Go Purple!

Augmenting Application Security Program by Adapting Purple Team Strategy

Trupti Shiralkar
About Me

“Life is either a daring adventure or nothing at all” ~ Helen Keller
Agenda

Challenges

• Negative stereotype around security
• Immature security processes
• Incompatible security tools & technology

Goals

• Reduce risk and Improve Return On Security Investment (ROSI)
• Influence positive security using a holistic and realistic approach
• Weave security processes in business & SDLC using purple teams approach

Solution Approach

• Security Training
• Full Stack Security Assessments
• Seamless integration of security tools & assessment methodology
• Metrics Generation
Challenges

“Perception is the co-pilot to Reality” ~ Carla Harris

- Incompatible, outdated security tools
  - Security is expensive
  - Security Professionals are rude & unfriendly

- Immature Security Processes
  - Security is not essential
  - Security is complex

- Negative Stereotype around Security
  - There is no security issue if there is no exploit
  - Security tools break the application
  - Security is hard

- Security is a hindrance to innovation
  - Security is compliance/certification
  - Security is bottleneck Slows down deployment

- Security is expensive
  - The check point approach is considered a watchdog approach

- Security professionals are pessimists
  - Security scanning is time consuming

- Security is complex
  - Security is not essential
Application Security Program Elements

“If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.” ~ Bruce Schneier
People
Blue Team vs Red team
Economics of fixing Security Bugs

Cost of Software Bug (Source: IBM System Sciences Institute)

- $100 for Requirement Gathering phase
- $1500 In QA phase
- $10,000 In production phase

Cost of Software Bug (Source: NIST, 2003)
Blue Team (Optimist) vs Red Team (Paranoid)
Purple team (Realist)
Blue Team (Optimist) vs Red team (Paranoid)
Purple Team (Realist)

Prepare for the worst, hope for the best & make necessary adjustments
"Security is a chain; it's only as secure as the weakest link." "Security is a process, not a product." ~ Bruce Schneier
Security within SDLC

Waterfall SDLC
- Requirement
- Design
- Development
- Testing
- Deployment
- Threat Modeling
- Design Architecture Review
- Static code review
- Dynamic App Testing
- Penetration Test

Agile SDLC
- Start Requirement
- Development 1
- Integrate & Test 1
- Development 2
- Integrate & Test 2
- Release & Feedback Review
- Deployment
Advantages

• Streamlined engagement model
• Works well for Waterfall and Agile SDLC with longer sprints (major releases)
• Checkpoint approach blocks insecure deployments
• Vulnerability statistics and metrics are defined
• Integrates well with risk management processes

Challenges

• Slows down the deployment process
• More focus is on assessment which establishes a baseline for security and therefore not a holistic approach
• Security is not weaved in the business process
• Cannot be completely integrated with DevOps or Agile SDLC per release
Security within CI/CD

- Integrated Static code scan
- Code Commit
- Build Trigger
- Automated Security Configuration Checks
- Testing
- Publish to repository
- Automated Security Testing
- Deploy
- Monitoring
- Other Security Testing
Secure DevOps Approach

**Advantages**
- Security is built-in and no longer a road block
- Security automation increases speed of delivery/deployment
- Faster resolution of security issues
- DevOps engineers gain knowledge of security

**Challenges**
- Initial push back from DevOps teams for security testing
- Selection of incorrect security tools will result in false positives
- Learning curve for security tools & product security issues
- Not everything can be automated (human interaction is required, manual pen test, security requirement, design review)
Purple Team Approach

- Developer Security Training
- Requirement
  - Threat Modeling
  - Security & business Requirement mapping
- Design
  - Design, Architecture Review
  - Security Controls+
  - Exploitation Techniques
- Development
  - Static code scan code review
- Testing
  - Penetration Test
  - Dynamic Web App Testing
- Deployment
Key Aspects

Security Principles
- Confidentiality
- Availability
- Integrity
- Identification
- Authentication
- Authorization
- Accountability
- Auditing
- Nonrepudiation
- Privacy

Security Requirements
Example Authentication:
Session ids should be sufficiently long, random, unique across the correct, active session base

Security Controls
Session ID=digest(random number)

Exploitation Techniques
Entropy Analysis + Brute Force
Foundations for a Positive Security Process

- Invest in your stakeholders to build a foundation with security knowledge
- Empower all stakeholders with appropriate security tools & technology
- Map business requirements -> security requirements -> controls -> exploitation techniques
- **Speak the same language as the SDLC**
- Create win-win situations for all involved parties without compromising security
Application Security Program
(the Purple way!)

- Create Security plan
- Full stack assessment
- Report findings
- Track remediation
- Metrics
- Conduct security training sessions
- Define security requirement and controls
- Add (new & existing) apps in inventory
- Classify apps based on risk rating
- Create Security plan
- Full stack assessment
Application Inventory

Acquire a list of all applications

• Home grown applications
• Applications acquired through acquisitions and mergers
• Third party applications/components

Create a profile for each application

• Description and functionality
• External & internal interfaces
• Technology stack
• Components/ micro-services
• Business criticality & risk
• Highest data classification
Engagement

Training
• Conduct Security training sessions
• Internal training vs external training
• Teach attacker’s mindset and exploitation techniques

Product Onboarding
• Map business requirements with security requirements
• Tie security requirements with controls and respective exploitation techniques
• Implement controls/security services
Unrestricted File Upload

Security Controls
- Allow only authorized users
- Check File size
- Check file extension
- Use white-list approach
- Run AV on the server

Exploitation Techniques
- Bypass the function level AC check
- Manipulate the magic number & extension
- Insert injection attack strings as part of the content in the file
Blind XSS

Security Controls
- Input Validation
- Output Encoding

XSS Injections
- ";!--"<XSS>=&{()}
- <A HREF="http://0x42.0x0000066.0x7.0x93/">XSS</A>
- [1].find(alert)
- %252E%252E%252F
Replay Attack
Create Security Plan

- Educate stakeholders on security assessment types
- Determine what security checks can be automated
- Identify and schedule necessary assessment types
- Determine frequency of assessment types
- Work with release managers to assign priorities
- Define success criteria and remediation SLA
## Full Stack Assessment

<table>
<thead>
<tr>
<th>Layer</th>
<th>Assessments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire App</td>
<td>Threat Modeling</td>
</tr>
<tr>
<td></td>
<td>Architecture, design review</td>
</tr>
<tr>
<td>Source Code</td>
<td>Automated static code scan</td>
</tr>
<tr>
<td></td>
<td>Manual code review</td>
</tr>
<tr>
<td>Open Source libraries, components, frameworks</td>
<td>Open source vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>Operational dependencies</td>
</tr>
<tr>
<td></td>
<td>Licensing risk</td>
</tr>
<tr>
<td>Web app (front end, backend, DB, microservices)</td>
<td>Dynamic app scan</td>
</tr>
<tr>
<td></td>
<td>Internal app pen test</td>
</tr>
<tr>
<td></td>
<td>External app pen test</td>
</tr>
<tr>
<td>Container layer</td>
<td>Hardening check / benchmark</td>
</tr>
<tr>
<td>Host VM</td>
<td>Hardening check</td>
</tr>
<tr>
<td></td>
<td>Missing patches/vulnerability assessment</td>
</tr>
<tr>
<td>Network</td>
<td>Vulnerability assessment</td>
</tr>
<tr>
<td></td>
<td>Internal network pen test</td>
</tr>
<tr>
<td></td>
<td>External network pen test</td>
</tr>
<tr>
<td>Crypto</td>
<td>Crypto specific testing</td>
</tr>
<tr>
<td>HSM</td>
<td>Hardware security testing</td>
</tr>
</tbody>
</table>
Reporting

- Use same bug tracking mechanism or tool for recording security issues or integrate security bug tracking tool with the one used by the product teams
- Educate stakeholders on subjective nature of risk

\[ \text{Risk} = \text{Likelihood} \times \text{Impact} \times \text{Threat Agent} \]

- Automate vulnerability statistics (critical, high, medium, low, vulnerability age)
How do you communicate a vulnerability?

“Security tickets may hurt, but breaches kill the product”
Remediation Consulting

“Vulnerability is the birthplace of innovation, creativity and change.” ~ Brene Brown
Metrics

“If you can't measure it, you can't improve it” ~ Peter Drucker

Chief Security Officer
- Overall Enterprise Risk
- **Return on Security Investment (ROSI)**
- Compliance, Benchmark

Director of Security
- Security Program Initiative
- Security Posture/ Trend Analysis
- % change Audit, compliance and finding

Security Manager
- Security process improvement
- % change in threats once the controls are in place
- % policy violation

Security Team
- Vulnerability statistics (critical, high, medium, low, vulnerability age)
- Security controls
- Top 10 app sec threats in your organization
Conclusion

“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.”
— Sun Tzu, The Art of War

• A holistic security approach is necessary to minimizes gaps
• Security is effective and efficient when weaved in the business and not just in the SDLC
• Adapt Security as a Service model
• Security communities need to work towards changing the negative stereotype against security
Question?

Contact: s.trupts@gmail.com
Thank You