Secure Coding.
Practical steps to defend your web apps.

Sponsored by WhiteHat Security, Inc.

Application Security: Tools for Getting Management Support and Funding

AUGUST 2013

A SANS Whitepaper

Written by John Pescatore

Introduction

Most security professionals understand the importance of finding and eliminating application vulnerabilities. Yet, based on the rate of exploitation of vulnerable websites, it appears that web application protections have been neglected, because of underfunding or lack of focus.

Exploitation of Internet-exposed applications is the leading threat to critical business data and sensitive customer information. Further, recent studies show that the major barrier to effective application security programs is the lack of management buy-in for required resources.

Web application security should start at a project's inception and follow through with management and protection throughout the life cycle. The number of incidents should be reduced when appropriate protections are in place. When you compare the cost of dealing with an incident exploiting your web application to the price of protecting against that incident beforehand, you create a basic metric that those in charge of the budget should truly understand. Using that approach alone has produced limited success, however.

Making the argument for better application security isn't easy. Convincing management to spend on something as ethereal as security requires metrics and tools that demonstrate the efficiencies that secure web application management provides. This paper will provide tools and techniques that demonstrate the need for better application security and the appropriate level of investment.

What Is Application Security and Why Do We Need It?

Applications have been vulnerable for as long as they've existed. Over the past few years, aside from operating systems, they've been cited as the leading vector for attacks. In 2011, NIST attributed 92% of reported vulnerabilities to the application level. The most exposed and, therefore, the first and easiest application to attack is usually the web layer.

Targeting Web Apps

Web applications have become the prime targets for several reasons:

1. By definition, web-based applications are exposed to the Internet with standard interfaces. Attackers can easily discover applications and look for vulnerabilities.
2. Common wisdom dictates that websites need to be refreshed constantly in order to attract and retain users. This often leads to shortcutting configuration management and control procedures, resulting in untested and misconfigured web applications being exposed.
3. Web applications generally consist of hybrids of off-the-shelf software, business- or contractor-developed applications and open source components. Much of the vulnerability in these components goes unnoticed until exploited or a patch notification arrives. In addition, the volatility and complexity of these components can render software development and testing processes ineffective.

Web applications provide attackers with a good launching point to penetrate into the organization, such as a connected database, or to exploit the site in order to download malware onto the computer of customers visiting the site.

These and other risks are among the reasons that applications are so commonly targeted—and why application security is so urgently needed.

Defining Application Security

There are a number of definitions for "application security." In broad terms, application security is about understanding, assessing and managing the risks to an organization's application portfolio.[1] The Critical Security Controls (CSCs) effort defines "Application Software Security" as "The processes and tools organizations use to detect/prevent/correct security weaknesses in the development and acquisition of software applications."[2]

CSC documentation expands this definition to include mitigating application-level vulnerabilities during the deployment and operation of applications. This captures the basic goal of any application security program: reducing application vulnerabilities before they are deployed and reducing the likelihood of successful exploitation of any vulnerability that does make it into production. The desire here is to detect and mitigate vulnerabilities before attackers discover them, preferably during development of the application and ideally before an application is approved for production use, but definitely while the application is operational.

---

[1] http://www.sans.org/reading-room/analysts-program/sans-survey-appsec.pdf

[2] http://www.sans.org/critical-security-controls

Understanding Secure Life Cycle Practices

To meet these definitions, an application security program needs to span the full life cycle of applications. This includes the design and development phase; through coding, integration and testing; into final QA/certification accreditation testing prior to production release; with ongoing vulnerability monitoring and shielding for production applications. Security reviews should be part of all of those phases, using manual processes where necessary, but taking advantage of technology where possible. From a process, controls and technology point of view, this includes:

- **Static application testing:** Detecting and correcting vulnerabilities in individual components of an application at the source, at object code or at binary level.
- **Dynamic application testing:** Detecting vulnerabilities using penetration-testing techniques at the "black box" level during final QA and during operational use.
- **Application-level vulnerability shielding:** Deploying and managing technologies such as application-level or web application firewalls, to prevent the exploitation of vulnerabilities in operational applications before patches or updates can be applied.

Why You Need Application Security

The automotive industry learned decades ago that removing defects before selling a car to the public leads to an overall increase in profitability. The software industry only recently appreciated this position.

The 2013 Ponemon Institute *Cost of a Data Breach Report* looked at 277 security incidents that involved the exposure of personally identifiable information. The study found that the average incident in the U.S. cost $5.4 million. The prevention of just one incident would more than cover the cost of application security.[3]

Reducing application vulnerabilities should mean fewer and less severe security incidents. In that case, application security software not only decreases the attack surface, but also it reduces cost-per-incident, because fewer vulnerabilities need to be repaired. And it is many orders of magnitude less costly to repair software defects pre-production.

While these figures differ for each individual company, several studies confirm this. For example, in the 2002 report *The Economic Impacts of Inadequate Infrastructure for Software Testing*, The National Institute of Standards and Technology (NIST) developed a conceptual model based on taxonomy. It estimated that removing an application defect after production was three times more expensive than removing the defect prior to production release and twice as expensive as removing the same defect after customer beta testing. See Table 1 from the NIST report.

Requirements Gathering and Analysis/Architectural Design	Coding/Unit Test	Integration and Component/RAISE System Test	Early Customer Feedback/Beta Test Programs	Post-Product Release
1X	5X	10X	15X	30X
X *is a normal unit cost and can be expressed in terms of person-hours, dollars, etc.*				

Table 1. Relative Cost to Repair Defects When Found at Different Stages of Software Development (Example Only)

[3] https://www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf

Another report by Forrester Research Group—*The State of Application Security: Immature Practices Fuel Inefficiencies, But Positive ROI is Obtainable*[4]—found that 33% of organizations that emphasized security testing before releasing code for production saw a reduction in developer debugging time equal to the percentage that saw similar benefits from secure design practices. The survey polled 150 organizations on their current state of application security practices to get feedback on the benefits of application security considerations during development, quality assurance and pre-preproduction testing of software.

Barriers to Adopting Application Security Practices

In December 2012, The SANS Institute surveyed 700 organizations regarding their application security programs and practices. That survey revealed that the primary barriers to implementing secure application management programs were a "lack of management funding/buy-in," followed by a lack of resources and skills. See Figure 1.

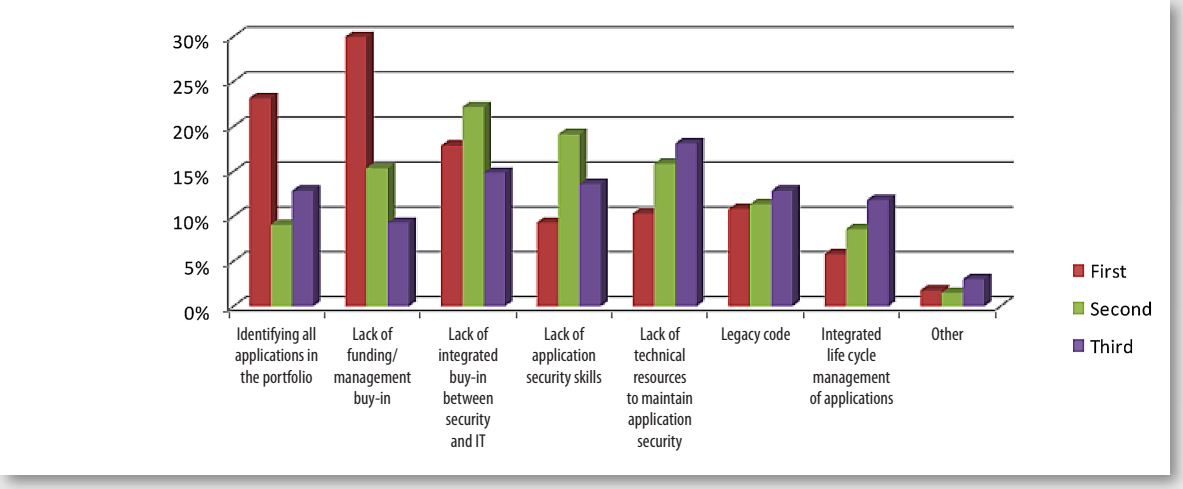

Figure 1. Barriers to Implementing Application Security Programs[5]

A 2013 Microsoft survey obtained similar results. In that survey more than 2,200 IT professionals and 490 developers worldwide were asked about secure development life cycle processes. The top barriers cited were lack of management approval, lack of training and support, and cost.[6]

Both of these studies reveal that security professionals understand that application security is important, but have been unable to convince IT and corporate management to allocate sufficient resources to fund effective application security programs.

[4] http://download.microsoft.com/download/5/F/F/5FF064F6-34C0-4EFE-BA6D-6EA492CC20D3/State%20Of%20Application%20Security.pdf

[5] http://www.sans.org/reading_room/analysts_program/sans_survey_appsec.pdf

[6] http://blogs.technet.com/b/trustworthycomputing/archive/2013/05/08/security-development-conference-2013.aspx

Obtaining Management Buy-In for Application Security

As the saying goes, hindsight is a wonderful thing. It explains why sales of backup power generators surge *after* a natural disaster. Similarly, one tried-and-true method for obtaining management approval for application security is to wait until after your organization experiences a damaging incident caused by a vulnerable application. In that event, make sure the incident report (whether done internally or by an external consultancy) contains information on the cause of the breach and recommendations for remediation—including any holistic improvements needed in the life cycle. If financial losses can be proven, that is all the better.

Using a Publicized Incident to Illustrate Risk/Benefit

The goal of this paper is to provide tools to obtain management support *before* such an incident. So the next best thing is public data on an incident that happened to *another* company—ideally a peer in your industry. These incidents don't always make the press, but there are a number of good sources to help you find them:

- **Verizon Data Breach Investigations Report (DBIR):**[7] For the past six years, the Verizon DBIR has become one of the largest sources of data on security incidents. The latest version analyzed 47,000 reported security incidents that represented more than 600 confirmed security breaches and continues to show web applications as a leading vector.
- **Ponemon Institute:**[8] Funded by vendors, the Ponemon Institute conducts surveys and studies on the cost of cybercrime and security incidents.
- **Microsoft Security Intelligence Report (SIR):**[9] Using data collected from more than 600 million Windows-based computers worldwide, the Microsoft SIR provides information on attack trends and global distribution.
- **Web Hacking Incident Database:**[10] The Web Hacking Incident Database (WHID) is a Web Application Security Consortium project maintaining a list of web application–related security incidents.
- **eSecurity Planet hacker news tracking:**[11] The eSecurity Planet website collects public reports on compromised websites and makes them available online.

A recent example from the WHID was the exposure of 80,000 user accounts when a World Wildlife Federation web server was compromised via an SQL injection attack. This example can be used by nonprofit organizations, which often have a hard time convincing management they would ever be targeted. Using these resources, you can easily pull together a "This could happen to us, unless … " recommendation to management. If costs associated with the breach are demonstrated, that would help set up the case of cost for securing apps versus the cost of not securing apps.

---

[7] http://www.verizonenterprise.com/DBIR/2013/

[8] http://www.ponemon.org/library

[9] http://www.microsoft.com/security/sir/default.aspx

[10] http://projects.webappsec.org/w/page/13246995/Web-Hacking-Incident-Database#SearchtheWHIDDatabase

[11] http://www.esecurityplanet.com/hackers

Managing Regulatory Pull

While compliance does not equate to security, corporate managers are cognizant of and responsive to compliance requirements for their industries and data being processed through their applications. This is especially true for chief legal counsels and chief financial officers. A number of regulations and compliance make direct reference to application security requirements.

Starting with the Payment Card Industry (PCI), all companies accepting credit card payments online face security requirements. Two—the PCI Data Security Standard (PCI DSS) and the PCI Application Data Security Standards (PCI PA DSS)—are relevant to application security. PCI DSS Section 6, 11.2, 11.3 mandates secure development practices, application vulnerability scanning, penetration testing and the use of web application firewalls when vulnerabilities can't be eliminated.[12] PCI PA DSS Sections 5.2 and 7.1 require that payment application software developers use secure development practices that reduce vulnerabilities and that they test all software for vulnerabilities.[13]

NIST also mandated several controls related to application security in federal systems, including NIST 800-53 Rev. 4, "Security and Privacy Controls for Federal Information Systems and Organizations," (controls RA-5, SA-11, SC-7)[14] and NIST 800-64, Rev. 2 "Security Considerations in the System Development Life Cycle."[15]

Industry-based application security specifications also exist for nontraditional devices and their applications, such as medical and supervisory control and data acquisition (SCADA), or other control systems. For example, the Food and Drug Administration (FDA) provides guidelines for medical device security,[16] and NERC CIP-005 R4 and CIP-007 R8 incorporate application security into guidelines for connected control devices.[17] Regardless of industry or regulation, core tenets of most of these security requirements address the three stages of application life cycle management, starting with development and then managing applications in use. These steps include secure development processes and testing, vulnerability scanning, patching and mitigation of applications in production.

[12] https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf

[13] https://www.pcisecuritystandards.org/security_standards/documents.php?association=PA-DSS

[14] http://dx.doi.org/10.6028/NIST.SP.800-53r4

[15] http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf

[16] http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM356190.pdf

[17] http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx

Taking Advantage of Industry Governance Standards

In addition to regulatory standards, community standards and frameworks also provide implementation advice. These include the CSCs, wherein Controls 3 and 6 specifically describe processes for assessment and secure configuration of software applications.[18] Another well-known industry group is the Open Web Application Security Project (OWASP). The OWASP top 10 examines the 10 most commonly exploited application vulnerabilities that, at a minimum, developers and application managers need to guard against.[19] These include common mistakes in coding, such as failure to validate input, which makes applications prone to SQL injections, cross-site scripting (XSS), and more.

Many IT organizations have established governance standards to improve the quality and efficiency of IT operations. These standards address security in general, and many address application security—either directly or by reference. If your organization has committed to any of these efforts, there are numerous areas where you can attach the need for improving practices in application security:

- **Information Technology Infrastructure Library (ITIL):** Widely adopted set of practices for IT service management (ITSM) published in a series of five core publications, each of which covers a stage in the ITSM life cycle. ITIL essentially provides the best practices underpinning for ISO/IEC 20000. Based on the ISO 27001 standard, the ITIL security management process has been integrated into those life cycle stages in ITIL and describes the structured fitting of security in the management organization.

  ITIL includes "Management of Risks" across all life cycle elements. This includes a coordinated set of activities that identifies and assesses vulnerabilities and control risks.[20]

- **Control Objects for Information Technology (COBIT):** Developed by ISACA, COBIT provides an IT governance framework to help enable managers to bridge the gap between control requirements, technical issues and business risks. COBIT Control Objectives Planning and Organization, Assess Risks (PO9), Acquire and Maintain Application Software (AI2), Delivery and Support, Ensure System Security (DS5) and Monitor Processes (M1) apply directly to application security.[21]

- **ISO 27034:**[22] In November 2011, Subcommittee SC27 of the International Standards Organization (ISO) (also responsible for the ISO 27001 series of security documents) published ISO/IEC 27034-1, "Information technology – Security techniques – Application security," which focuses on managing overall levels of application security. It takes a process-based approach by defining a targeted level of trust (TLT) for an application and then assuring that the design, development, test and operational security policies exist to assure that the TLT is met. A related standard that is still under development (ISO 27036) extends this idea to supply chain risk management.

  Microsoft has become the industry's most vocal proponent of ISO 27034, recently using it as a baseline for assessing its own software development life cycle processes. Demand for increased levels of application security will increase as other parts of ISO 27034 are published and these ISO standards become increasingly seen in tenders and RFPs.

[18] http://www.sans.org/critical-security-controls

[19] https://www.owasp.org/index.php/Top_10_2013-Table_of_Contents

[20] http://www.itil-officialsite.com/home/home.aspx

[21] http://www.isaca.org/cobit/pages/default.aspx

[22] http://www.iso.org/iso/catalogue_detail.htm?csnumber=44378

- **Capability Maturity Models:** In the 1980s, the U.S. Department of Defense (DOD) realized that software development was becoming the greatest risk in building, deploying and supporting complex defense systems. The DOD funded the Software Engineering Institute at Carnegie Mellon University (CMU) to develop a software engineering capability maturity model. This organization defined a number of maturity levels (from Chaotic to Optimizing) with the reduction in software defects and vulnerabilities a key benefit of moving to higher levels of maturity. This became the Capability Maturity Model Integration (CMMI).[23]

  In the mid-1990s, the U.S. National Security Agency initiated a Systems Security Engineering CMMI effort, modeled after the SEI CMMI. This became ISO/IEC 21827:2008, which describes the essential characteristics of an organization's security engineering process, including key security issues with applications. This standard is not widely used, however.

  Gartner Inc. developed its proprietary IT Score for Security and Risk management modeled after the CMMI. The Gartner IT Score has been applied across several domains, with application security featured in information security and risk management. Information and toolkits are available to Gartner services subscribers.[24]

- **Benchmarking:** Capability maturity models are essentially a form of benchmarking, or comparing your security processes against defined levels of maturity. This type of benchmarking focuses on the *effectiveness* of your security processes as defined in the maturity model. The more common form of benchmarking focuses on the *efficiency* of processes. It compares the cost of hardware, software, personnel and services needed to secure applications against costs to organizations similar to yours. For example, CIOs have longed used benchmarking to justify help desk staffing or telecoms expenditures as a function of company revenue and headcount. CEOs often use industry benchmarks to look at marketing or R&D spending levels compared to an organization's peers.

  To be done reliably, benchmarking should be applied against mature, well-defined processes. This has limited the scope of security benchmarking. It has also created a "chicken and egg" problem, because to be meaningful, benchmarking results need to be compared against a large database of samples. There are some efforts underway by OWASP, Security Leadership Research Institute,[25] Information Security Forum Benchmark Service and analyst surveys, for example.

Understanding Return on Investment and Total Cost of Ownership Models

Corporate managers often use Return on Investment (ROI) or Return on Capital models to evaluate funding requests. ROI models generally establish a predicted yearly return by looking at both expenditures and revenue over time. While ROI models work well for pure financial investment analysis, two major problems with trying to use ROI arguments for security exist. First, security controls are typically not directly related to increases in revenue. Second, the investment in security controls is a complex mix of hardware, software, support and personnel.

To address these issues, the Australian government developed a methodology called *Return on Security Investment (ROSI)*[26] and spreadsheet-based tools for performing the calculations. The ROSI approach looks at "Avoidable Loss Expectancy" as the return and the total cost of ownership of the security solution as the investment.

---

[23] http://cmmiinstitute.com/

[24] http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1422519&ref=QuickSearch&sthkw=itscore+security

[25] https://www.securityexecutivecouncil.com/about/spotlight.html?sid=25509

[26] http://www.finance.nsw.gov.au/inside-dfs/information-communications-technology/publications/return-security-investment

Pulling the Numbers Together: A Strawman Budgetary Model

Corporate managers are familiar with using benchmarks to evaluate spending. Here, we define a simple model using public data that can be tailored to your needs.

Methodology

Spending and risk are the two assumptions in this simple model, both captured in a spreadsheet shown on the next page.

Assumptions on spending. Starting with your company's revenue, use the publicly available ratios of IT spending as a percentage of revenue, security spending to IT spending, and application security spending as a percentage of security spending to produce an "average" level of application spending.

Based on publicly available data, the model starts with the following average spending assumptions, adjusted to match the different types of companies used in each example. The user can update or modify these assumptions as new data becomes available publicly or through your company's subscription to research services:

1. **IT spending as a percentage of revenue:** 4.2% (Gartner 2009)
2. **Security spending as a percentage of IT revenue:** 5.6% (Gartner 2011)
3. **Application security spending as a percentage of security spending:** 33% (BSIMM 3, mature programs)

Application risk and environmental assumptions. This risk level will be used to adjust the recommended level of spending to be above or below the average, depending on estimated risk. We've based this on the Common Vulnerability Scoring Standard (CVSS), a widely used method of assigning risk levels to vulnerabilities, including the capability to factor in unique aspects of your application environment. This model uses a slightly modified CVSS spreadsheet to produce a CVSS-like score for an organization's application risk level.[27]

Combining these two factors produces a "recommended" percentage of the security budget be allocated to application security. This will be an Excel spreadsheet delivered with the report. A few examples of using this method are shown next. In each case, the only variable changed in the application risk level assessment was the environmental factor.

[27] http://nvd.nist.gov/cvss.cfm?calculator

Risk vs. Investment Examples

Here are three examples of using this approach, highlighting different sized companies in different industries with different risk levels and budgets.

Example 1: Small Co.

Small Co. is a 1,000-employee company doing $100M in revenue that is not in a regulated business. It has a medium application risk environment, average levels of IT and security spending, and an immature application security program. In this case, the model shows typical spending on application security would be approximately $48K/year.

Small Co.	
Revenue	$100,000,000
IT Spend/Rev factor	4.2%
IT Spend	$4,200,000
Sec Spend/IT factor	5.6%
Sec Spend	$235,200
App Sec/Sec factor	20.0%
App Sec spend	$47,040
Risk Adjust Factor	102.0%
Final App Sec Spend	$47,981

Using this model, Small Co. would compare its existing application security spending level against the $48K prediction. Many small companies don't perform regular web applications security scanning or pre-production code testing. They would fall below that level, being more at risk due to under-spending, something that can be easily communicated to management. Organizations that are spending above this level and continue to experience website vulnerabilities and successful attacks should re-evaluate their approach to application security.

Example 2: HighSec Co.

HighSec Co. does $1.5 billion in annual revenues. It is in a regulated industry such as finance, spends slightly above average on IT and IT security, and has a mature approach to application security. In this example, the model shows typical spending on application security would be approximately $2.3M/year.

HighSec Co.	
Revenue	$1,500,000,000
IT Spend/Rev factor	5.0%
IT Spend	$75,000,000
Sec Spend/IT factor	6.0%
Sec Spend	$4,500,000
App Sec/Sec factor	33.0%
App Sec spend	$1,485,000
Risk Adjust Factor	156.0%
Final App Sec Spend	$2,316,600

In this example, HighSec Co. has much to lose if applications are compromised and should be funding all three elements of an effective application security program (Static Application Testing, Dynamic Application Testing, and Application Vulnerability Shielding).

Example 3: Wide Open Co.

Wide Open Co. is a 5,000-employee organization in a widely distributed, open environment (such as R&D or university-level academics). It has below industry average spending on IT and IT security. It also has a small number of applications handling personally identifiable information and an average level of maturity in its approach to application security. The model shows typical spending on application security for an organization similar to Wide Open Co. would be approximately $30K/year.

Wide Open Co.	
Revenue	$200,000,000
IT Spend/Rev factor	4.0%
IT Spend	$8,000,000
Sec Spend/IT factor	5.0%
Sec Spend	$400,000
App Sec/Sec factor	25.0%
App Sec spend	$100,000
Risk Adjust Factor	30.0%
Final App Sec Spend	$30,000

Universities are good examples of environments that must maintain an open environment, and they often have very immature IT and security processes. Wide Open Co. may only be able to afford weekly web application vulnerability scanning, but it could use open source application vulnerability shielding approaches to minimize the risk quickly after vulnerabilities are discovered.

Many security managers haven't experienced a major security incident or been subject to compliance regulations. As a result, they may not have had the opportunity to be part of an IT governance or process improvement effort. This is when the budgetary model approach should be used, but you should tailor it to your own organizational "language." Also, remember that funding for application security improvements competes against other security investments, overall IT investments, business investments, cost-cutting initiatives, etc. Look at examples within your organization where competing groups successfully obtained management buy-in. Borrow from their successes and use those methods to better focus your argument and data.

Conclusion

In this paper we identified a number of approaches for convincing management to invest in an appropriate application security program. Which approach, or combination of approaches, is best depends on factors unique to each organization and its current business, IT and risk environment.

The number one goal is to prevent security incidents. But if your company has recently experienced a breach in application vulnerabilities, or you are aware of a public incident involving a peer company or direct competitor, by all means take that approach and show decision makers the cost and results of the breach. Strike while the iron is hot—the searing pain of a major security incident has a half-life of approximately 90 days, or one business quarter.

Matching application security with compliance requirements is often the next most effective approach to gaining support for application security. But this practice can be a double-edged sword. While CEOs, COOs, and chief legal counsels understand the need to be compliant, their experiences may convince them that the goal is to meet the compliance demands at the lowest possible cost. This often means simply producing more pages of submission reports to feed the compliance regime. Relying solely on "PCI says we have to do this," as the argument may work in the short term, but it doesn't lead to the sustainable investments needed to foster meaningful risk reduction.

Getting application security investments embedded into IT governance, process improvement or benchmarking efforts has proven to be a highly effective approach. Security managers who have succeeded here and cultivated their peers in application development, QA and audit have been able to drive major increases in the maturity of application security and see meaningful reductions in vulnerabilities and incident costs.

About the Author

John Pescatore joined SANS in January 2013, with 35 years of experience in computer, network and information security. He was Gartner's lead security analyst for more than 13 years, working with global 5000 corporations, government agencies and major technology and service providers. In 2008, he was named one of the top 15 most influential people in security and has testified on cybersecurity before Congress.

Prior to joining Gartner Inc. in 1999, Mr. Pescatore was senior consultant for Entrust Technologies and Trusted Information Systems, where he started, grew and managed security consulting groups focusing on firewalls, network security, encryption and public key infrastructures. Prior to that, he spent 11 years with GTE developing secure computing and telecommunications systems. In 1985 he won a GTE-wide Warner Technical Achievement award.

Mr. Pescatore began his career at the National Security Agency, where he designed secure voice systems; and the United States Secret Service, where he developed secure communications and surveillance systems, and the occasional ballistic armor installation.

He holds a bachelor's degree in electrical engineering from the University of Connecticut and is an NSA-certified cryptologic engineer. He is an Extra class amateur radio operator, call sign K3TN.

SANS would like to thank its sponsor:

Last Updated: September 25th, 2017

Upcoming SANS App Sec Training

Secure DevOps Summit & Training	**Denver, CO**	**Oct 10, 2017 - Oct 17, 2017**	**Live Event**
SANS Seattle 2017	**Seattle, WA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS Austin Winter 2017	**Austin, TX**	**Dec 04, 2017 - Dec 09, 2017**	**Live Event**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS Cyber Defense Initiative 2017 - DEV522: Defending Web Applications Security Essentials	**Washington, DC**	**Dec 14, 2017 - Dec 19, 2017**	**vLive**
SANS Security East 2018	**New Orleans, LA**	**Jan 08, 2018 - Jan 13, 2018**	**Live Event**
SANS Amsterdam January 2018	**Amsterdam, Netherlands**	**Jan 15, 2018 - Jan 20, 2018**	**Live Event**
SANS Oslo 2018	**Oslo, Norway**	**Feb 05, 2018 - Feb 10, 2018**	**Live Event**
Community SANS Indianapolis DEV534	**Indianapolis, IN**	**Feb 05, 2018 - Feb 06, 2018**	**Community SANS**
Cloud Security Summit & Training 2018	**San Diego, CA**	**Feb 19, 2018 - Feb 26, 2018**	**Live Event**
Communtiy SANS Seattle DEV534	**Seattle, WA**	**Feb 26, 2018 - Feb 27, 2018**	**Community SANS**
SANS San Francisco Spring 2018	**San Francisco, CA**	**Mar 12, 2018 - Mar 17, 2018**	**Live Event**
San Francisco Spring 2018 - DEV522: Defending Web Applications Security Essentials	**San Francisco, CA**	**Mar 12, 2018 - Apr 17, 2018**	**vLive**
SANS Northern VA Spring - Tysons 2018	**Tysons, VA**	**Mar 17, 2018 - Mar 24, 2018**	**Live Event**
SANS 2018	**Orlando, FL**	**Apr 03, 2018 - Apr 10, 2018**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**