The Need for Secure Software

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

Introduction

Since 2005, when the Privacy Rights Clearinghouse started collecting and publishing the Chronology of Data Breaches[i], not one year has gone by without noteworthy data breaches. To date, 226 million records have been reported to have been disclosed or breached.

All of these instances have things in common: large numbers of records disclosed, consumer victims, and colossal punitive damages and fines levied on the organizations. Further analysis of the cause of these and other data breaches invariably indicate one of, a combination of, or all of, the following:

- Insufficient protection of data during transit or at rest
- Insecure software designed, developed, and deployed (in built or third-party)
- Improper or inadequate configuration of software security controls
- Wireless and physical security breaches (thefts) leading to data compromise
- Lack of layered security defensive measures at the perimeter, hosts, and applications

Hackers Do Not Discriminate

While the provided highlighted breaches might suggest that the main victims of data breaches are large corporations, it's important to understand that hackers do not select their targets based on the size of the organization. Nor are data breaches unique to any one industry. Government, education, healthcare, banking, retail, wholesale, insurance, and media sectors alike have experienced some kind of data breach.

Highlighted Breaches

2005 – ChoicePoint reports disclosure of 163,000 consumer records, resulting in identity thefts of the victims whose information was breached, and a US$10 million settlement fine.

2006 – Through one of AT&T's vendors, computer hackers access the account data and personal information of nearly 19,000 AT&T credit card holders. This is followed by phishing emails soliciting even more personal information.

2007 – TJ stores (TJX), including TJMaxx, Marshalls, Winners, HomeSense, AJWright, TKMaxx, and possibly Bob's Stores, report a breach which includes, as is estimated at this writing, the records of close to 100 million credit and debit card accounts. This is arguably the most publicly known data breach to have occurred in the U.S. to date, with a recovery cost estimated to be about US$216 million.

2007 – HM Revenue & Customs in the UK reports the loss of personal data of nearly 25 million people. Although this breach occurred due to the fact that compact discs containing sensitive information were lost, the more underlying and important issue is that the data stored on these CDs should have been, but was not, protected from disclosure. Even with a very conservative cost estimate, Gartner Research estimates the recovery costs to be about US$500 million[ii].

2008 – 4.2 million credit and debit card numbers are stolen during the credit card authorization transmission from the supermarket chain Hannaford Bros., resulting in 1,800 cases of fraud reported so far.

Changing Landscape of Security

The security landscape as we know it today is changing. It is no longer adequate for organizations to protect just their perimeter, with the intent of keeping the bad people out. In addition to perimeter security controls such as firewalls, host-based protection mechanisms such as anti-virus and host-based intrusion detection systems are necessary. Yet even protection of both perimeter and hosts are proving insufficient, as hackers and crackers have started to focus their attention on one of any organization's most critical assets – its data.

Increased incidences of data loss have led to greater governmental oversight and regulation. The California State Bill (SB1386) requiring organizations to notify any California resident of disclosure of their personal information to unauthorized individuals was the first law of its kind. This groundbreaking bill led to a multitude of data security and protection acts ratified by various other state governments requiring such protection for their residents as well. In the U.S., as of today, 38 out of the 50 states have enacted some sort of breach disclosure law, and a few others are currently petitioning.

In addition to being affected by an increasing number of such regulations, organizations are also being forced to implement and abide by a plethora of compliance initiatives:

- Sarbanes Oxley (SOX) is the U.S. mandate that requires adequate controls to be in place to protect sensitive data and assets for publicly-traded companies.
- The Gramm-Leach-Bliley Act (GLB Act) which includes provisions to protect consumers' financial information is mandatory for financial institutions.
- The Payment Card Industry Data Security Standard (PCI DSS) encompasses credit card transaction protection.
- The Health Insurance Portability and Accountability Act (HIPAA) requires protection of personal health information.
- The U.S. Federal Information Security Management Act (FISMA) has similar data security management requirements for United States federal government organizations.

Although many of these compliance requirements originated in the U.S. for U.S. organizations, most are applicable globally. In fact, some countries have imposed even more stringent compliance requirements than those of the U.S. The Basel II is a compliance requirement for the European financial services sector comparable to the SOX Act in the U.S. Japan's Financial Instruments and Exchange Law (J-Sox) and India's Information Technology Act covering cyber security are other noteworthy global examples of this changing landscape of security.

Today's business environment dictates that in order to maintain a competitive advantage, organizations need to be configured with a vanishing or opened perimeter. Sensitive customer and business data are now available to privileged third parties, including contractors, outsourcers, business partners, supply chain nodes, and other business network users/stakeholders. This reality of today's marketplace places a significant emphasis on the urgency and need to protect internal assets and data. One of the "10 Big Predictions" for 2008 by the IT Compliance Institute indicates that IT management can't rely on, and be content with, merely home-field security efforts[iii]. Security measures need to expand to provide control measures for today's "over-the-wall" data transfers.

Drivers for Software Security

The 2008 Global Information Security Workforce Study (GISWS), sponsored by (ISC)$^{2®}$ and conducted by Frost & Sullivan, noted an interesting shift from prior years: prevention of damage to an organization's reputation was now the highest priority for many of the respondents.

If the changing environment (increased data loss incidents, increased number of regulations and compliance requirements, and growing number of perimeter-less organizations) is not enough to mandate the need for software assurance, here are a few other drivers for secure software:

- Negative publicity
- Investigations and litigation
- Liability (corporate as well as personal)
- Erosion of brand and reputation
- Loss of public or customer confidence and trust

Customers, partners, shareholders and stakeholders who do business with your organization expect their information to be protected, and any violation of their trust, regardless of the cause, could be disastrous.

Data: The Next Frontier

Data is an organization's most critical asset. Data can be defined in many ways, from personally identifiable information (PII), to personal health information (PHI), to financial and intellectual property (IP). No matter how it's defined, it needs to be protected. The 2008 GISWS report noted that organizations are finding that significant costs result from data breaches. Estimates are from US$50 to $200 per record lost. And of course this doesn't take into account the intangible costs of reputation damage and loss of trust.

Software applications are the conduits and processors of data. A weakness in software is like a weakness in the lock of a safe. No matter how strong the iron walls of the safe are, its overall security is nonetheless compromised. By the same token, with an organization's data, relying merely on perimeter controls is insufficient. The famous candy slogan "Hard and Crunchy on the Outside, Soft and Chewy on the Inside" aptly describes most organizations today. They have heightened network perimeter controls in place (hard and crunchy) but, internally, their applications (software) and data are mostly left unprotected (soft and chewy).

Secure Software: A Necessary Solution

Software development has changed considerably from a decade or two ago. Computer systems were usually islands in and of themselves, with limited or no connectivity. Applications were generally self-contained units and were deployed after being tested for functionality on unconnected systems. At that time, software as a service did not exist except in the concept stage. Security was not deemed as critical since a breach primarily meant that the worst that could happen was that the attacker could attack only themselves in a contained environment.

But with the growth of the Internet there has been a paradigm shift in the way computer systems are networked and how applications are developed, resulting in a momentous effect on security. The high degree of computer connectivity present in today's society means that applications not designed to operate securely are susceptible to attack from both outsiders and insiders. Today's applications need to be secure and must be expected to operate in potentially hostile environments.

The following are the leading imperatives that are driving the need for software assurance in today's computer environment:

- Security - an afterthought in the Systems Development Life Cycle (SDLC)
- Need for a security mindset
- Attacker's advantage vs. defender's dilemma
- Insider threats (malicious or otherwise)

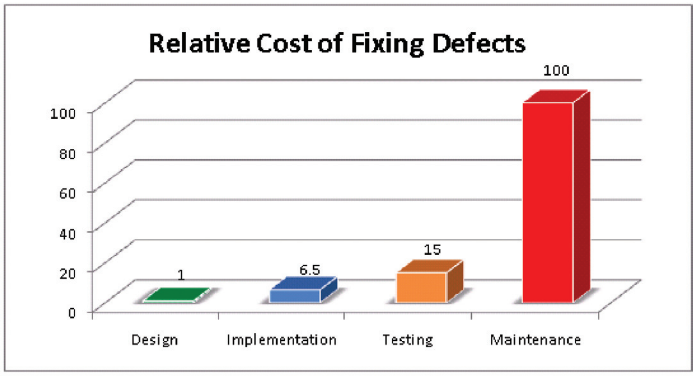
Security - An Afterthought in the SDLC

There are many SDLC models that are used by organizations to effectively develop information systems, including traditional linear waterfall, rapid application development (prototyping), and agile extreme programming models. The U.S. National Institute of Standards and Technology (NIST) cites that as a general practice, software development needs to include these five phases:

- Initiation
- Acquisition/Development
- Implementation/Assessment
- Operations/Maintenance
- Sunset (Disposition)

If your organization is directly involved in software development, either as a business or to meet internal needs, security controls should be incorporated from the initiation (concept/planning) phase of the project and validated all the way through the operations/maintenance phase of the project. Unfortunately, in most cases of development, security is an afterthought. Lack of time, limited personnel resources, and a pervasive lack of awareness of the value of security, are some of the reasons why security controls are not built in from the beginning. Security is all too often bolted on only afterwards, as a response to some threat or after a vulnerability has been exploited. But incorporating security early, and maintaining it throughout all the different phases of the SDLC, has been proven to result in less expensive and more effective security than adding it to an operational system. The too-prevalent mindset that building security in upfront is too expensive and time-consuming is a major challenge to overcome.

Studies show that the relative cost of fixing defects in production is 30 to 100 times more expensive. Figure 1 depicts the results of a study conducted by IBM Systems Sciences Institute on the relative cost of fixing defects.



Source: Implementing Software Inspections, IBM Systems Sciences Institute

Figure 1: Investing in security early can dramatically decrease the cost of fixing defects

If your organization depends on third-party vendors and commercially-off-the-shelf (COTS) software, it is essential that these third-party processes are scrutinized from a secure application development standpoint, and essential as well that contractual and insurance agreements are established to protect your customers, stakeholders, and your organization.

Need for a Security Mindset

Internationally renowned security technologist and author Bruce Schneier writes that teaching designers and developers a security mindset would go a long way toward making technological systems more secure[iv]. Schneier notes that the lack of a security mindset explains a lot of existing poor security. Designers are so busy concentrating on making systems work that they don't stop to see how they might fail or be made to fail.

If development teams had a security mindset, systems and applications would be more secure and organizations like ChoicePoint, AT&T, TJX Stores, HM Revenue & Customs, and Hannaford Bros. might not find themselves at the brunt end of lawsuits, absorbing huge fines and earning customer distrust. Privacy exposures wouldn't be the significant issue it currently is. Stolen devices would not have sensitive information in unprotected format. Confidentiality and ID theft and credit card fraud, usually requiring data compromise, would be reduced considerably. Through design, development, and deployment, security should be always at the forefront.

Attacker's Advantage vs. Defender's Dilemma

As soon as software is deployed and enters the operational/maintenance phase, it immediately becomes a potential target for attacks. Ideally, it needs to be able to defend itself from the moment of implementation. A malicious attacker has many advantages, leaving the defender in a dilemma. One of the primary advantages for the attacker is that the attacker needs to exploit only one weakness, while the defender has the responsibility to defend against all threats. Additionally, attackers have the luxury of attacking applications and systems however and whenever they like, while the defender needs to be on constant vigil. Furthermore, the attacker need not (and almost never does) play by the rules of engagement, while the defender is required to follow the rules[v].

Software security measures would shift the advantage to the defender, putting the attacker in the dilemma. With proper threat modeling of the attack surface, and security design and architecture, the defender becomes aware of the possible vulnerabilities that an attacker can exploit, and can mitigate the risks before they're discovered and exploited by the attacker. Proper auditing and alerting automation in the software aids the defender, and allows the defender to be on guard constantly. Knowledge of sentry features in the software would repel many attackers, and reverse the dynamic, putting the attacker in the position of having to play by the defender's rules.

Insider Threat

While firewalls provide excellent protection against attacks that originate from outside your organization's perimeter, they are ineffective if the attack generates from inside, from your employees - the very people you trust with privileged access to your organization's critical assets and data. The insider becomes an even more serious threat than the external hacker.

In fact, the (ISC)²® GISWS survey showed that 51% of those surveyed feel that internal employees, who have privileged access to internal sensitive information, are indeed the bigger threat.

A myriad of information breaches and data loss incidents have been attributed to privileged third parties who have not been in the forefront of managerial security oversight. Contractually, enforceable security controls pertaining to software assurance must be put in place along with internal policies, standards, and procedures, complementing the information security program to ensure that the risk of insecure software is either accepted, transferred (via insurance), or mitigated.

All of these call for data protection to be a necessity. Successful exploitation of insecure software can lead to data breach and information leakage (confidentiality exposure), modification or alteration of data (integrity exposure) and/or defacement, downtime, and denial of service (availability exposure), besides financial loss. Undetected and surreptitious exploitation can also lead to implantation of malicious software (Malware) within your organization, giving the malicious attacker both the ability and potential to attack any time, even perpetually.

Secure Software: It is not just about writing secure code

A common misconception about secure software is that since software is made up of code, secure software is all about technology or code security. While writing secure code (secure programming) is a critical component of the secure software lifecycle, there is a great deal more to consider. Secure software lifecycle is a convergence of policy, processes, and people.

Policy (Pertinent and Enforceable)

Policies, standards, and procedures should be formulated to address software development methodology and establish practical built-in security features. The policies should not just be documented but enforced, tested, and measured. In addition to internal policies that govern software assurance, external regulatory, legal, privacy, and compliance requirements should be factored into the software security requirements. Some examples appropriate to software assurance include identification and authentication policy, remote access policy, use of company resources policy, software security standards, data classification standards, encryption standards, logging and monitoring standards, and disaster recovery and business continuity standards.

Having policy and standard requirements that personnel need comply with (or face serious penalty including termination of employment) has been proven to significantly decrease internal threats.

Threat modeling and security design and architecture review gives the defender the upper hand.

Process (Formal and Structured)

Along with policy controls, secure software processes are needed. In fact, the policies should enable the processes. Secure software processes must include incorporation of security into the SDLC, secure programming, and software risk management. Surprisingly, even today, in spite of the obviously and well-known benefits (including cost savings) of building security in from the very start of development rather than bolting it on after the fact, many organizations still do not have a formalized and structured software development methodology, let alone a secure software development methodology.

Security in the SDLC includes all the processes and technical security controls that need to be in place to develop hacker-proof software. From the initiation stage to the sunset stage, security should be woven into the SDLC methodology. This should, at the bare minimum, during the initiation and development phase, necessitate checks of the software requirements against policy and regulatory, legal, privacy and compliance requirements; as well as include threat modeling, security design and architecture review, secure code development, security code and peer reviews, and quality assurance and testing from a security standpoint. During the implementation and operational phase of a project, vulnerability assessments and penetration testing should be conducted to ensure the software developed was up to specification and not vulnerable to exploits. Finally, during the sunset phase of the project, proper media control measures pertinent to software security should be taken to mitigate issues of data remanence, wherein data is left behind on the media. Secure programming is about writing secure code. Common

Table 1 is an example of secure software controls that should be considered through the life cycle of a software development project. Depending on the type of organizations, some or all of these controls should be factored. Military organizations generally have stringent controls that need to be built in than civilian organizations.

Security in the Systems Development Life Cycle

NIST Phase	SDLC Phase	Security Control
Initiation (Envisioning & Planning)	Requirements Gathering	Business Partner Engagement Identify Policies & Standards Identify Regulatory & Legal Requirements Identify Privacy Requirements Identify Compliance Requirements Develop C,I, A* Goals & Objectives Develop Procurement Requirements Perform Risk Assessment
	Design	Use and Abuse Case Modeling Secure Design Review Secure Architecture Review Threat & Risk Modeling Generate Security Requirements Generate Security Test Cases
Acquisition/ Development	Development	Writing Secure Code Security Code Review Security Documentation
	Testing	Security Testing Redo Risk Assessment
Implementation/ Assessment	Deployment	Secure Installation Vulnerability Assessment Penetration Testing Security Certification & Accreditation Risk Adjustments
Operations/ Maintenance	Maintenance	Change Control Configuration Control Recertification & Reaccreditation Incident Handling Auditing Continuous Monitoring
Sunset (Disposal)	Disposal	Secure Archiving Data Sanitization Secure Disposal Learn and Educate

C,I,A- Confidentiality, Integrity and Availability
Adapted from the NIST Computer Security Division – InfoSec in the SDLC brochure

software vulnerabilities, especially those of Internet-facing software, such as Buffer Overflows, SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), should undoubtedly be addressed and controlled. One aspect of writing secure code is to do so in a manner that makes the code less susceptible to exposures of confidentiality, integrity, availability, authentication, authorization, and audit. The other aspect is to write it as specified in the software design and architecture.

Last, but not least by any means, is the idea that security should always be balanced with risk. President Ronald Reagan's famous words "Trust but verify" have profound meaning in the context of software security. The business necessity of pursuing a competitive advantage in an increasingly threat-pervasive environment means often being forced to trust sensitive information and assets outside the perimeter and to be exposed to potential breach. Software developed should have the necessary verification control measures so that the potential for harm is either addressed, transferred, or mitigated, but never ignored. If your organization has a corporate-wide risk management program, software assurance should be a big part of it. The risk to the organization as a result of exposed or exploited vulnerabilities should be measured. The benefits of this are that it not only gives you a larger, more panoramic view of your organization's risk, but it also allows you to calculate and re-prioritize your efforts. To be truly effective, software assurance very simply requires a strong connection with risk management.

All of the policy and process control security measures are totally futile without the first line of defense – people.

Having software assurance processes in place that are structured and secure means security becomes a part of the software development process as it's happening, rather than having it involved only as an afterthought. Threat modeling and security design and architecture review gives the defender the upper hand.

People (Trained and Qualified)

All of the policy and process control security measures are totally futile without the first line of defense – people. People, the most critical asset of your organization, are absolutely vital in protecting your second-most critical asset, your data. It's essential that your people are not only aware of the need for security, but trained, educated, and qualified to implement it appropriately as well. They should possess a security mindset. In addition to being able to design, develop, and deploy software, they should be able to do so while balancing threats with countermeasures, and business with technology.

Awareness, Training, and Education (AT&E) programs should be tailored to disseminate secure coding policies, processes, and technologies. Your overall environment should be such that your personnel's security awareness is advanced to the level where the security mindset is the default way of thinking.

Conclusion: What Next?

With increased software security incidents, regulatory and compliance requirements, and globalization all changing the landscape of security, one simply cannot take the chance of releasing vulnerable software. Hackers are now targeting your organization's data, putting at great risk your organization and its stakeholders. Damage to your reputation caused by a security breach, and the ensuing loss of customer trust and confidence, might prove irreparable.

In today's business environment, software assurance is imperative. In addition to network perimeter security controls, organizations must ensure that software security controls are designed, developed, and deployed to protect their critical information assets. A secure, formal and structured software development methodology, along with enforceable and pertinent policies, must become a part of any organization's operations. Augmenting this should be trained and qualified people who are empowered with the knowledge of how to implement software security controls, balance threats and countermeasures, and balance business with technology. Anything short of this provides the opportunity for misuse of data, and the potential for an organization to be in the news for all the wrong reasons – being hit with a fine of perhaps a few million dollars, and an entry into the software insecurity hall of shame.

While security awareness, training, and education, are, in and of themselves, powerful in influencing people's behavior and giving them the knowledge to be and stay secure, a stunning combination of software assurance is achieved when these things are combined with a professional certification.

While security awareness, training, and education, are, in and of themselves, powerful in influencing people's behavior and giving them the knowledge to be and stay secure, a stunning combination of software assurance is achieved when these things are combined with a professional certification.

Secure software certification has become a necessity, and (ISC)$^{2®}$, the not-for-profit global leader in educating and certifying information security professionals throughout their careers, can help meet this emerging, vitally-important need.

About (ISC)²®

The International Information Systems Security Certification Consortium, Inc. [(ISC)²®] is the globally recognized Gold Standard for certifying information security professionals. Founded in 1989, (ISC)² has now certified over 60,000 information security professionals in more than 130 countries. Based in Palm Harbor, Florida, USA, with offices in Washington, D.C., London, Hong Kong and Tokyo, (ISC)² issues the Certified Information Systems Security Professional (CISSP®) and related concentrations, Certified Secure Software Lifecycle Professional (CSSLP℠), Certification and Accreditation Professional (CAP®), and Systems Security Certified Practitioner (SSCP®) credentials to those meeting necessary competency requirements. (ISC)² CISSP and related concentrations, CAP, and the SSCP 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 a continuing professional education program, a portfolio of education products and services based upon (ISC)²'s CBK®, a compendium of information security topics, and is responsible for the (ISC)² Global Information Security Workforce Study. 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 currently authoring the Official (ISC)² Guide to the CSSLP, is a contributing author for the Information Security Management Handbook, 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.**

[i] *The Chronology of Data Breaches*
http://www.privacyrights.org/ar/ChronDataBreaches.htm

[ii] Litan, A. *Data Loss Could Have Huge Impact on U.K. Banking Industry.*
Gartner Research Publication (ID Number: G00153682).

[iii] Brewer, C. *IT and Compliance: 10 Big Predictions for 2008*
http://www.itcinstitute.com/display.aspx?id=4693

[iv] Schneier, B. *Inside the Twisted Mind of the Security Professional.*
Wired.com - http://www.wired.com/politics/security/commentary/securitymatters/2008/03/securitymatters_0320

[v] Howard, M. and LeBlanc, D. *Writing Secure Code*, 2nd Ed.

(ISC)^2®