Security Architecture Blueprint

By Gunnar Peterson

The purpose of the security architecture blueprint is to bring focus to the key areas of concern for the enterprise, highlighting decision criteria and context for each domain. Since security is a system property it can be difficult for Enterprise Security groups to separate the disparate concerns that exist at different system layers and to understand their role in the system as a whole. This blueprint provides a framework for understanding disparate design and process considerations; to organize architecture and actions toward improving enterprise security.

Security Services

Security services provide confidentiality, integrity, and availability services for the platform. Security services are implemented as protection services, such as authentication and authorization, detection services, such as monitoring and auditing, and response services, such as incident response and forensics. These services have served as the goals and objectives for information security programs for many years, but they do not provide an actionable blueprint as such. This document describes a way to map these security services into an overall enterprise security architecture blueprint.

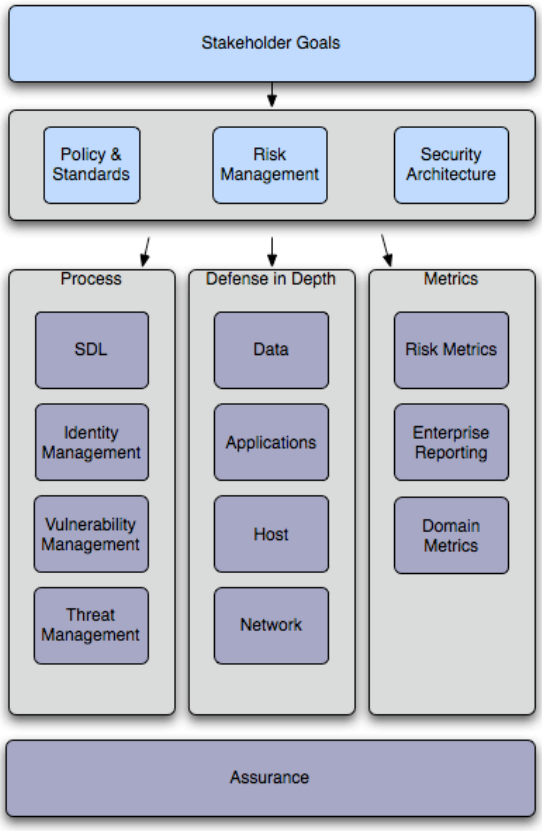

Figure 1: Security Architecture Blueprint

The security architecture blueprint below depicts an approach to map the system's stakeholders' conceptual goals to a logical view fo security, which is set of security policy and standards, security architecture, and risk management domains. The decisions in the logical layer drive the security processes, defense in depth services and security metrics through design time to run time.

Stakeholders

Anyone with a material stake in the systems development and operations, including business users, customers, legal team, and so on. The stakeholder's business and risk goals drive the overall security architecture. While it may initially appear that enterprise security does not have many allies, there may be more than expected. The challenge for enterprise security groups is to identify stakeholders in the enterprise that have a stake in the system's security posture and to educate them about the actual risks and available countermeasures; finally giving the stakeholders' their own, custom metrics, tools and process they can bring to bear on the problem. Specifically, legal teams are generally very interested in understanding risks, so they may be receptive to the OWASP legal project work in defining contractual language for secure software[1]. Business analysts can be trained with methods to specify security concerns in use cases/user stories [2]. Quality assurance teams can be educated on security specific testing tools, such as vulnerability scanners and fuzzers, to identify defects during systems testing. Architects can learn how to design reusable security services that make it simpler for developers to build security into their systems. Once security concerns are embedded in test plans and use cases, and aligned with business goals, the overall burden on defining demand for security services does not solely fall on the information security team, and the development and operations staff has far greater organizational support for the demands of extra initial time and expense required to build a more robust system.

Risk Management: This enterprise security architecture blueprint takes risk management, not "perfect" ivory tower security, as its central organizing concept. Risk is comprised of assets, threats, vulnerabilities, and countermeasures.

$$\text{Risk} = \left(\frac{\text{Threats x Vulnerabilities}}{\text{Countermeasures}} \right) \text{x Assets}$$

Figure 2: Risk equation

[1] OWASP Legal Project
http://www.owasp.org/index.php/Category:OWASP_Legal_Project

[2] "Top Ten Information Security Considerations in Use Case Modeling" by Gunnar Peterson, http://www.arctecgroup.net/secusecase.htm

A risk management centric approach allows for the security architecture to be agile in responding to business needs. Risk is a function of threats exploiting vulnerabilities against assets. The threats and vulnerabilities may be mitigated by deploying countermeasures. The risk management process implements risk assessment to ensure the enterprise's risk exposure is in line with risk tolerance goals. This does not mean that behavior is uniformly risk averse or risk seeking. The system should take on the appropriate level of risk based on business goals.

> *"Don't think, however, that we have lost our taste for risk. We remain prepared to lose $6 billion in a single event, if we have been paid appropriately for assuming that risk. We are not willing, though, to take on even very small exposures at prices that don't reflect our evaluation of loss probabilities…Our behavior here parallels that which we employ in financial markets: Be fearful when others are greedy, and be greedy when others are fearful."*
> -Warren Buffett, 2006 Shareholder Letter[3]

Learning from Warren Buffett, we see that information security should enable, to the extent possible, a business to take the risks it is prepared to take on, by designing and deploying countermeasures that allow for sensible business risk. Additionally, seemingly small exposures should be dealt with if there is a business case. The role of the security architecture is not to steer the business away from risk, but rather to educate their business partners about the risks they are taking and provide countermeasures that enable the business to take as much risk as suits their goals.

Security policy and standards: organizational policies and standards that govern the system's design, deployment, and run time. The security policy describes both what is allowed as well as not allowed in the system. Security standards should be prescriptive guidance for people building and operating systems, and should be backed by reusable services wherever practical. This is very important, it is no longer acceptable for enterprise security to exclusively function as an arbiter; security in the enterprise needs architecture and design advocates, and backing at runtime. Security policy and standards are not end goals in themselves, they need to be backed by a governance model that ensures they are in use, and that it is practically possible to build, deploy, and operate systems based on their intent. *In practice this means that the security architecture must define reusable security services that allow developers to not be security experts yet still build a secure system.*

Security architecture: unifying framework and reusable services that implement policy, standards, and risk management decisions. The security architecture is a strategic framework that allows the development and operations staff to align efforts, in addition the security architecture can drive platform improvements which are not possible to make at a project level. A given software development project may not be able to make a business case to purchase an XML Security Gateway for improved web services security, but at the architecture level, architects can potentially identify several projects that could

[3] Berkshire Hathaway 2006 Shareholder Letter, http://www.berkshirehathaway.com/letters/2006.html

leverage such a reusable service. In this instance the security architecture delivers improved XML/ Web services security, a simplified programming model for developers, and saves development costs, because the wheel is not reinvented multiple times.

Risk management, security policy and standards, and security architecture govern the security processes and defense in depth architecture through design guidance, runtime support, and assurance services. Security metrics are used for decision support for risk management, security policy and standards, and security architecture. The security architecture should have a reference implementation for developers and other IT staff to review what functions the security mechanisms performs, and how they do it.

Security processes

Security processes carry out the intent of the enterprise risk management, security policy and standards, and security architecture. They are broken into discrete domains because they solve very different problems, and require different staffing, support models, and success criteria.

SDL: Security functions as a collaborative design partner in the software development lifecycle (SDL), from requirements, architecture, design, coding, deployment, and withdrawal from service. Security adds value to the software development lifecycle through prescriptive and proscriptive guidance and expertise in building secure software. Security can play a role in all phases of the SDL, but an iterative, phased-based integration of security into the SDL is the wisest path, each additional security process improvement must fit with the overall SDL approach in the enterprise, which vary widely. The DHS Build Security In portal defines process improvements that enterprises can leverage throughout their SDL.[4] Every security process added into the SDL adds incremental expense to the developer's time, so the enterprise security group must wisely choose the artifacts and activities to add in the SDL.

As the overall Security Architecture and related components such as Identity Management evolve over time, these security components and services should be baked into the SDL in a prescriptive way – making it easier for developers to build secure software. The diagram below shows an example approach for iterating through a number of security artifacts and evolving the SDL over time. The goal is to identify reusable services that, over time, can speed development of reliable software, for example building reusable attack patterns that are implemented across a particular set of threats like a set of web attack patterns that can be used for security design in any enterprise web application.

[4] "Build Security In", DHS, https://buildsecurityin.us-cert.gov/daisy/bsi/home.html

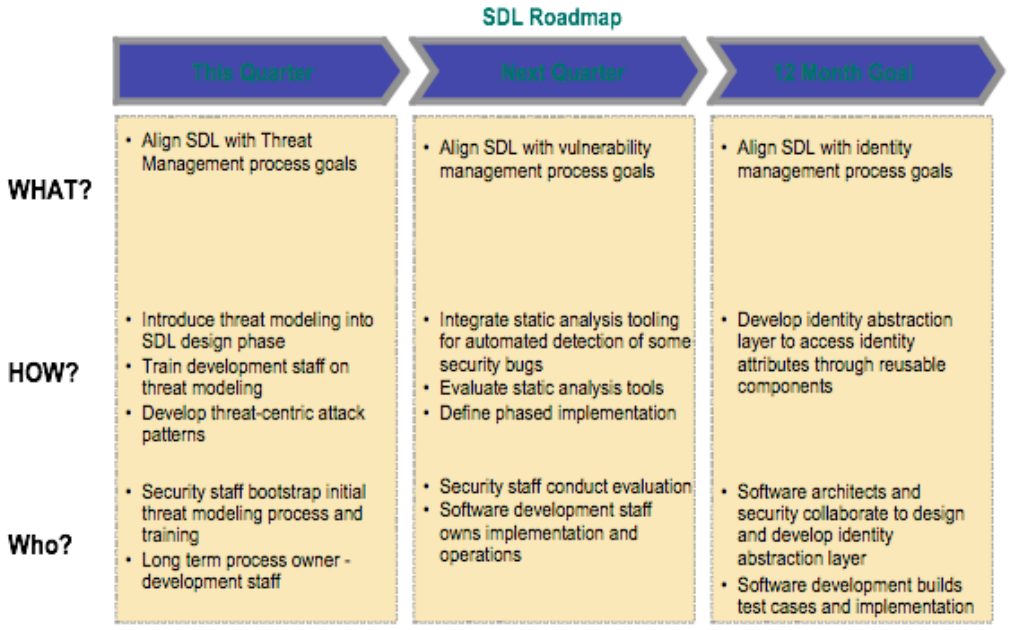

Figure 3: Example roadmap for adding security to the SDL

The above SDL roadmap shows an example incremental improvement roadmap for adding security services into a SDL. It is not a complete SDL such as CMM.

Identity management: deals with the creation, communication, recognition, and usage of identity in the enterprise. Identity management includes provisioning services, directories, multi-factor authentication, federation, and so on. All access control is predicated on identity, a central concern to security architecture, the quality of the system's authentication and authorization cannot be stronger than the identity management process. Identity management architecture is important to identify points of leverage across projects, because identity management components are often not able to support a business case individually. Strategically the enterprise should align investment, architecture, and implementation in the identity space to increase the quality, reusability, and strength of identity. The net benefit is to improve the authentication, authorization, and auditing services for the system as a whole. The utility of the identity management architecture comes through mapping the subject request's claims (or assertions) to policy enforcement decision workflow; and the object's protection model, often in the form of group and/or role membership.

Threat management: deals with the threats to systems such as virus, Trojans, worms, malicious hackers, force majeure, and intentional and unintentional system misuse by insiders or outsiders. Threats differ from vulnerabilities in that threats are the actors that breach or attempt to breach security policies and mechanisms. The security gaps that are exploited by threats are called vulnerabilities. Threat Management tools and processes include: Security Monitoring, Web Application Firewall, Security Incident Management Processes, Security Event Management System, Incident Response Planning Processes, cryptography, and Forensic Analysis Process and Tools. The threat environment is

inherently unpredictable and in large part out of control of the enterprise. Developers can assist the security team in understanding attack vectors and signatures to monitor for, but *it is impossible to predict all threats, meaning that threat management has a large detection and response component.* Monitoring systems and audit services at various levels in the system can identify threats that circumvent expected paths and controls.

Vulnerability management: the set of processes and technologies for discovering, reporting, and mitigating known vulnerabilities. The vulnerabilities may reside at any system layer – database, operating system, servers, and so on; specialized tools probe for known vulnerabilities. It is important to differentiate threat management and vulnerability management. The threat environment contains many unknown mysteries around attacker techniques and goals, attackers will identify currently unknown vulnerabilities (zero day attacks), but there are many known vulnerabilities that the security team can act on, while the threat landscape is inherently less predictable meaning security is reactive to threats and can be generally proactive towards dealing with known vulnerabilities. This has direct implications on staffing, prioritization, and investing in these areas, because vulnerability management has a more predictable lifecycle based on the known quantity of many vulnerabilities.

Defense in depth

Defense in depth is predicated on the notion that every security control is vulnerable somehow, but that if one component fails another control at a separate layer still provides security services to mitigate the damage, for example a Unix web server may be compromised, but if the web server process executes inside a chroot jail which constrains the attack's privileges to launch further attack, then the possibility of a cascading failure is reduced. Each level of the defense in depth stack has its own unique security capabilities and constraints. The core security services - authentication, authorization, and auditing apply at all levels of the defense in depth stack, for example audit logging occurs at network, host, application, and data access levels. The security architect's job is to identify the proper combination of the core security services at each level in the stack to deliver a cohesive security posture that reflects the enterprise's risk management objectives.

Network security: design and operations for security mechanisms for the network. Please note this differs from assuming that "the network is secure" which is the fourth fallacy of distributed computing[5]. Network security mechanisms, such as network firewalls and network intrusion detection devices, are generally a convenient and scalable point to apply security controls and are an important locale for defining chokepoints and zones. Zones define logical and/or physical boundaries around a group of systems, for example the DMZ pattern in web applications. Chokepoints define places to cross boundaries into and out of zones, where special security considerations apply.

[5] "Fallacies of Distributed Computing"
http://en.wikipedia.org/wiki/Fallacies_of_Distributed_Computing

Host security: is concerned with access control on the servers and workstations. Host Intrusion Detection Systems identify host anomalies and security events. Host Integrity Monitoring checks and protects the integrity of the critical files and programs on the host. Baseline Configuration Scanners provide assurance that the systems in use in the field meet the policy and standards at a granular level. These scanners may be automated to support highly distributed and large scale environments. Using the zones and chokepoints defined in the network security architecture, the security architecture defines a baseline configuration for each locale.

Application security: deals with two main concerns: 1) protecting the code and services running on the system, who is connecting to them, and what is output from the programs through a combination of secure coding practices, static analysis, threat modeling, participation in the SDL, application scanning, and fuzzing. 2) delivering reusable application security services such as reusable authentication, authorization, and auditing services enabling developers to build security into their system. Security frequently collaborates with software architects and developers in this area to build security into the system.

Data security: deals with securing access to data and its use, this is a primary concern for the security architecture and works in concert with other domains. Vulnerability management tools conduct specialized scans against database hosts. The SDL defines secure patterns for database integration based on data classification defined in the policy. Database intrusion detection and monitoring provides ongoing intelligence as to the threats against the database. The value in performing detection and monitoring at this layer is that attackers may not traverse the expected path to get to the asset that the security system is trying to protect: **data**. Database, XML documents, transient messages, and other resources are protected by data security mechanisms. Security frequently collaborates with database administrators in this area to drive secure database configuration and operations.

Metrics

Security metrics are a basis for assessing the security posture and trends of the systems. The security metrics data is fed forward to inform future assessments, risk management decisions, and overall security architecture, in an iterative fashion. Metrics provide a way to assess security through qualitative and quantitative analysis. The goal of security metrics is objective measurement that enables decision support regarding risk management for the business without requiring the business to be information security experts to make informed choices. Audit, assurance services, and risk assessment use security metrics for ongoing objective analysis.

Risk metrics: measure the overall assets, and their attendant countermeasures, threats, and vulnerabilities. Since risk metrics are focused on assets, they allow the security architecture to be measured in business terms. Risk metrics inform stakeholders on security posture based on information that is harvested from the security processes, especially vulnerability management and threat management, and the defense in depth stack.

Enterprise reporting: enterprise view of security and risk. Enterprise reports show the states and rates of security, they can show which areas deserve additional focus and where the security services are increasing or decreasing the overall risk exposure. Enterprise reports are rolled up versions of domain metrics and risk metrics. The audience of the enterprise security metrics report will govern what areas are highlighted in the report. The importance of the enterprise security metrics report is in its objective and quantitative nature, which allows for ongoing assessment of security states and rates of change.

Domain specific metrics: domain specific instrumentation of metrics, for example vulnerabilities not remediated, provide granular view of security in a system. These can be aggregated into risk metrics and enterprise reporting formats. Run time metrics, such as alerts and warnings can be used to understand the security events that are visible across a number of systems. The example below shows a sample web application security scorecard that measures the security posture against web application security domains' specific threats and vulnerabilities.

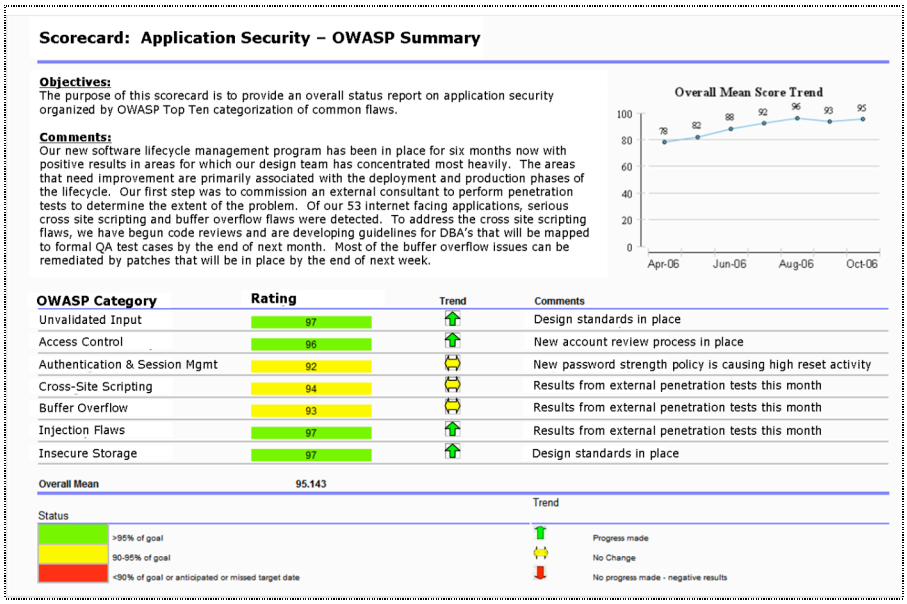

Figure 4: Web Application Security Dashboard (source: Clearpoint Metrics)

Security Metrics is an emerging force in enterprise security. Work needs to be done to create the right metrics for each functional concerns and additional mapping is required to map to individual enterprises. Still security metrics hold great promise because the barriers to get started are low, numbers are an effective enterprise communication tool (Pat Christiansen, my Arctec Group co-founder, says architecture is 50% technical ability and 50% communication), and security metrics represent a quantitative way to analyze the system's security instead of axioms.

Assurance

Assurance is the set of activities that create higher confidence in the system's ability to carry out its design goals even in the face of malicious abuse. These activities are performed by, or on behalf of, an enterprise as tests of the security practices. Activities include penetration testing, code auditing and analysis, and security specific hardware and software controls. The security processes, defense in depth technologies, and metrics are all built on sets of assumptions; assurance activities challenge these assumptions, and *especially the implementations*. Assurance activities should be applied in conjunction with overall risk management goals, for example when the business elects to take on a risky integration with a business partner, some of the exposure can be mitigated by increasing assurance activities on the system. Assurance activities are applied to all of the core security services – protection, detection, and response. The security architecture should identify areas where assurance services can be leveraged across the multiple projects. For example where multi-factor authentication is federated across domains, or where an XML security gateway provides reusable input validation and authentication services for multiple web services.

Putting it all together

Security Architecture Process

Risk management process drives the security architecture and implementation of the overall enterprise security blueprint. The security architecture process is an iterative process that unifies the evolving business, technical, and security domains. The four main phases in the process are: Architecture Risk Assessment, Security Architecture & Design, Implementation, and Operation & Monitoring.

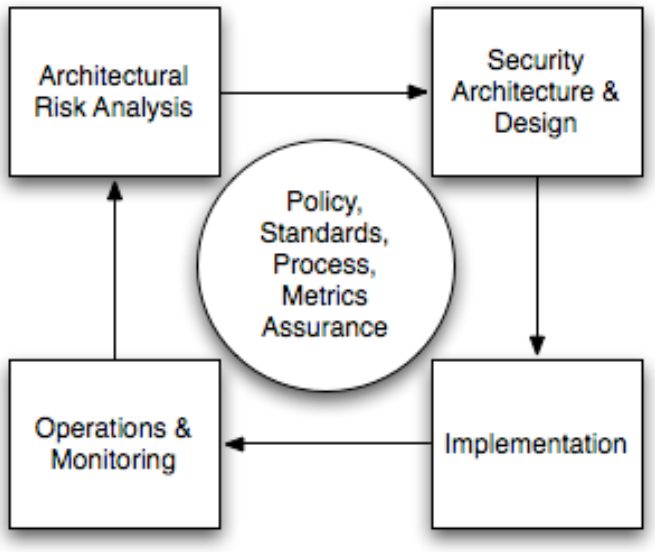

Figure 5: Security Architecture Lifecycle

Architecture Risk Assessment: assesses the business impact to critical business assets, the probability and impact of security threats and vulnerabilities. Since security is a system property, the architectural level is the proper level of abstraction to identify many of the most critical security flaws. The DHS Build Security In paper "Architectural Risk Analysis"[6] defines a method for assessing the application's assets, threats, and vulnerabilities.

Security Architecture and Design: architecture and design of security services that enable business risk exposure targets to be met. The policies and standards, and risk management decisions drive the security architecture and the design of the security processes and defense in depth stack.

Implementation: security processes and services implemented, operational, and managed. Assurance services are targeted at verifiying that the Risk Management, Security Policy and Standards, Security Architecture decisions are reflected in the actual runtime implementation.

Operations and Monitoring: Ongoing processes, such as vulnerability management and threat management, that monitor and manage the operational state as well as the breadth and depth of systems security. Operational and monitoring processes should be instrumented with security metrics to better measure the runtime environment.

Dashboard Reporting

The information security dashboard provides a way to track progress over time across the security architecture and processes. Given the many moving parts in a distributed enterprise, tracking and alignment of efforts is a challenge. The example dashboard below shows one way to roll up across multiple efforts and report on progress at an executive level.

[6] "Architectural Risk Analysis", Hope, Lavenhar, Peterson, https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/architecture/10.html?branch=1&language=1

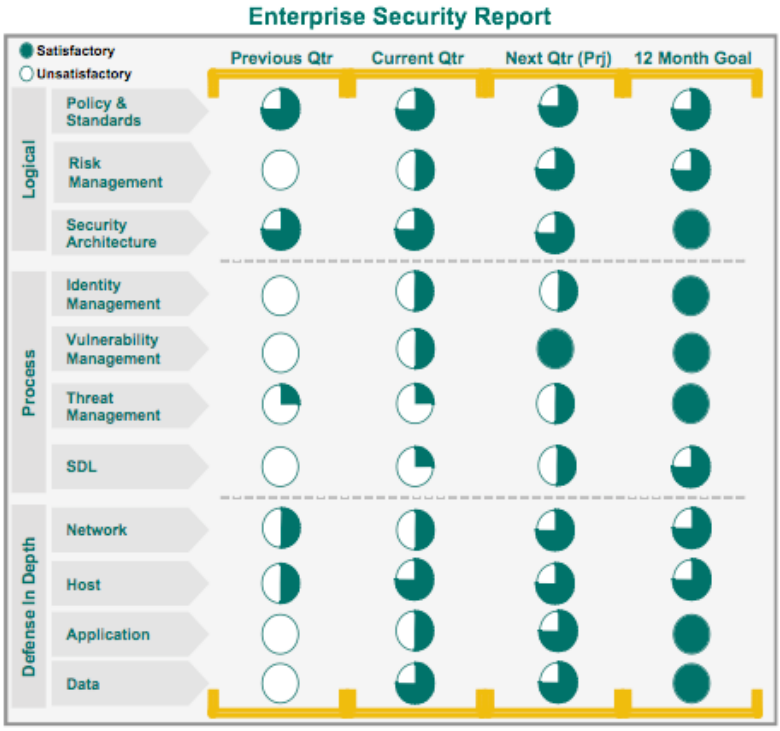

Figure 6: Enterprise Security Executive Report

The security architecture blueprint describes the key decisions, building an Enterprise Security Executive Report helps for senior management to understand the domains, the progress in those domains, and the key investment areas.

Example: Applying the Enterprise Security Architecture Blueprint

This section describes a brief example of activities that enable applying the blueprint in the context of a static analysis project. Static analysis is the process of scanning and analyzing source code to identity security vulnerabilities. As with many security projects these efforts are typically treated as one off projects driven by a single goal, such as compliance, and not ordinarily mapped into a strategic context. Using basic defense in depth architecture we know that static analysis is an important part of the application security defense in depth layer, and that static analysis in the SDL gives the development team the best chance to identify security bugs early in the lifecycle. However, there are many other considerations to deploying static analysis.

- Security policy & standards: define the authorized and unauthorized security postures that the static analysis tools use to build the signatures and patterns to scan for.
- Risk management: it is highly unlikely that all applications will be scanned, and for the ones that are scanned, that all known vulnerabilities will be remediated. Risk management informs static analysis by focusing the scanning, analysis, and remediation work on the assets and efforts that offer the most business value.

- Security architecture & design: static analysis tools are not magic silver bullets. To get real business value the enterprise's authentication, authorization, auditing, and assurance mechanisms should be mapped into the tool to identify that their usage is correct.

At the process level static analysis helps enterprises deal with the following security concerns:

- Identity management: usage of authorized identity mechanisms, identify usage of weak or unauthorized authentication and authorization.
- Threat management: scan source code for signature for known attack patterns
- Vulnerability management: utilize static analysis tool's data set for known vulnerabilities
- SDL: provide developers with integrated, automated security scanning support.

At the defense in depth level, static analysis is focused on

- Network security: static analysis enables the assessor to identify areas where unauthorized or weak network security services are deployed
- Host security: some host security configuration is implicit or explicit in the source code, for example
- Application security: static analysis tools main concern is application security supporting enterprise secure coding best practices and patterns
- Data security: identify secure and insecure data storage practices in the code

Metrics are improved by static analysis through reporting on the application security posture throughout the SDL. Lastly, the overall assurance of the system is increased through automated, repeatable tests that static analysis provides. The assurance is increased when the depth of the static analysis deployment reaches more of the above security concerns, rather than a generic scan that lacks strategic context.

Note

The author wishes to thank Pat Christiansen, James McGovern, Jim Nelson, and Brian Snow for comments and feedback on earlier drafts of this paper. Errors and omissions are the author's.

Author

Gunnar Peterson is a Managing Principal at Arctec Group. He is focused on distributed systems security for large mission critical financial, financial exchanges, healthcare, manufacturing, and insurance systems, as well as emerging start ups Mr. Peterson is an internationally recognized software security expert who is frequently published. He is an Associate Editor for IEEE Security & Privacy Journal on Building Security In, an Associate Editor for Information Security Bulletin, a contributor to the SEI and DHS Build Security In portal on software security, and an in-demand speaker at security conferences.