Quick Start Guide

WhiteHat SECURITY

Five Days to Setting Up an Application Security Program

About this Guide

This guide is intended to be a short, straightforward introductory guide to standing-up or improving an Application Security Program[1]. The intended goal of the AppSec program is to implement measures throughout the code's life-cycle to prevent gaps in the application security policy or the underlying system through flaws in the design, development, deployment, upgrade, or maintenance of the application.

The application security program should effectively manage the security of its application systems, protecting information from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide integrity, confidentiality and availability.

A fundamental component of this improved application security management is the ability to demonstrate acceptable levels of risk based on defined KPIs, including but limited to:

1. The number of vulnerabilities present in an application
2. The time to fix vulnerabilities
3. The remediation rate of vulnerabilities
4. The time vulnerabilities remain open

The application security program deliverables include a holistic view of the state of security for each application, identifying the risks associated with the application and the countermeasures implemented to mitigate those risks, explaining how security is implemented, planning for system downtimes and emergencies, and providing a formal plan to improve the security in one or more of these areas.

Audience

The intended audience of this document is anyone from security engineers, developers, program managers, senior managers or a senior executive. This guide should be considered the start of a comprehensive approach, it is intended to give the basic questions and answers that should be asked by those who are in charge of the application security program in your organization, this includes those responsible for managing the risk of the entire organization.

1 This Guide makes the assumption that network security, physical security, and other relevant security domains have been or are being addressed by other means.

Contents

Day 1

Key Activities

Evaluation

- Dedicate time to understanding the “lay of the land” and the key players, along with their objectives and motivations.
- Spend time with the parts of the organization that you are least familiar with.
- Identify the security mandate as defined by the business/management.
- Identify how IT Operations align with the security mandate.
- Be alert to your existing biases.

Key Questions

Management

Has the business defined an internal security mandate?

Business drivers may originate from legal or industry compliance obligations, internal policy, or customer or partner requirements. All actions and activities in the Application Security program should tie back to specific business obligations.

If there is no business mandate, a case must be made to the business that an application security program is necessary. There are a number of industry statistical reports and benchmarks available that address what other, similar organizations have implemented and why.

What are the current and planned human and financial resources dedicated to Application Security?

If resources are currently dedicated to ensuring the integrity of applications, examine the current allocation of those resources against their outcomes. The goal is to ensure that in a world of limited resources, each activity is measured for actual efficacy.

If there are few or no resources allocated to ensuring the integrity of applications, the case must be made by leveraging data relevant to the organization that can be quantified in financial terms. Building a formal business case is the most effective way to approach the problem. The case must address the needs as they apply to existing business objectives. This is not the time for expounding on the virtues of security. You must translate the business benefit of reallocating financial resources. Simply asking for X number of dollars to reduce Y number of vulnerabilities is typically ineffective.

What are the IT & Business priorities?

IT is both an enabler and supporter of the business. Security is no different in this respect; however, it has the added challenge of needing to be transparent where it can be, and be a part of the ordinary workflow, or be perceived as a roadblock. Once the business priorities have been identified, the application security goals must be aligned with those priorities. If the business priorities are to release new functionality at a rapid pace, then application testing must keep pace; if the priorities are to maintain the user experience over a long period of time, application testing must be rigorous before the user is ever exposed to the application.

Security

What is the overall security budget and the priorities of that budget?
What percentage of that budget is allocated to Application Security?

Most departments will feel under budget to some degree. Examine the overall security budget (including cyber-insurance budget if possible, as it likely falls outside of the security budget) to understand what receives financial priority and how the Application Security program could benefit from those activities. If there is significant spend on security infrastructure, ensure that they are leveraging controls to protect applications -- for example, alerting on the lack or improper use of security headers.

Which of your assets are most frequently attacked?

Not all attacks are equal, and not all attacks are opportunistic. Careful examination of attack frequency and velocity can identify the assets requiring additional testing to ensure resiliency. Often, low(er) priority assets will be targeted for exploitation as they tend not be as resilient to attacks.

What security tools and/or services do you as a company currently own/use?

Query other parts of the security organization to understand what tools and services they already have access to. Software and services are often bundled in contracts but not used as they were not the primary purpose for purchase. If the QA team already has access to security tools that were bundled as part of their QA software acquisition there are opportunities to leverage that existing financial commitment and even allow for native integration.

IT Operations

What are your assets? Include assets you own and assets hosted or owned by third parties.

Unknown or unmanaged assets cannot be protected. It is also difficult to fix issues if the person or team responsible for those assets is unknown. Developing an accurate and update-to-date asset inventory can be challenging, however, for the purpose of this exercise we will be focusing on web-based assets. The narrowed scope can be aggressively targeted and maintained.

How are web assets isolated and distributed throughout the infrastructure?

If an attacker were able to compromise a machine (say a webserver, or a database server) what would prevent them from pivoting and hacking into other nearby machines? Are they physically isolated, logically isolated, or do they share similar credentials, etc?

How frequently do you perform network and server vulnerability scans?

Addressing web application vulnerabilities on a server that never patches its operating system is a waste of resources. Understand how often infrastructure is assessed and patched – this should match or exceed the pace of attack frequency.

What is your tolerance around production safety?

Some environments are more sensitive to production testing, as there is always some likelihood of impact; the goal is to get the likelihood of impact as close to zero as possible. As the likelihood of impact approaches zero, the frequency of testing should be increased. Ultimately, production systems are the primary targets of our adversaries – they should be tested as often as possible; at a minimum, at least as frequently as the application itself is changed or updated.

Engineering Groups (Including QA and Software Development)

According to the software development group's understanding of the current processes, whose responsibility is application security?

Security lives in our corporate cultures and psyche. Developers are ultimately responsible for their code; understand whether they also believe they are responsible for the integrity of that code. Security is not the sole responsibility of either the developer community or the security department. Foster a healthy environment of mutual responsibility and accountability from all stakeholders. Begin by sharing information in a non-aggressive manner.

Where does Security fit into the software development lifecycle?

If a development lifecycle does not exist, the first priority is to demonstrate the business value of having a defined process. Most organizations will have some process in place, even if it is an immature one. The absence of a development process could also prove to be an opportunity to ensure security is built into a process from the start.

Roughly, the steps to building or updating software can be generalized as:

1. Planning / Analysis
2. Design
3. Implementation
4. Testing
5. Maintain
6. Decommission

Let's break down each step and discuss basic security activities that are often considered to reduce risk.

Planning / Analysis: Ensure business analysts and stakeholders have considered and can detail the security needs and risk tolerance of an application. This may reference internal data classification policies to describe the data sensitivity. Threat modeling may also help clarify the potential threat agents who may be motivated to attack the proposed application.

Design: A security architecture review may reveal security design flaws in key areas such as authentication, access control, or separation of concerns, and of course may identify missing categorical security controls.

Implementation: The implementation process normally consists of the coding and development of the overall architecture design formulated in the previous phase. Developers should be receiving continuous security feedback to ensure all security issues are being identified and mitigated in conformance with the organization's risk tolerance.

Testing: Testing should include security tests as well as functional tests. Areas of concentration should be on vulnerabilities that would not have been uncovered during the implementation phase, such as business logic vulnerabilities.

Maintenance: Once the application is promoted to production, continuous testing of security issues should be ongoing throughout the life of the application. As vulnerability vectors and attacks evolve, the application should be tested to ensure defensibility against these new attacks.

How are software defects documented, trended, and prioritized?

The key to adoption of an Application Security program is alignment with and transparency to current workflows. Identify how application defects are documented, trended and prioritized, and plan to insert the application security defects into these existing documents and processes.

Are developers encouraged to develop secure code?

Positive incentive programs foster ownership and accountability for output. Corporate culture varies from each organization; tapping into that culture to offer incentives for producing rugged code will create a natural momentum for finding and fixing security issues. Some cultures will value access to otherwise less accessible personnel, such as lunch with the CTO; others will be motivated by gifts that they might not have purchased for themselves, such as a robotics kit. Take the time to understand the culture of your organization and tap into the inherent desire most people have to be rewarded. Remember that money does not motivate everyone, and can even be demotivating in some cases.

Are abuse and misuse cases part of test scripts?

Test scripts are often developed for the sole purpose of ensuring the application performs the intended functionality. Introduce test scripts that could identify the misuse of intended functionality, such as the ability to execute similar functionality a user should not be able to access.

Is everyone in the organization expected to have general software security knowledge or is there a team/individual tasked with being the "Security Deputy"?

Security deputy programs are a good approach to disseminating application security information to non-security focused departments; they also have the added bonus of fostering a two-way relationship.

Implement a deputy program that:

- is focused on the goals of the application developers and application security testers
- is aimed at achieving results beyond simple security awareness
- can address the points throughout the development cycle where vulnerabilities are introduced – at the time the code is written

Day 2

Key Activities

- Become intimately familiar with what you are meant to protect and at what level.
- Define processes, procedures, and checklists to align assessment strategies to business needs.
- Effectively communicate the introduction and goals of the Application Security assessment program.
- Provide a single point of contact for the program.

Asset Discovery

- Gather Internal, External and Hosted IP ranges.
- Catalogue known domains and subdomains.
- Identify asset meta-data locations. (CMDBs, GRCs, etc.).
- Identify site owners, where those are not already known.
- Gather assessment credentials, including multiple roles for horizontal and vertical testing.
- Identify the rate of application change (e.g. monthly, weekly, etc.…)

Asset Risk Prioritization

- Develop or leverage existing methodology for stack ranking the value of your assets to the business based on impact to confidentiality, integrity and availability (C.I.A.). (See: http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf)

	POTENTIAL IMPACT		
SECURITY OBJECTIVE	**LOW**	**MODERATE**	**HIGH**
Confidentiality Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [44 U.S.C., SEC. 3542]	The unauthorized disclosure of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Integrity Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. [44 U.S.C., SEC. 3542]	The unauthorized modification or destruction of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Availability Ensuring timely and reliable access to and use of information. [44 U.S.C., SEC. 3542]	The disruption of access to or use of information or an information system could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

- Map asset criticality against attacker profiles with use of a GRC* (Governance Risk Management and Compliance) tool if available, or using an information asset register such as the University of Oxford Information Asset Register Tool

  For example:
  - Tier 1 = Targeted Govt./State sponsor.
  - Tier 2 = Hactivism
  - Tier 3 = Random Opportunistic

- Implement ISO 17799: Asset Management or similar standard to improve governance of application assets.

Communication Plan

- Set expectations of assessment program for all interested parties.
- Alert Operations team of upcoming activities.
- Gather written buy-in from application stakeholders for the assessment activities.
- Develop, publish, and maintain comprehensive application security and privacy standards, policies, procedures and guidelines and enforce these in compliance with relevant global regulations and standards.
- Define, document and share application business continuity and incident response plan.[2]

2 Business Continuity Plan Resources: ITIL, COBIT, NIST)

Day 3

Key Activities

- Measure current vulnerability posture.
- Initiate vulnerability testing.
- Triage vulnerabilities.

Vulnerability Assessments

To determine what sort of vulnerability assessment is most appropriate, consider your current status and resources:

Scenario	Appropriate Assessments
Resources and support for immediate unlimited continuous assessments	Perform assessments across all discovered assets
Limited resources or appetite for unlimited continuous assessments	At a minimum frequency of testing should keep or exceed rate of change in asset
Application currently exist in production environment:	Begin dynamic* assessments for these existing applications
Source Code of your application(s) is available on the internet (Freely available, stolen, etc.)	Begin Static Analysis assessments
Business is subject to compliance mandate requiring Static Analysis	Begin Static Analysis assessments
New development project of an application	Begin Static Analysis assessments
Dynamic assessments completed and application security program in continuous improvement cycle	Begin Static Analysis assessments Begin Dynamic Analysis in QA/Staging

Other scenarios to consider Static Analysis as the first assessment type or in parallel with Dynamic Analysis:

- High developer attrition rate
- Known internal bad actors
- Disgruntled current or former employee with access to source code
- Outsourced code

Static Analysis

Static Code Analysis (also known as Source Code Analysis or Static Application Security Testing (SAST)) is usually performed as part of a Code Review (also known as white-box testing) and is carried out at the Implementation phase of a Security Development Lifecycle (SDL). Static Code Analysis commonly refers to the running of Static Code Analysis tools that attempt to highlight possible vulnerabilities within ‘static’ (non-running) source code by using techniques such as Taint Analysis and Data Flow Analysis.

Dynamic Analysis

Dynamic Application Security Testing (DAST), also referred to as “black-box” testing, identifies vulnerabilities in running web applications – testing of the application from the outside in.

Vulnerability Delivery

To deliver valuable vulnerability information to your business, you must:

- Document vulnerability delegation and vulnerability lifecycle process
- Feed issues into existing tracking systems where possible to preserve the existing workflow
- Triage vulnerabilities prior to feeding them into your defect management systems
- Ensure only true positives are fed to development teams
- Track re-testing of vulnerabilities via new incident/ticket or update existing incident/ticket.
- Define which issues are important to the business
- Create a baseline of the issues that are important to the business
- Align vulnerability remediation strategy with loss exposure versus resources available to fix
- Create a knowledge base of common issues and their solutions
- Dedicate resource(s) to developer interactions, including educating developers on security topics
- Publish aggregate metrics internally
- Match or outpace release cycles in detecting and responding to vulnerabilities.

Day 4

Key Activities

- Measure and improve assessment service delivery.

Measured Metrics

- Compare against industry metrics and interdepartmental metrics.
- Compare behaviors to measured metrics to identify which initiatives drive improvement of metrics and security program.

Metric	Definition
Number of Vulnerabilities	The total count of vulnerabilities during the analysis period; valuable as a metric over time.
Time Open	This value represents the number of partial days since the vulnerability was opened as of the specific evaluation date. It only includes open vulnerabilities and not vulnerabilities that were closed. It is computed as the evaluation date less the open date for the vulnerability.
Time-to-Fix	The Time-to-Fix is the number of partial days required to close a vulnerability. It is based on the vulnerabilities that were closed during the analysis period.
Remediation Rate	The Remediation Rate is the ratio of the number of vulnerabilities closed over the number of vulnerabilities opened over a given period of time. A vulnerability is considered closed if it closed during the analysis period. A vulnerability is considered open if it was open at some time during the analysis period. Therefore, vulnerability could be counted as open and closed.
Vulnerability Class Likelihood	Vulnerability Class Likelihood is the percentage of active applications that have at least one open vulnerability in a given vulnerability class over a given period of time. It is determined by counting the number of applications that have at least one open vulnerability in a given vulnerability class over the number of active applications.

Day 5

Key Activities

- Implement compensating controls & mitigation controls
- Remediation Prioritization

Compensating Controls

- Implement compensating controls to limit the likelihood of successful attacks; for example, deploy web application firewalls (WAFs) that inspect all traffic flowing to the web application for common web application attacks.

Mitigating Controls

- Implement mitigating controls to discover and prevent mistakes that may lead to the introduction of vulnerabilities; for example, Control 6 of the CSIS 20 Critical Security Controls – Application Software Security. Build security into the development lifecycle.

Remediation Prioritization

- Implement remediation prioritization driven by financial calculations. Compare the cost of fixing specific vulnerabilities to the expected loss and cost of mitigation resources required in the event of a successful attack.

Conclusion

Setting up an effective application security program does require commitment from all elements of the business, and a clear understanding of what resources need to be protected and what level of risk is acceptable. However, given that information, setting up an application security program need not be confusing, difficult, or complex.

The keys to success involve planning, making key financial decisions, ensuring all roles and responsibilities are clearly assigned and that all stakeholders within the organization know what to expect.

WhiteHat SECURITY

WhiteHat Security, Inc. | 3970 Freedom Circle | Santa Clara, CA 95054 | 1.408.343.8300 | www.whitehatsec.com