Managing Application Security

Insights & observations from a study of AppSec program management

Chris Horn

September 2018

This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via contract number HHSP233201600058C.
About Chris Horn

Product management at Code Dx

Researcher at Secure Decisions
  ▪ An R&D division of Applied Visions
  ▪ Birthplace of Code Dx

Experience
  ▪ 17 years in research, software systems, and new product development
  ▪ Focused on the developing technologies to improve application security
Outline of today’s talk

PART I  About our study
  ▪  Who we interviewed
    –  Types of organizations
    –  Roles of people

PART II  Application security programs
  ▪  Purpose
  ▪  Boundaries
  ▪  Organizational structure

PART III  What directors are paying attention to
  ▪  Goals, questions, metrics, and tools
PART I

About our study
Studied literature and spoke with AppSec practitioners

Literature review

- Read over 75 research papers, technical reports, books, magazine articles, blog posts, and presentations

Interviews

- Interviewed 13 people in application security roles
  - Commercial healthcare insurers, software producers, and military/defense contractors
  - Federal government independent verification & validation groups
  - Plus, one state agency IT group
- Spoke over voice & screen share Web conference, typically for 1 hour
Spoke with two types of AppSec organizations

Internal department

- An application security group operating as a functional department in its parent organization

External reviewer

- An independent verification group, most commonly a legally separate third-party
Interviewed people in one of two roles

<table>
<thead>
<tr>
<th>Director</th>
<th>Analyst</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal department</strong></td>
<td><strong>Analyst</strong></td>
</tr>
<tr>
<td>Responsible for the AppSec program</td>
<td>▪ Work directly with application security testing tools</td>
</tr>
<tr>
<td>▪ Champions secure development practices with software development group</td>
<td>▪ Screen findings for review with development teams</td>
</tr>
<tr>
<td>▪ Establishes the structure, roles, and responsibilities of their team</td>
<td>▪ Serve as security subject matter experts who answer questions that arise during design and development</td>
</tr>
<tr>
<td>▪ Defines testing policies and processes</td>
<td></td>
</tr>
<tr>
<td>▪ Selects testing tools</td>
<td></td>
</tr>
<tr>
<td>▪ Hires analysts</td>
<td></td>
</tr>
<tr>
<td>▪ Manages departmental budget</td>
<td></td>
</tr>
<tr>
<td><strong>External reviewer</strong></td>
<td>Similar responsibilities as internal</td>
</tr>
<tr>
<td>Similar responsibilities as internal</td>
<td></td>
</tr>
<tr>
<td>▪ But typically no software development organization with which to champion secure development practices</td>
<td></td>
</tr>
</tbody>
</table>
PART II
Application Security Programs
Purpose of AppSec is risk management

Achieve the right balance of risk to remediation cost

- AppSec focuses on risks caused by undesirable behaviors of software applications
Risks are uncertain events

A risk is a chance of gaining or losing something of value

Expected value of a risk:

\[ \text{expected value} = \text{cost of the event’s outcome} \times \text{probability of the event} \]

Three ways to decrease risk:

- Decrease the probability of the event (aka threat)
- Decrease the probability of the event’s success
- Decrease the cost/ severity of the outcome
Examples of AppSec risks

Unauthorized disclosure of user credentials through man-in-the-middle attack on **load balancer** using captured TLS session data.

Disclosure of sensitive data due to insecure configuration of **cloud object store**.

Disclosure of database credentials due to remotely exploitable vulnerability in **source code library**.

Loss of database contents due to inadequate sanitization of user input in **application source code**.
Where is the boundary for application security?

Where does AppSec jurisdiction stop?
- First-party source code
- Third-party library
- Third-party application (database, application server, OS)
- Infrastructure (hypervisor, load balancer, router, SAN)
Most organizations leave infrastructure security to IT

First-party source code 42%
Third-party library 42%
Third-party application 42%
Infrastructure 16%*

TREND
AppSec growing responsibility for ensuring security of whole systems

* Safety-critical systems are certified as a whole system

N=13
Service bureaus & champions all around

Internal departments almost exclusively organize AppSec analysts into a central “service bureau”

- Conduct testing & answer questions as a service to development teams
- Working to increase the security literacy of developers
- Aims to develop at least one strong security champion on each development team
- Growing practice of providing security input during the early, architecture-design phases of a project
PART III
What directors are paying attention to
Goal, question, metric (GQM) method

**GOAL**
Conceptual-level goal for a product, process, or resource

**QUESTION**
Constituent part of goal

**METRIC**
Reliable means of assessing or characterizing an answer to each question
Sample GQM tree

GOAL
Reduce expected losses attributable to undesirable behaviors of software applications to an acceptable level

QUESTION
Where are the application vulnerabilities in my software?

METRIC
Number of defects detected by static analysis testing
Defect density (number of defects per SLOC)
Percent of systems under secure development lifecycle (SDL)
Seven top-level questions

1. Where are the application vulnerabilities in my software?
2. Where are my blind spots?
3. How do I communicate & demonstrate AppSec’s value to my management?
4. Are we getting better at building in security over time?
5. Demonstrate compliance with requirements (e.g., internal commitments, external standards such as NIST 800-53, OWASP Top 10 or Application Security Verification Standard (ASVS), and DISA STIGs)
6. How do I make attacks/breaches more difficult for adversary?
7. What is the AppSec team’s input to the broader organization’s acquisition decisions?
Measure the application itself (or a portfolio of apps.)

EXAMPLES
- Number of defects/vulnerabilities [by severity, type, ?]
- Number of findings
- Percent of findings deemed true
- Percent of findings remediated
- Percent of security requirements satisfied
- Defect density (i.e., defects / source lines of code)
- Difficulty to exploit
- Estimated bug bounty value (dollars would pay)
- Total defect age / lifetime
- Number of compromise incidents
- Code change / churn
Measure the things that create an application

EXAMPLES
- Num./pct. of system in SDL
- Time to detect
- Time to remediate (calendar)
- Code coverage of sec. testing
- Percent of staff w/ sec. training
- Scan frequency & duration
- Presence of threat model
- Person hours spent remediating
Measure the organization or policy level

**EXAMPLES**
- Time since last contact with dev. team
- Headcount
- AppSec budget percent of dev. budget
- Security testing tool budget
- Presence of best practice checklist
- Full range of services provided?
Measuring risk is hard

Optimal information security investment strategy research discusses many practically infeasible ways to estimate the expected loss of risk

Acknowledged difficulties include:

- Insufficient data to estimate the probability of most events
- Modeling systems is complex and requires too much information
- Scope of outcome cost estimates (including things like liability, embarrassment, market share and productivity losses, extortion and remediation costs) is daunting
Two workarounds to estimating risk

One organization has analysts fill out custom forms in Atlassian Jira
- 8 probability factors that model threat and vulnerability
- 6 outcome cost factors that model financial and operational effects per finding
- Relies on human mental simulation of how a low-level problem would affect a large system

Another organization* models cost using would-be bug bounty payout
- Avoids expected loss estimation problem altogether
- Models risk associated with chained attacks that move between different micro-services
  - Have a system that records trust relationships between services
  - Can report threat-risks that are “inherited” from other services

Many systems for measuring & tracking metrics

Ad-hoc personal observations

Manual spreadsheets

Reporting features in commercial software security tools

Basic in-house solutions
  - A relational database, sometimes with a Web interface

Elaborate in-house solutions
  - Multiple systems and databases
  - Automated extract transform and load (ETL) jobs
  - One or more data warehouses
  - Third-party governance risk and compliance (GRC) software
You need to decide what to measure

Every organization is different

- Different risk tolerance
- Different technology systems
- Different challenges

Measuring security is a process

RECOMMENDED READING
Contact information

Chris Horn
Researcher, Secure Decisions

chris.horn@securedecisions.com
@chornsec

https://codedx.com
https://securedecisions.com