Building a Product Security Team

The Good, the Bad and the Ugly – Lessons from the Field
Peter Morin

– Over 20yrs in the field
– Principal Cyber Engineer with Forcepoint
– Incident Response
– Worked in the past for the various military and government agencies
– Specialize in protection of critical infrastructure and DFIR
My Story

• I lead the Product Security Team for Forcepoint
• My career with Forcepoint started in December of 2015 at Raytheon
• Learned about what works and what doesn’t
• Want to share my experiences
• Don’t want to paint a bad picture – it is a success story
Forcepoint

- Many moving parts, numerous ways of doing things
- No one organization had a dedicated prod-sec effort
- Different people in the organization who thought they knew what product security should be
- 75+ products supported across the companies
- Raytheon having very structured security given its work with the US government and military
“We already do static code analysis with Coverity...”

“Not sure why we need a product security team, we hire third party testers every few years...”

“Engineering will decide when and how we fix security issues...”
Selling Myself...

• Hardest part of my whole story
• At Raytheon, only a small number of people knew who I was and what my mandate was
• I was invisible to most
• People confused what I was doing with other Raytheon projects (i.e. Raytheon Code Center)
• Most were so tied up in strategizing, building, marketing and selling products
“At times it felt like I could do anything but also felt like I was achieving nothing..."
Geographic Insanity

12 development centers over multiple time zones
Development Scariness

Development Environment
- git
- Gradle
- Subversion
- JIRA
- klocwork
- Jenkins
- Bamboo
- coverity
- PERFORCE

Tech/Platforms
- AngularJS
- Perl
- C++
- Java
- Python
- Oracle
- Kafka
- Docker
- CentOS
Do I Need Product Security?

• Does your company develop and sell software?
  – Packaged products, web applications, embedded software, mobile applications
  – Public consumption or custom built

• How are product security related vulnerabilities managed?

• Is your product tested for security flaws?
Product Security vs. Corporate Security

- **Corporate security**
  - Tasked to defend and protect the enterprise
  - Reports to a CISO, frequently IT department

- **Product security**
  - Focuses on designing security into the products
  - Frequently reports into engineering, CTO
Product Security vs. Corporate Security

• **Very different goals**
  – Can lead to conflicts or compromise of each others goals
  – Two security incidents reported nearly at the same time – one team has to decide, protect the corporation or protect the customer
Product Security vs. Corporate Security

• Very different skills
  – Producing a product involves all the phases of engineering, from requirements gathering, to design, development, and testing
  – Product security engineers involved in writing security requirements, code reviews, vulnerability testing
  – Corp. security involved in network design, deployment of off-the-shelf products, etc.
Product Security vs. Corporate Security

• **Different budgets**
  - Sharing budgets can be very difficult due to different goals
  - Separated teams allow the two budgets to be created and consumed without concern to the others
  - Product security budgets can also factor into larger engineering/dev budgets
Creating a Program

• So you have decided to build a product security team
  – Defined mission
  – Defined program
  – Leadership support/sponsorship
  – Budget
  – Headcount
Building a Culture of Security

• Most important thing you need to start
• Not tools, people, budget, etc.
• **You need an executive sponsor that will champion your cause**
• Someone with an ear to the ground who has everyone’s attention and is willing to put you in front of influential people
Building a Culture of Security

- Get everyone on board
- Present a story
- Show all the shortfalls that others have had
- Use real-world examples
- I met with VPs and directors (i.e. marketing, product management)
Mission Statement

FORCEPOINT™

PSIRT

The mission of the Forcepoint PSIRT is to establish, oversee and carry out plans of action for any vulnerability that potentially threatens the confidentiality, integrity or availability of Forcepoint’s products and services.

Global team assisting customers with the ongoing security of their networks through identification, resolution and prevention of vulnerabilities in Cisco products and industry-wide vulnerabilities.
Defined Program

• What do I offer? What can I offer?
• You need bookends
• Need to set expectations early on
  – Response (i.e. vulnerability management, researchers)
  – Testing (i.e. penetration testing, code review, efficacy)
  – Design, architecture and consulting
  – Training
  – Research and development
  – Create and publish advisories and notices
Defined Program

• Know (and define) your constituency
  – Who are your customers?
  – Do researchers contact you? Media?
  – Don’t forget internal customers (i.e. account reps, product management, engineering teams, QA teams, technical support, etc.)
Reporting Structure

• Where to be in the org chart to institute change?
• Will you report to CTO, CISO, CIO, director?
• Important that you report to someone who shares a similar mission
• Certain reporting structures could be red flags (i.e. IT)
• Can you be bypassed, by who?
• Veto power?
Headcount - Leadership

- Security bugs are not found unless actively sought out — it is hard work to institutionalize this type of scrutiny
- Ensure that engineers are not punished and keep them coming to security for advice
- When the team goes into incident response mode this role should be the calm champion
Headcount – Response Analyst

• Focused on responding to incidents
• Needs to have a very good understanding of the product (SME)
• Someone with good customer interaction skills
• Works well under pressure
• Strong interpersonal and leadership skills
• Strong analytical & evaluative thinking
Headcount - Breaker

• Focused on tearing apart your product
• Can show you if something is resilient to an attack or not.
• You’ll ask “is this secure?” to this person frequently.
• The most entry level breaker may not even be able to write code at all and could still surprise a seasoned software engineer with an exploit.
Headcount - Fixer

• Lives for troubleshooting critical bugs, owning the commit, and identifying the short term fix
• They’re all over the codebase and become that broad knowledge base of how the product works
• This role in a lot of cases may be a dotted line into an engineering team
Headcount – Program Manager

• Optional – may be needed as the team and responsibilities grow
• Managing testing schedules & the SDLC integration – staffing, etc.
• Scheduling external consultant reviews (i.e. contracts)
• Managing the overall success of various programs.
Building a Product Security Team:
Building Blocks of a PSIRT
Know Your Product

- Know the product you are supporting inside out - shadow engineers, use the product, etc.
- If you don’t understand the product, you can’t be expected to find the flaws
- Slows down the response speed
- You will need access to the software – may require a lab to be established
Policies & Processes

- Make sure these are “company” policies and not simply your policies
- Incident response SLAs
- Engineering patch development SLAs
- EOL policy
- Exception process
- Secure coding standards
- How to engage the product security team
- Disclosure policies
Policies & Processes

• Very critical that you document the process
• Make it available
• Invalidate all old processes
• Share this with everyone in the organization
• Ensure they sign-off that they understand it
• Have your executive sponsor enforce it
Testing Your Product

• You will have to determine a number of key items to be successful in testing:
  – Test plan – are you following a process?
  – Scope – what are you testing?
  – Resourcing – do you have the right people?
  – Scheduling (i.e. when, how often) – SDLC integration should determine this
  – Reporting – who is the audience?
Testing Your Product

• Many methods (static code, dynamic/pen, efficacy, etc.)
  – Are you planning on integrating your testing with the build cycle / continuous integration
• Use a standard (OWASP, NIST, OSSTM, etc.)
• Many tools out there (i.e. AppScan, Webinspect, Burp, Fortify, Checkmarx, etc.)
  – Be aware, these are expensive – you will need budget
Security and the SDLC

- Initially you will find testing your product prior to release is all you will be able to handle
- Eventually you want to be integrated into the SDLC
- Will involve being involved in design, requirements, etc.
- **You** need to own this - **not** engineering
- In many cases your customers will want to see a process like this in place
Security and the SDLC - Transparency

Developing Vulnerability Resistant Products

At Forcepoint, we know that secure products and services are essential for organizations to perform and succeed in an ever more challenging threat environment. The Forcepoint Trust Program was created to establish the highest level of trust and confidence with our customers and the security community. Led by our Product Security Team, the Forcepoint Trust Program ensures that security is part of every phase of our software development lifecycle and that transparency is maintained in how we handle security.

Our commitment to the security of our products and services can be found throughout our product lifecycle and includes key elements such as:

- Product Security involvement in the design and architecture of Forcepoint products
- Investment in associated security professionals, results, and tools
- Rapid response to any security issues identified
- Ensuring response time is maintained

To ensure the security, our:
- Security is development
- We partner with developers
- Our products practice
- Our design involves
- We provide tools
- We engage customers
- Most important involved

FORCEPOINT Secure Development Process

1. Define security requirements
2. Design security integration
3. Implement security measures
4. Test security functionality
5. Monitor security performance

Dynamic Application Security Testing (DAST)

Dynamic application security testing involves testing the application during execution. This involves scanning the application for vulnerabilities such as SQL injection, XSS attacks, and other security issues. This type of testing identifies key vulnerabilities that can be exploited to gain unauthorized access to sensitive information.

Information gathering:
- Scanning the application for vulnerabilities
- Analyzing the application code for security issues
- Identifying security weaknesses in the application

Forcepoint's Secure Testing Methodology

- Use a multi-layered approach
- Implement security best practices
- Use automated tools
- Conduct regular security audits
- Ensure compliance with industry standards

Application stacks:
- Use a containerized environment
- Monitor application performance
- Use a DevOps approach

During this phase, the tester will focus on identifying common application security flaws, including:
- SQL injection
- XSS attacks
- Cross-site scripting
- Directory traversal
- Buffer overflows
- Code execution
- Cross-site request forgery
- Cross-site scripting
- Cross-site request forgery (CSRF)
Alliances & Champions

- Work to gain confidence from other groups within your organization
- Offer lunch and learns, etc.
- Organize events such as internal bug bounties
- Make yourself available
- Highlight employees who are helping to win for you
- You cannot be successful without the rest of the organization contributing
When to use a third party tester?

• Can be useful, though expensive, and could be never ending
• We do these for each major release
• Good way to have third party validation
• Engagement should be management by prod-sec
• Have third party prepare an “external” version of the report
• Alternate between third parties year to year.
Measuring Your Success

- Align metrics with the audience
- What is valuable, what is not = relevance
- Executives have limited time, you don’t want to lose their attention in pages of numbers
- Less is often more
- With the S-SDLC in place, trying to show security involvement vs. vulnerabilities can help
Measuring Your Success

- We produce a weekly report for executive management
  - Threat research & intel
  - New incidents
  - New KBAs released
  - New hotfixes and updates
  - Open tickets with status
  - App vs. third party
Perception is Key

• How are you marketed outside the organization?
• How are you perceived by researchers?
• How do you deal with disclosure?
• External advertising of your PSIRT (website, e-mail, etc.)
Perception is Key

• What about researchers?
  – Embargos (fixing before disclosing)
  – Make sure you know what engineers and product management can offer prior to negotiating with researchers
  – You need to know what you can bargain with
Perception is Key

- Communications
  - No response is a response
  - Align with PR and legal – you will need them
  - Ensure they are not putting out statements without you – may contradict your program
  - Does prod-sec have a public presence?
Perception is Key

- Knowledge Base Articles
  - We use KBAs to communicate the status of vulnerabilities
  - Provide detailed info about security issues that involve our products and require an upgrade, fix, or other customer action
  - Critical that the public understand that you know the vulnerability exists and that you are assessing
Tracking...

- Important to track incidents as well as vulnerabilities
- Don’t get hung up with other group’s ticketing system
- You may need to deploy your own system given most don’t adapt as well to IR work
- Also, we wanted a system we owned
- We ended up using RTIR
Tracking...

- We open RTIR ticket:
  - Initial incident report
  - All reporter info gets logged here
  - Track progress of incident

- We open a JIRA ticket if:
  - We need to escalate to engineering
  - We require a fix to be developed
  - Interact with engineering
Use Industry Standards

• Using industry standards will allow you to set a common structure throughout the organization
• Scoring vulnerabilities
• CVE, CWE, CVSS – will make it easy for others to understand this common “language”
• Other standards such as CERT Coding standards, SafeCode
Community Involvement

• More you are involved with the community, the more the public will be aware of your company’s prod-sec efforts
• Ensure your team is able to frequent events
• Promote public speaking in your team
Bug Bounty

• Start small – internal bug bounty
• Work with a trusted third party
• Some do it themselves...
• What you are prepared to offer?
• Remember bounties also bring a spotlight onto your organization
• Some people make a living off this
• Noticed researchers who come to us ask whether we have rewards
In Closing

- **Best practices for success – in my humble opinion**
  - Operate with as much transparency as possible – will go along way with researchers and customers
  - Know your product inside and out
  - Find a good executive sponsor
  - Be persistent with your organization – show them the value of what you’re trying to do!
  - Be an active part of the community
Questions? Comments?

Peter Morin
petermorin123@gmail.com
Twitter: @petermorin123
http://www.petermorin.com