companies that don't. Developers who feel they don't have "a face" can differentiate themselves when they are trained to develop software with a security mindset, thereby addressing the identity crisis that they may otherwise feel.

For the discipline of software engineering to retain its stature as the most attractive career, it must be coupled with software assurance efforts. Security processes and technology must be an integral part of the software development process, from the initial requirements gathering phase to the final retirement phase. Technological controls must be implemented in conjunction with other process- and people-centric controls. User education on security and administrative controls such as background checks are very important in reducing the incidence of fraud and social engineering attacks.

Not only should the client organization's culture be taken into account, but one must pay attention to the personal and local cultural context and motivational factors of developers when software is developed in an offshore location. Cross-cultural training is pivotal in ensuring the assurance capabilities of software – its reliability, resiliency, and recoverability.

Not taking into account software security in a holistic manner, and failing to look into the software assurance controls and capabilities of the software being produced, are sure ways to leave your organization "flat on its face" in this flat world.

About (ISC)²®

(ISC)² is the largest not-for-profit membership body of certified information security professionals worldwide, with nearly 75,000 members in more than 135 countries. Globally recognized as the Gold Standard, (ISC)² issues the Certified Information Systems Security Professional (CISSP®) and related concentrations, as well as the Certified Secure Software Lifecycle Professional (CSSLP®), Certified Authorization Professional (CAP®), and Systems Security Certified Practitioner (SSCP®) credentials to qualifying candidates. (ISC)²'s certifications are among the first information technology credentials to meet the stringent requirements of ANSI/ISO/IEC Standard 17024, a global benchmark for assessing and certifying personnel. (ISC)² also offers education programs and services based on its CBK®, a compendium of information security topics. More information is available at www.isc2.org.

About the Author

Mano Paul, CSSLP, CISSP, AMBCI, MCAD, MCSD, Network+, ECSA is CEO and President of Express Certifications and SecuRisk Solutions, companies specializing in professional training, certification, security products and security consulting. His security experience includes designing and developing software security programs from Compliance-to-Coding, application security risk management, security strategy and management, and conducting security awareness sessions, training, and other educational activities. He is the author of the Official (ISC)² Guide to the CSSLP, and is a contributing author for the Information Security Management Handbook. He writes periodically for Certification, Software Development and Security magazines and has contributed to several security topics for the Microsoft Solutions Developer Network. He has been featured in various domestic and international security conferences and is an invited speaker and panelist in the CSI (Computer Security Institute), Catalyst (Burton Group), TRISC (Texas Regional Infrastructure Security Conference), SC World Congress, and the OWASP (Open Web Application Security Project) application security conferences. He can be reached at mano.paul@expresscertifications.com or mano.paul@securisksolutions.com.

References

Ten best jobs of 2011. CareerCast survey. February 15, 2011. http://www.careercast.com/jobs-rated/10-best-jobs-2011

Developer Anthem, By Motherjane. Great Indian Developer Summit 2010.

Analytic Framework for Cyber Security. Peiter "Mudge" Zatko. Shmoocon 2011

Best jobs in America (Top 10 best jobs). Money Magazine and Salary.com http://money.cnn.com/magazines/moneymag/bestjobs/2006/

BBC News | Millenium Bug | India. Brain Drain. http://news.bbc.co.uk/hi/english/static/millennium_bug/countries/india.stm

Beware the Reverse Brain Drain to India and China. By Vivek Wadhwa. Oct 17, 2009. Tech Crunch.

Saving Face in India. By Dat Nguyen. http://culture-4-travel.com

Cybergang infects all ATMs in Russian city. December 03, 2010. http://www.net-security.org/malware_news.php?id=1555

Indian call center workers charged with Citibank fraud.

Secure development (for a secure planet). Eoin Keary. OWASP Belgium 2009.

Going East: Bargain-hunting companies look off shore for programming, zbut at what cost? Erik Sherman.

DEVELOPER ANTHEM

I've got whole worlds staring at screens
And yet no one ever senses me in between.
I could be any of the ones or the zeros
One of its zillion unsung heroes
I'm told I'm a geek god in a glass temple
Worshipped. Overpaid. Replaceable.
A human architect in this digital crucible,
Neo, Invincible, One, Invisible.
I exist, God-like in a desk job mode
And on the 7th day I rest
I can be benched, right sized or
dropped on the road
Never lived for, though I too will live
forever with the index
So my friend, if you, like myself,
are seeking me now
I pray this song finds you somehow.
Coz my world is so much smaller these days
So small, precious little is face to face
And if I've earned one epitaph no one grudges me,
It's for being human, behind the screen
For putting the ones and zeros in place
And most of myself in this digital embrace
You see the world is so much smaller these days
How could something so small take away my face?