HCL

Application Security

An SDLC Imperative

Authors

Sunil Anand, Architecture and Technology Services (ATS)
HCL Technologies, NOIDA

Dr Usha Thakur, ATS Technical Research
HCL Technologies, Chennai

Application Security: An SDLC Imperative

November, 2009

Contents

Introduction

We have come a long way from when knowledge was communicated and received orally within a restricted group of people (for example from father to son) on a number of specialised subjects such as medicinal herbs, weather patterns, water currents, agriculture, money matters, and so on. In the 20th century, the written word has given us a platform for not only sharing knowledge with a much wider audience but also engaging in commercial transactions with them; what varies is the medium. Until recently, print was a very popular medium, but the Information Technology revolution has sobered its reach and impact such that today we cannot imagine storing data or information in mediums other than a software application.[1] And, it goes without saying that the priority an organisation gives to the security aspects of a software application is relative to the significance it gives to the data stored in that application. Today, many applications are accessed via the Web and the need to secure them (at the application level) is greater than ever before.

Purpose

The main objective of this paper is to examine why application security is high on the agenda of many organisations and what are the key elements of security architecture. It also highlights the long-term business benefits of making security an integral part of the software development life cycle (SDLC) and gives a glimpse of HCL's Application Security Service.

Why Protect Applications?

Common sense tells us to protect our valuable assets (movable and immovable) against their violation. In order to protect them, however, we must first anticipate all the possible ways and means in which our assets may be violated. We cannot have a security system in place to protect our house unless we have prior knowledge about two aspects:

1. Known vulnerable spots from where intruders may gain entry into the house
2. Assets within the house that need to be protected

At a conceptual level, protecting information is no different. Instead of physical assets within a house, we are dealing with assets (i.e. data) of many people in electronic format. Naturally, there are some security rules and regulations according to which data can be stored and accessed. For instance, a bank needs to ensure that all data pertaining to our accounts in its IT system is accurate and that it can be accessed only by legitimate account holders and authorized bank personnel. This is easier said than

[1] Throughout this paper we will use **software application**, **application**, and **system** interchangeably.

done, given the fact that in our society we have people who wish to access assets (in this case money) that do not belong to them. This is no mom-pop activity but involves professionals whose job is to hack the security fence of IT systems in order to gain access to information, which could be used for benign or malicious purposes. Let us take a brief look at just how big this business really is.

Here are some high-profiled incidences of data breach:

- Although hacking activity is said to have started in May 2008, Heartland Payment Systems (provider of credit and debit card processing services) got wind of it only in October 2008 "after being alerted by Visa® and MasterCard® of suspicious activity surrounding processed card transactions. Heartland enlisted the help of several forensic auditors to conduct a thorough investigation into the matter…the investigation uncovered malicious software that compromised data that crossed Heartland's network."[2] This went on to be reported as the "world's biggest data breach" by the news media involving over 100 million credit cards.[3] It was only in August 2009 that the Department of Justice made an announcement about one suspect.[4]

- The Heartland data breach occurred a few months after the breach of IT systems at TJX, a world renowned American retailer in the apparel and home fashions business. In this case, some 45 million credit cards were compromised.[5]

- In December 2008, RBS WorldPay (formerly RBS Lynk), the U.S. payment processing arm of The Royal Bank of Scotland Group acknowledged that its "computer system had been improperly

---

[2] See "Heartland Payment Systems Uncovers Malicious Software In Its Processing System," http://www.2008breach.com/Information20090120.asp [September 2009]->represents when this website was accessed.

[3] For details of this case, see Bill Brenner, "Fixing the World's Biggest Data Breach," (August 13, 2009), http://www.computerworlduk.com/management/security/cybercrime/in-depth/index.cfm?articleid=2438 [September 2009]. See also Rachael King, "Lessons from the Data Breach at Heartland," (July 6, 2009) http://www.businessweek.com/technology/content/jul2009/tc2009076_891369.htm [September 2009].

[4] "[t]he indictment, which details the largest alleged credit and debit card data breach ever charged in the United States, alleges that beginning in October 2006, Gonzalez and his co-conspirators researched the credit and debit card systems used by their victims; devised a sophisticated attack to penetrate their networks and steal credit and debit card data; and then sent that data to computer servers they operated in California, Illinois, Latvia, the Netherlands and Ukraine. The indictment also alleges Gonzalez and his co-conspirators also used sophisticated hacker techniques to cover their tracks and to avoid detection by anti-virus software used by their victims." See "Alleged International Hacker Indicted for Massive Attack on U.S. Retail and Banking Networks," Press Release (August 17, 2009), http://www.usdoj.gov/opa/pr/2009/August/09-crm-810.html [September 2009].

[5] See Jaikumar Vijayan, "Heartland Data Breach could be Bigger than TJX's," (January 20, 2009), http://www.computerworld.com/s/article/9126379/Heartland_data_breach_could_be_bigger_than_TJX_s September 2009].

accessed by an unauthorized party, affecting "approximately 1.5 million cardholders and other individuals."[6]

- In 2005 the US Air Force discovered a massive breach in its Assignment Management System at Randolph Air Force Base, Texas, whereby unknown quantities of data and information in such areas as commence and control, logistics, personnel, scheduling, and even in classified research and development areas were downloaded by a hacker, whose identity remains unknown.[7]

These are just a few well-known cases; in actual fact, as reflected in the studies of IDTheftCenter.org, privacyrights.org, and datalossdb.org, the number of data breach incidence is quite high.[8]

All data breaches, irrespective of their size prove expensive for businesses that are attacked. Heartland has already spent $12.6 million on activities resulting from the intrusion of its system and is said to have set aside $32 million for breach expenses.[9] TJX is said to have set aside $118 million after-tax in the second quarter of 2007 to cover the costs arising from the data breach of its system. Furthermore, it is estimated that TJX have spent $125 million before-tax dollars on security improvements, both before and after the breach. [10] Overall, TJX is said to have incurred more than $171 million in expenses related to the attack on its IT systems. [11]

According to the results of the fourth annual *U.S. Cost of a Data Breach Study* by PGP Corporation and Ponemon, "data breach incidents cost U.S. companies $202 per compromised customer record in 2008, compared to $197 in 2007. Within that number, the largest cost increase in 2008 concerns lost business created by abnormal churn, meaning turnover of customers...[S]ince the study's inception in 2005, this

---

[6] See "Press Release: RBS WorldPay Announces Compromise of Data Security and Outlines Steps to Mitigate Risk," (December 23, 2008), http://www.rbsworldpay.us/media/news_media25.htm [September 2009].

[7] For full details see Mark Kagan, "Best Practices: ProveIT Case Study for U.S. Air Force Software Assurance Center of Excellence," http://www.appsecinc.com/techdocs/whitepapers/IDC-Case-Study-Featuring-Application-Security-Inc.pdf [August 2009].

[8] See http://datalossdb.org/statistics [September 2009], http://www.privacyrights.org/ar/ChronDataBreaches.htm#CP [September 2009], and http://www.idtheftcenter.org/artman2/publish/lib_survey/ITRC_2008_Breach_List.shtml [September 2009].

[9] See Bill Brenner, Op cit., p. 2. See also Alex Goldman, "Heartland Hit With $12M Breach Tab," (May 8, 2009), http://www.internetnews.com/security/article.php/3819596 [November 2009] and Digital Transaction, "Heartland Hit by Drop in Same-Store Sales, But Encryption Moves Ahead," (November 27, 2009) http://www.digitaltransactions.net/newsstory.cfm?newsid=2284 [November 2009].

[10] Avivah Litan, "Use TJX Breach to Improve Protection of Customer Data," (August 20, 2007) http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&id=513206&ref= [November 2009].

[11] Rachael King, Op cit.

cost component has grown by more than... 40%. [The average] total per-incident costs in 2008 were $6.65 million, compared to an average per-incident cost of $6.3 million in 2007." [12]

Recently, the researchers at Purdue University's Center for Education and Research in Information Assurance and Security conducted a study on the security of information in eight countries. Announcing the result of the findings, McAfee revealed that the companies surveyed lost a "combined $4.6 billion worth of intellectual property last year alone, and spent approximately $600 million repairing damage from data breaches. Based on these numbers, McAfee projects that companies worldwide lost more than $1 trillion last year."[13]

There is, indeed, a general consensus that it is relatively cheaper to build security features within a software application than to try and cover the gaps later. Companies that have had to incur the cost of a data breach as a result of insecure applications need no convincing; they have, in fact, already taken steps to secure their IT applications and network.

Need for Application Security

Businesses, institutions, and government organisations are now acknowledging the fact that it is not enough to protect only their IT infrastructure with firewalls and various intrusion detection systems (IDS). Since the focus of attackers has shifted to exploiting vulnerabilities in application design, source code, runtime code, and deployment configurations, there is a consensus among software experts that security needs to become an integral part of the software development life cycle (SDLC).

According to Security Expert, Theresa Lanowitz,

[12] See "Press Release: Ponemon Study Shows Data Breach Costs Continue to Rise," (February 2009) http://www.pgp.com/insight/newsroom/press_releases/2008_annual_study_cost_of_data_breach.html [September 2009]. See also Ross Kerber, (August 15, 2007) Cost of data breach at TJX soars to $256m Suits," http://www.boston.com/business/globe/articles/2007/08/15/cost_of_data_breach_at_tjx_soars_to_256m/?page=2 [September 2009].

[13] See, "McAfee, Inc. Research Shows Global Recession Increasing Risks to Intellectual Property," http://www.mcafee.com/us/about/press/corporate/2009/20090129_063500_j.html [September 2009]. Kevin Prince has compiled a list of the cost companies had to incur following a security breach. The cost, which runs into millions takes into account factors such as lost productivity, lost customer opportunity, remedial measures, class actions by those affected by the breach, cost per lost records, and high customer churn. See Kevin Prince, "A Comprehensive Study of Financial Data Security Breaches in the United States – 2008," http://www.perimeterusa.com/databreach_wp-ty.html [September 2009], pp 21-21.

Businesses have always been under threat of reliability and security events due to vulnerable source code… 75 percent of hacks occur at the application level... [Therefore, the] best network, host and data security can't effectively protect a weak application. Security must be considered first in the application.[14]

Increasingly the focus is now on fixing vulnerabilities during architecture, design and coding phases as opposed to testing and maintenance.[15] Making a code secure implies eliminating or significantly reducing the likelihood of buffer overflows, error handling, command injection, unnecessary code, malicious code, broken threads, invalidated parameters, cross-site scripting, caching, pooling and reuse errors.[16] Application security can, therefore, be enhanced only if software teams focus on:

- ✓ Capturing security requirements early in the project life cycle (according to the relevant Enterprise Information Security standards and the Compliance and Regulatory requirements)
- ✓ Ensuring proactive security assessment of architectural blueprint
- ✓ Reducing security vulnerabilities and risks by implementing secure coding practices
- ✓ Improving security features and functions (e.g., authentication, encryption or auditing)
- ✓ Integrating security as part of the project life cycle
- ✓ Considering the security hosting environment and the enterprise security infrastructure

[14] Theresa Lanowitz, "Now Is the Time for Security at the Application Level," (December 1, 2005) http://www.sela.co.il/_Uploads/dbsAttachedFiles/GartnerNowIsTheTimeForSecurity.pdf [September 2009] pp 3-4.

[15] According to research conducted by Cigital, identifying and fixing vulnerabilities in software during development can save companies more than $2 million (on a code base of 2 million lines of code). Study quoted by Fortify in "The Case for Application Security: How Real Is Your Threat and What Are Your Options," http://www.fortify.com/landing/downloadLanding.jsp?path=%2Fuser%2FFortify_Case_For_Application_Security.pdf [August 2009].

[16] For a comprehensive analysis (by MITRE Corporation) of trends in Common Vulnerability Exposure (CVE) over a five year duration, see Steve Christey and Robert A. Martin, "Vulnerability Type Distributions in CVE," (May 22, 2007) http://cwe.mitre.org/documents/vuln-trends/index.html [September 2009]. See also, OWASP, "Top 10 to 2007," http://www.owasp.org/index.php/Top_10_2007 [September 2009] and WASC, "Threat Classification," http://www.webappsec.org/projects/threat/ [August 2009].

Today, a number of processes, standards, life-cycle models, frameworks, and methodologies are available for developing secure software[17]; some of the well-known ones are as follows:

- System Security Engineering Capability Maturity Model (SSE-DMM)[18]
- Microsoft's Trustworthy Computing Security Development Lifecycle[19]
- Software Assurance Maturity Model (SAMM), which is maintained through the Open Web Application Security Project (OWASP)[20]
- Software Security Framework (SSF), which was developed jointly by Cigital and Fortify[21]
- Open Source Security Testing Methodology Manual (OSSTMM), which is a peer-reviewed methodology for performing security tests and metrics[22]
- Software Engineering Institute's (SEI) Team Software Process for Secure Software Development (TSP-Secure)[23]
- Praxis High Integrity Systems' Correctness by Construction Methodology[24]

---

[17] For a concise summary, see Noopur Davis, "Secure Software Development Life Cycle Processes," (Carnegie Mellon University: July 21, 2009) https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/sdlc/326-BSI.html [August 2009].

[18] For further information, see http://www.sse-cmm.org/index.html [September 2009].

[19] For details, see Steve Lipner and Michael Howard, "The Trustworthy Computing Security Development Lifecycle," (March 2005) http://msdn.microsoft.com/en-us/library/ms995349.aspx [September 2009].

[20] For details on this models, see http://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model [September 2009].

[21] For information on SSF, see http://www.bsi-mm.com/ssf/ [September 2009]. See also Gary McGraw and Brian Chess, "Software [In]security: A Software Security Framework: Working Towards a Realistic Maturity Model," (October 15, 2008) http://www.informit.com/articles/article.aspx?p=1271382 [September 2009].

[22] For details, see Pete Herzog, "OSSTMM - Open Source Security Testing Methodology Manual," http://www.isecom.org/osstmm/ [November 2009].

[23] Carnegie Mellon Software Engineering Institute, "Team Software Process," http://www.sei.cmu.edu/tsp/ and http://www.sei.cmu.edu/library/abstracts/presentations/tspsecure.cfm [September 2009].

[24] For details of this methodology, see http://www.praxis-his.com/services/software/principles.asp [September 2009]. See also Martin Croxford and Dr. Roderick Chapman, "Correctness by Construction: A Manifesto for High-Integrity Software," (December 2005), http://elsmar.com/pdf_files/A%20Manifesto%20for%20High-Integrity%20Software.pdf [September 2009].

- Agile Methods (that are compatible with Security Assurance Practices)[25]
- ISO/IEC 15408 Standard, which is based on the Common Criteria for Information Technology Security Evaluation (CCITSE)[26]
- Software Security Touchpoints[27]

Although the manner in which each company integrates security into its SDLC may vary, [28] the basics, as illustrated in Table 1 are likely to be similar.

[25] See Konstantin Beznosov and Phillipe Kruchten, "Towards Agile Security Assurance," *Proceedings of the 2004 Workshop on New Security Paradigms* *http://portal.acm.org/citation.cfm?id=1066034&dl=GUIDE&coll=GUIDE&CFID=52354040&CFTOKEN=57262083* and *http://konstantin.beznosov.net/doc/talks/Towards_Agile_Security_Assurance-Waterloo_presentation.pdf* [September 2009].

[26] See "ISO/IEC 15408-1:2005 Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model," http://www.iso.org/iso/catalogue_detail.htm?csnumber=40612 [September 2009]; "ISO/IEC 15408-2:2005 Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements," http://www.iso.org/iso/catalogue_detail.htm?csnumber=40613 [September 2009]; "ISO/IEC 15408-3:2008 Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance components," http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46413 [September 2009].

[27] Nancy Mead and Gary McGraw, "Portal for Software Security," (IEEE Computer Society: 2005) http://www.cigital.com/papers/download/bsi9-portal.pdf [September 2009].

[28] Take for instance HP's Application Security Maturity Model and Arctec's Security Architecture Blueprint, or US Department of Homeland Security (DHS) National Cyber Security Division's (NCSD) Building Security In. For details see, HP, "Mandate for App Security_3," https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200^14344_4000_100__ [August 2009]; Gunnar Peterson, "Security Architecture Blueprint, " http://arctecgroup.net/pdf/ArctecSecurityArchitectureBlueprint.pdf [August 2009]; Nancy Mead and Gary McGraw, Ibid. See also Nancy R. Mead, et al., "Incorporating Security Quality Requirements Engineering (SQUARE) into Standard Life-Cycle Models," (TECHNICAL NOTE: CMU/SEI-2008-TN-006) http://www.sei.cmu.edu/library/abstracts/reports/08tn006.cfm [September 2009].

Table 1: Building Security into SDLC

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
Policy and Standards for Development or Acquisition and Maintenance of any System	Use any one of the following Standards or any other that best fits your organization's needs: NIST Special Publication 800-64, ***Security Considerations in the Information System Development Life Cycle***[29] ISO/IEC 27001:2005, ***Information technology -- Security techniques -- Information security management systems – Requirements.***[30] PCI Security Standards Council, ***Payment Card Industry - Data Security Standards (PCI DSS)***[31]. Or any other security standards that best fit an organization (e.g., ***Health Insurance Portability and Accountability Act [HIPPA]) Compliance***.[32]	Define the authorized and unauthorized security postures using Static Analysis tools (listed under Coding in this table)
Requirements Gathering	Security Requirements Elicitation (Misuse and Abuse Cases)	Attack Patterns ('Make the Client Invisible, ' Shell Command Injection – Command Delimiters)[33]

[29] For details of this Standard, go to http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf [September 2009].

[30] For details on this Standard, go to http://www.iso.org/iso/catalogue_detail?csnumber=42103 [September 2009].

[31] For complete information, go to https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml [November 2009]. For an overview, see "Payment Card Industry Security Standards," https://www.pcisecuritystandards.org/pdfs/pcissc_overview.pdf [November 2009].

[32] For information on compliance requirements, see Department of Human Health and Human Services, USA, "HIPPA Security Guidance," http://csrc.nist.gov/publications/nistpubs/800-66-Rev1/SP-800-66-Revision1.pdf and "Health Information Technology," http://www.hhs.gov/ocr/privacy/hipaa/understanding/special/healthit/index.html [September 2009].

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
Architecture & Design	Application Security Architecture & Design (Architecture Risk Analysis, Threat Modeling)	▪ Threat Classification (STRIDE / DREAD)[34] ▪ Vulnerability Scoring (CVSS)[35] ▪ Risk Methodology (OCTAVE)[36] ▪ Modeling Tools (Trike,[37] Practical Threat Analysis,[38] CVSS Spread Sheet, Microsoft Threat Modeling[39])
Coding / Application Development	Application Security Reviews and Inspections	Static Code Analysis Commercial[40] ▪ Fortify 360 Source Code Analyzer[41] ▪ SPI DevInspect[42]

[33] For a good overview of Attack Patterns as well as their generation and usage, see Building Security In, "Attack Patterns," https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/attack.html [September 2009].

[34] For further information, go to http://www.owasp.org/index.php/Threat_Risk_Modeling#Threat_Risk_Modeling [September 2009].

[35] For details, go to www.first.org/cvss/cvss-dhs-12-02-04.pdf [September 2009].

[36] For more information, go to http://www.cert.org/octave/ [September 2009].

[37] For details, go to http://www.net-security.org/article.php?id=807 [September 2009].

[38] For details, go to http://www.ptatechnologies.com/ [September 2009].

[39] For details, go to http://msdn.microsoft.com/en-us/security/dd206731.aspx [September 2009].

[40] For a comprehensive analysis of Static Code Analyzers and how various vendor fair in Gartner's Magic Quadrant, see Joseph Feiman and Neil MacDonald, "Magic Quadrant for Static Application Security Testing," (February 6, 2009) http://www.fortify.com/landing/downloadLanding.jsp?path=/user/GartnerMQ_SAST.pdf [August 2009].

[41] The Source Code Analyzer is one of many components in Fortify 360, which contains a suite of Software Security Assurance solutions. For further information, go to http://www.fortify.com/products/fortify-360/ [September 2009].

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
Coding / Application Development (Continued)	Application Security Reviews and Inspections (Continued)	Static Code Analysis (Continued) ▪ Ounce Core[43] ▪ Micro Focus DevPartner[44] ▪ Coverity Prevent™ Static Analysis[45] Open Source ▪ JLINT[46] ▪ PIXY[47] ▪ FX-COP[48]

[42] For details, go to https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200%5E9564_4000_100__ [September 2009].

[43] Ounce Lab is now owned by IBM. For details on Ounce Core, go to http://www.ouncelabs.com/products/ounce_core [September 2009].

[44] Compuware's DevPartner suite is now owned by Micro Focus. For more details, go to http://www.microfocus.com/products/DevPartner/index.asp [September 2009].

[45] For details, see http://www.coverity.com/products/coverity-prevent.html [September 2009].

[46] For details, go to http://www.jslint.com/lint.html [September 2009].

[47] For further details, see http://pixybox.seclab.tuwien.ac.at/pixy/ and http://www.cs.ucsb.edu/~chris/doc/oakland06_pixy.pdf [September 2009].

[48] For details, go to http://msdn.microsoft.com/en-us/library/bb429476(VS.80).aspx and http://www.owasp.org/index.php/FxCop [September 2009].

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
		Dynamic Vulnerability Analysis Commercial ▪ Hailstorm[49] ▪ SPI Web Inspect[50] ▪ Rational AppScan[51] Open Source ▪ Paros Proxy[52]
Application Testing	Application Security Testing, Risk Identification and Management (Black /White / Gray Box Testing, Penetration Testing, Fuzz Testing)	Commercial[53] ▪ Hailstorm, Appscan, NTOSpider,[54] Security Innovation,[55] WebInspect, DevInspect ▪
		Open Source ▪ Nikto[56] ▪ Odysseus[57]

[49] For further information, go to http://www.cenzic.com/products/cenzic-hailstormEntARC/ [September 2009].

[50] For details, go to www.whitehatinc.com/products/spi_dynamics/webinspect/downloads/WebInspect_DataSheets.pdf [September 2009]. Also go to https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200%5E9570_4000_100__ [September 2009].

[51] For more details, go to http://www-01.ibm.com/software/awdtools/appscan/ [September 2009].

[52] For details, go to http://michaelboman.org/wiki/index.php?title=Paros_Proxy [September 2009].

[53] For a fairly comprehensive list of commercial and Open Source tools, go to http://www.owasp.org/index.php/Appendix_A:_Testing_Tools [September 2009].

[54] For further information, go to http://www.ntobjectives.com/products/ntospider.php [September 2009].

[55] For details, go to http://www.securityinnovation.com/holodeck/index.shtml [September 2009].

[56] For details, go to http://cirt.net/nikto2 [September 2009].

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
Application Testing (Continued)	Application Security Testing, Risk Identification and Management (Black /White / Gray Box Testing, Penetration Testing, Fuzz Testing) (Continued)	Open Source ▪ WebScarab[58] ▪ Paros Proxy ▪ SPIKE[59]
Application Assurance	Application Security Assurance	▪ Authentication (User-id/Password or multifactor, SSO) ○ Authentication Questions ○ HTTP/s Request/Response attributes ○ Hardware/Software Token based Challenge/Response OTP ○ Hardware/Software Token based One-time Passwords (OTP) ○ SMS based OTP ○ USB Tokens/Smart cards (PIN and PKI Certificates) ○ PKI Certificate ○ Biometrics (Fingerprints) ○ USB Token/Smart cards (PKI and Match-on-card Biometrics). ▪ Authorization (ACLS, RBAC,DAC, SoD) ▪ Encryption (Symmetric (AES), Asymmetric(RSA)) ▪ Secure Logging & Auditing (Standard application API

[57] For further information, go to http://www.bindshell.net/tools/odysseus [September 2009].

[58] For further information, go to http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project [September 2009].

[59] See Sean Barnum and Amit Sethi, Cigital, "Attack Pattern Usage," (2006) https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/attack/588-BSI.html [September 2009] and C. C. Michael and Will Radosevich, "Black Box Security Testing Tools," (2009) https://buildsecurityin.us-cert.gov/daisy/bsi/articles/tools/black-box/261-BSI.html [September 2009].

Typical SDLC Activities	"Secure" SDLC (SSDLC) Activities (Best Practices)	SSDLC Security Techniques / Tools (Some Examples)
		support e.g., Log4J)
Application Deployment	Application Security Monitoring (Intrusion Detection System / Intrusion Prevention System)	▪ Tipping Point[60] ▪ Juniper Networks IDP Series[61] ▪ IPS-ETM[62] ▪ ISS-Proventia[63] ▪ Symantec Security[64]

[60] For details, go to http://www.tippingpoint.com/ [September 2009].

[61] For details on all the appliances in the IDP Series, go to http://www.juniper.net/us/en/products-services/security/idp-series/ [September 2009].

[62] For details on the Intrusion Prevention System (IPS) and Enterprise Threat Management (ETM) system, go to http://www.sourcefire.com/solutions/etm/ips [September 2009]. For a report on where key vendors in the IPS space are placed in its Magic Quadrant, see Greg Young and John Pescatore, "Magic Quadrant for Network Intrusion Prevention System Appliances," Gartner RAS Core Research Note G00167303 (April 14, 2009; R3047 041622010). This report is available at http://www.sourcefire.com [September 2009].

[63] For details on various type of intrusion detection and prevention systems in IBM's Internet Security Systems portfolio, go to http://www-935.ibm.com/services/us/index.wss/offerfamily/iss/a1029097 [September 2009].

[64] For details on various type of intrusion detection and prevention systems in Symantec's Security portfolio, go to http://www.symantec.com [September 2009].

Application Security Architecture

Just as "Secure" SDLC requires an iterative integration of security activities in all phases of application development,[65] Security Architecture also requires an iterative process whereby the business, technical, and security requirements of each stakeholder can be mapped to a logical view of the application (to be developed) in terms of policy and standards, security architecture, and risk management.

Writing in the context of Enterprise Security Architecture, Security Expert, Gunnar Peterson, suggests that Security Architecture Lifecycle (as he calls it) should be driven by a well-defined Risk Management Process and that it should comprises the four phases mentioned below,[66] each of which also hold true for our purpose i.e., at the level of Application Security Architecture.

- APPLICATION ARCHITECTURE RISK ANALYSIS ensures that an application's risk exposure is in line with tolerance goals of stakeholders
- APPLICATION SECURITY ARCHITECTURE AND DESIGN ensures the integrity of
    - *Application Security Process* via SDLC, Identity Management, Threat Management, and Vulnerability Management
    - *Application Security Defense In Depth* via Data, Applications, Host, Network Protection
    - *Application Security Metrics* via Audit, Assurance, and Risk Assessment
- APPLICATION IMPLEMENTATION ensures that Risk Management, Security Policy and Standards, and Security Architecture decisions are reflected in the runtime implementation
- APPLICATION OPERATION AND MONITORING measures security metrics in runtime environment

[65] For a lucid explanation of how secure applications can be built, see Theresa Lanowitz, Op cit. pp. 4-7.

[66] For an excellent high-level overview, see Gunnar Peterson, "Security Architecture Blueprint," http://arctecgroup.net/pdf/ArctecSecurityArchitectureBlueprint.pdf [August 2009]. In this article Gunnar also give a practical illustration of an incremental roadmap for adding security to SDLC in the context of What, How, and Who. See pp 4-5. Other high-level generic Security Architecture Frameworks that may be used as a starting point are available at http://en.wikipedia.org/wiki/Enterprise_information_security_architecture [September 2009].

Application Security Service from HCL

This service comprises reliable approaches, methods, and frameworks that our experts leverage for customers who need to meet stringent application security standards.

The starting point in all security engagement is to have a through understanding of our customer's security environment by analyzing current security practices. Table 2 illustrates a typical list of documents that our experts examine before developing a risk mitigation strategy in conformance with our customer's expectations and tolerance goals of stakeholders.

Table 2: Prerequisites for Understanding Client's Security Environment

Customer Documents Required & Examined	Purpose behind examining customer documents is to understand gaps in the current security environment and ensure that:
Enterprise Security Policy Requirements	Proposed architecture and design meets our customer's organizational security standards and objectives
Compliance and Regulatory Requirements	Proposed architecture and design meets Government regulatory compliance standards
Business and Functional (Requirements Classification)	Proposed architecture and design meets application security requirements, and that the security business and functional requirements are categorized according to their criticality
Technical Requirements	Proposed architecture and design meets our customer's platform security requirements
System Architecture Documents	Proposed architecture and design meets various security requirements that are devised in the customer's system architecture specifications
Deployment Diagrams	Enterprise polices are adopted in deployment
Network Architecture	Enterprise polices are adopted in the network architecture
SRS, Detailed Design Documents	Various security requirements are accommodated as part of the customer's requirement specifications and design

Architecture Security Assessment (Threat Modeling)

Our team uses threat modeling tools such as STRIDE/DREAD, CVSS, OCTAVE, and others for identifying threats in our customer's system, planning a mitigation strategy, and for developing robust applications as well as network security architecture and design. As shown in Table 3 our threat modeling comprises a number of important activities and tangible deliverables.

Table 3: Threat Modeling Activities and Deliverables

Activities during Threat Modeling Stage	Purpose	Deliverable
Identify Assets that require Protection (Data Classification)	Classification of assets and data to ascertain the appropriateness of security in place for assets that are considered to be security-critical	Complete threat model for the system. Each threat model will capture the following details: ▪ Report with STRIDE classification and DREAD (1-3) score ▪ List of context diagrams & DFDs to understand the context problem ▪ Attack Trees to identify how the threat is being generated ▪ Network details ▪ Attack surface ▪ Data security ▪ Flow charts for complex scenarios (which can helps in capturing the exception conditions)
Identify Usage Scenarios	Clearly classify scenarios that cannot be supported	
Identify Assumptions	Document any assumptions or guidelines that if violated may compromise application security	
Identify Internal and External Dependencies	Be aware of dependencies on external system, given that if those dependencies are not respected, the security of future systems may be compromised	
Identify Entry and Exit Points (Application Architecture, Network, and Deployment diagrams)	Ensure that all the entry and exit points are listed and appropriately secured (e.g., authentication & authorization are in place for each point)	
Identify Trust Levels, Roles and Access Matrix	Group external entities in logical user groups and implement appropriate security at group level	

The outputs of this phase are as follows:

- Secure Application Architecture
- Technical Assessment Report
    - Observations and Findings
        1. Key policy and process findings
        2. Key technical findings
        3. Compliance gap analysis
    - Threat Model with DREAD Score Report
    - Mitigation Strategies and Best Practices
    - Security Recommendations

Security Design Review and Testing

As illustrated in Table 4, in this phase we:

- Evaluate the application design against customer requirements
- Identify security-critical transactions
- Review the extent to which security has been integrated in the application design and code

Table 4: Security Design Review and Testing Activities and Deliverables

Activities during Design Review and Testing Stage	Purpose is to ensure that:	Deliverable
Security Design Review	System design meets customer requirements	Security Design Review Report
Manual Code Inspection	Requirements, coding guidelines and defined standards have been followed at code level, given that automated analysis tends to report false +ves and false –ves	Security Code Review Report
Manual Security Testing	▪ Security requirements (captured as part of functional/business requirements) have been tested ▪ Appropriateness of authentication and authorization in the system have been checked	Complete Log of reported Bugs, Priorities, Assignments, and Fixes
Static Code Analysis	Secure coding practices have been followed in the code that was implemented	Reports of: ▪ Source Code Scan ▪ Analysis of findings

Activities during Design Review and Testing Stage	Purpose is to ensure that:	Deliverable
		▪ Allocation of identified threat with suggested fixes to development team
Dynamic Vulnerability Analysis (Blackbox Testing)	▪ Application performs various input validations to assure it is free from OWASP's top 10, Sans Top 15 vulnerabilities ▪ Dynamic Analysis performs penetration testing on the application and that the system meets various compliance requirements	Reports of: ▪ Source Code Scan ▪ Analysis of findings ▪ Allocation of identified threat with suggested fixes to development team
Platform Hardening Strategy	Application host platforms have been hardened and that the system is configured for least privilege.	Checklists for: ▪ Web Servers ▪ App Servers ▪ Database Servers ▪ Host Platform ▪ Firewall

Benefits

One of the best measures of a company's quality of service is the feedback from its customers. It is common knowledge today that no business engages the services of an IT company for the sake of its technology but for its know-how in resolving pressing business problems. The Application Security Service from HCL Technologies is all about aligning IT with the business goals of its customers.

Since Brand protection and Compliance to Government Regulatory Standards are among the most important motivators for our customers to enhance the security of their applications, our deliverables are designed to meet very high expectations; in fact in almost all cases our deliverables are put through a series of rigorous security tests by external agencies on behalf of our customers. Designing an application for security is not just about using tools but having prior knowledge and experience to visualize a system's behaviour within specific business domains and designing an application that anticipates intrusions and attacks from all possible angles. That is why our customers seek out our Application Security Service.