THE ART OF SECURING

100 PRODUCTS

Nir Valtman

@ValtmaNir
I work for as the Application Security

1st time speaking publicly, except at

Neither of my previous startups succeeded!
But at least I invented few open source tools.

Lastly… I’m not a fan of the buzzword ”Cyber”!

The trademark specified above is a trademark or a registered trademark of its respective company. This slide is intended for informational purposes only and does not represent any endorsement by this company.
Why Does This Talk Matter?

Provides Practical Approaches To Secure 100 Products

It's A Big Challenge To Secure 100 Products

You May Need To Secure Many Products

Someone Will Pay You Lots Of $$$ To Do It

You Like Expensive Stuff!
Agenda

Plan

Reality
Meet The Application Security Lead!

- Accountable for Product Security
  - Cloud-based, self-hosted or installed on customers’ premise
  - Part of the products are regulated
- Needs to keep the company out of the news
- Got executive leadership to support him

* Avatar generated on avatarmaker.com
The Daily Challenges

Need to secure a *single* high-risk product. Who’s involved?

Legal
IT Services
CISO
Hardware Solutions
Software R&D
Product Management
Solution Management
Professional Services
Internal Audit
Mapping The Business Owners

Product #1
✓ Software R&D  
✓ CISO  
✓ Legal  
✓ Product Management  
✓ Internal Audit

Product #2
✓ Software R&D  
✓ CISO  
✓ Legal  
✓ Product Management  
✓ Solution Management  
✓ Hardware Solutions

Product #100
✓ Software R&D  
✓ IT  
✓ CISO  
✓ Legal  
✓ Product Management  
✓ Solution Management  
✓ Professional Services

7
Will I Finish This Mapping Soon?

Start Date

Done!
WE NEED TECHNOLOGY!
RESOURCE DIVERSITY & LIMITATIONS
The slide displays a landscape view of infrastructure automation companies, categorized into sections such as BI Monitoring, Collaboration, Continuous Integration, Build, Cloud IAAS, Database, Deployment, Microservices, Release Management, Config Management, Config Provisioning, Networking, and Testing. Each category contains logos of companies related to that area. The slide also includes a note that the trademarks specified above are trademarks or registered trademarks of their respective owners and that this slide is intended for informational purposes only and does not represent any endorsement.
<table>
<thead>
<tr>
<th>Diverse Application Security Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Application Security Testing (SAST)</strong></td>
</tr>
<tr>
<td><strong>Dynamic Application Security Testing (DAST)</strong></td>
</tr>
<tr>
<td><strong>Interactive Application Security Testing (IAST)</strong></td>
</tr>
<tr>
<td><strong>Software Composition Analysis</strong></td>
</tr>
<tr>
<td><strong>&amp; More...</strong></td>
</tr>
</tbody>
</table>

- **Runtime AST**
- **Container Security**
- **Code Obfuscation**

The trademarks specified above are trademarks or registered trademarks of their respective owners. This slide is intended for informational purposes only and does not represent any endorsement.
What is the Application Security labor percentage of the engineering labor?

Labor Limitations

1%-2% of engineering org size
APPLICATION SECURITY MATURITY PROGRAM

Maturity is knowing when and where to be immature
Governance - Easy To Say, Difficult To Control

- Best Practice...

Develop an S-SDLC
- Provide Technology-Specific Training
- Define Risk Management & Risk Acceptance Process

Enforce the S-SDLC
- Map, Track & Drive Towards Completion Of Trainings
- Get Executives To Sign On A Security Risk
Threat Assessment
Documenting risks in agile development lifecycle consumes much resources

Security Requirements
Should app security be involved in ALL requirements sessions?

Security Architecture
Providing best practices for various product types
Verification - Roadblocks Ahead!

**Design Review**
- Get a design diagram from engineering teams... *lots* of teams!!!
- Working with *many* smart engineering people – they know everything!

**Code Review**
- Utilizing automation is great if *ALL* bug tracking, code repo, and build systems are centralized
- Scaling automation for 100 products is nearly impossible (technology & labor wise)
- Building a central security library is a waste of time if technologies are vary!

**Security Testing**
- $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
Deployment - Only Sounds Easy

“Shadow”-operated IRT

Work w/ every engineering team to QA hardening

Corporate IRT

If I define & It doesn’t work, they’re responsible
Perspective On 1 Of 100 Products

Additional Considerations

- Number of items per development lifecycle stage
  - E.g. pending QA, not started, in dev, etc.
- Average time to mitigate a vulnerability
- Prioritized list of outstanding Epics/US/Bugs
Perspective On 100 Products
Perspective On 100 Products

Top 15 Product with Security Issues in past 3 months Category=Multiple Categories and Multiple Products

Product names are sanitized
DON’T REINVENT THE WHEEL, JUST REALIGN IT
(Anthony J. D’Angelo)
NCR’s App Sec Team’s Specialties

- Application Security Architect
- Application Security Engineer
- Application Security Program Manager
- Application Security Risk & Compliance Manager
<table>
<thead>
<tr>
<th>OpenSAMM</th>
<th>Security Architecture</th>
<th>Program Management</th>
<th>Risk Management &amp; Compliance</th>
<th>Application Security Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Governance</td>
<td>Strategy &amp; Metrics</td>
<td></td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Policy &amp; Compliance</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Education &amp; Guidance</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Construction</td>
<td>Threat Assessment</td>
<td>V</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Security Requirements</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Secure Architecture</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verification</td>
<td>Design Review</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code Review</td>
<td>V</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Security Testing</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Deployment</td>
<td>Vulnerability Mgmt</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Environment Hardening</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operational Enablement</td>
<td></td>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>
Prioritizing Security

Product Type
Internal
Regulated
Internal-Facing & Regulated
Internal
Internet-Facing

Strategy
Investments & Commitments

Financial Impact
$1M-$5M
>$5M
<$500K
$500K-$1M
Budgeting Labor Correctly - The Formula

<table>
<thead>
<tr>
<th>Product Type</th>
<th>% Of R&amp;D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet-Facing &amp; Regulated</td>
<td>2%</td>
</tr>
<tr>
<td>Internet-Facing</td>
<td>1%</td>
</tr>
<tr>
<td>Internal &amp; Regulated</td>
<td>1%</td>
</tr>
<tr>
<td>Internal</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

R&D Labor Count

<table>
<thead>
<tr>
<th>Product Type</th>
<th>% Of AppSec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Manager</td>
<td>20%</td>
</tr>
<tr>
<td>Risk &amp; Compliance</td>
<td>10%</td>
</tr>
<tr>
<td>Architecture</td>
<td>23%</td>
</tr>
<tr>
<td>Engineering</td>
<td>47%</td>
</tr>
</tbody>
</table>

Example

An *Internet-facing & regulated* product suite that is developed by an org size of 1000 *employees* needs: 2% X 1000 = 20 *App Sec Team Members*, consisting of 4 PM, 2 R&C, 4.6 Architects and 9.4 Engineers
A Lesson Learned

Even with an aggressive strategy, hiring app sec people is a REAL bottleneck!
A Satellite Program

Give a poor man a fish and you feed him for a day. Teach him to fish and you give him an occupation that will feed him for a lifetime.”

(Chinese proverb.)
## A Satellite Program Example

<table>
<thead>
<tr>
<th>Online Training</th>
<th>Yellow Belt</th>
<th>Green Belt</th>
<th>Brown Belt</th>
<th>Black Belt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foundation app sec classes</td>
<td></td>
<td>Advanced classes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instructor-led or conferences participation</td>
<td></td>
<td>Various advanced topics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Special Interest</td>
<td></td>
<td>PII, GDPR, PCI, FFIEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static/Dynamic/Interactive Security Analysis</td>
<td>On-boarding</td>
<td>Tool/Process improvement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced</td>
<td></td>
<td>Threat modeling</td>
<td>Standards review, reusable IP</td>
<td></td>
</tr>
</tbody>
</table>
Measuring Effectiveness!

**Ongoing**

- Escalations asking for security resources by the engineering teams are good!
- Status reports must be balanced
  - Neither too Green nor Red
Measuring Effectiveness!

Year Over Year

- Overall Application Security Maturity rank increases
- Decreased number of security vulnerability reporting per X (you to define) lines of code
  - Engineers will always make mistakes
  - Use 3rd parties to assess it
Scaling Out Team’s Capabilities

Security Questionnaire For Engaging An App Sec Architect

10 Yes/No Questions
## Security Questionnaire For Engaging An App Sec Architect

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Data Classified?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Do You Follow The S-SDLC?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Data Encryption?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Handle Sensitive Data Related To PII/PCI?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Security Automation Integrated Into Pipeline?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Consumer-Facing Mobile App?</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>
Scaling Up Security

Application Security Must Fit Into Any Pipeline

DEV
- Plan
- Code
- Build
- Test

OPS
- Release
- Deploy
- Operate

Continuous Delivery
Continuous Integration
Agile Development
Containerized Application Security Pipeline Example

Plan
- Security Requirements
- Security Design
- Security Architecture

Code
- Code Review

Build
- SAST Scanning
- SCA Scanning
- Container Build Security Scan
- Push To Registry
- Container Registry Security Scan

Test
- Execute Tests
- Container Runtime Protection
- IAST Scanning
- Penetration Testing (incremental)

Release/Deploy/Operate
- Deploy
- Container Runtime Protection
- DAST/IAST/RASP
Scaling Up Security Using Release Automation

Static App Sec Testing
Interactive App Sec Testing (IAST)
Binary Signing
Code Obfuscation

Dynamic App Sec Testing (DAST)
Vulnerability Scanning
Runtime App Self Protection (RASP)
Scaling Up Security When Lacking Automation

Identify Quick Wins

- Code Obfuscation
- Static App Sec Testing
- Penetration Tests
- Manual Code Review
- Dynamic App Sec Testing (DAST)
- Binary Signing
- Vulnerability Scanning

Even A Long-Term Plan Is A Viable Plan
Additional Tips

- Securing 100 products takes years.
  - Start by investing 80% of the resources in 20% of the products.

- Reflect your success!
  - Trending charts of app sec metrics
  - Integration of tools into the build process
  - Share product certifications completion
  - Speak at Secure DevOps Summit 😊
Apply What You Have Learned Today

- Next week you should:
  - Generate security engagement questionnaire (10 Yes/No Qs)
  - Identify security tool implementation quick wins

- In the first three months following this presentation you should:
  - Establish an application security maturity program
  - Develop a product security strategy based on
    - Company’s strategy
    - Development methodologies & pipelining tools
    - Product Types

- Within six months you should:
  - Hopefully map all products & owners 😊
  - Start executing the strategy
THANK YOU

Nir Valtman
@ValtmaNir