# "I'm a Software Developer. What Do You Mean I'm on the Blue Team?" What we can learn in a red/blue world

Aaron Poffenberger

2016-03-31 Thu

# Outline

1. Introduction
2. Red Team/Blue Team Overview
3. Software Development Overview
4. Now We See the Impedance Mismatch Inherent in the System
5. Security Among Software Developers
6. Where To From Here?
7. Conclusion
8. Notes on Certifications
9. Resources

# Introduction – Background

- Software developer
  - 30+ years
  - 17+ years professionally
  - Security software developer
  - Design and implement secure APIs
  - Consulting
- IT Background
  - Boeing
  - ISP (dial-up land)
  - Consulting
  - DevOps

- Software Development Experience
  - PentaSafe Technologies
  - NetIQ
  - TheAnimenetwork.com
  - BRS Labs
  - Giant Gray
- InfoSec
  - Software vulnerability assessment
  - Auditing
  - CISSP 2005+
  - US Army

# Introduction – Other

- Founding member, Houston dc713
- Founding member, Houston Area Hackers Anonymous
- OpenBSD user
- Amateur radio enthusiast
- Electronics hobbyist

# Introduction – Survey

- Who here is Software Development?

# Introduction – Survey

- Who here is Software Development?
- Who here is on the Red Team?

# Introduction – Survey

- Who here is Software Development?
- Who here is on the Red Team?
- Who here is on the Blue Team?

# Introduction – Genesis of this Talk

- What do the following have in common?

# Introduction – Genesis of this Talk

- What do the following have in common?
    - telnet

# Introduction – Genesis of this Talk

- What do the following have in common?
    - telnet
    - ssh

# Introduction – Genesis of this Talk

- What do the following have in common?
  - telnet
  - ssh
  - Shellshock

# Introduction – Genesis of this Talk

- What do the following have in common?
  - telnet
  - ssh
  - Shellshock
  - chroot

# Introduction – Genesis of this Talk

- What do the following have in common?
    - telnet
    - ssh
    - Shellshock
    - chroot
    - hmac

# Introduction – Genesis of this Talk

- What do the following have in common?
  - telnet
  - ssh
  - Shellshock
  - chroot
  - hmac
  - HTTP BasicAuth

# Introduction – Genesis of this Talk

- What do the following have in common?
  - telnet
  - ssh
  - Shellshock
  - chroot
  - hmac
  - HTTP BasicAuth
  - npm

# Introduction – Genesis of this Talk

- What do the following have in common?
    - telnet
    - ssh
    - Shellshock
    - chroot
    - hmac
    - HTTP BasicAuth
    - npm
    - Apple App Store

# Introduction – Genesis of this Talk

- What do the following have in common?
    - telnet
    - ssh
    - Shellshock
    - chroot
    - hmac
    - HTTP BasicAuth
    - npm
    - Apple App Store
- A few examples of design and implementation decisions software developers have made that have had widespread affect (for better or worse) on organizational security

# Introduction – Genesis of this Talk

- What do the following have in common?
  - telnet
  - ssh
  - Shellshock
  - chroot
  - hmac
  - HTTP BasicAuth
  - npm
  - Apple App Store
- A few examples of design and implementation decisions software developers have made that have had widespread affect (for better or worse) on organizational security
- And yet. . . where are developers in the security life cycle?

# Introduction – Survey Part II

- How many people in Software Development raised their hands when I asked "Who is a member of the Red or Blue Teams?"

# Introduction – Survey Part II

- How many people in Software Development raised their hands when I asked "Who is a member of the Red or Blue Teams?"
- How many Red or Blue Team members have conducted security exercises where Software Development was part of the security team or the stakeholders?

# Introduction – Survey Part II

- How many people in Software Development raised their hands when I asked "Who is a member of the Red or Blue Teams?"
- How many Red or Blue Team members have conducted security exercises where Software Development was part of the security team or the stakeholders?
- How many Red or Blue Team members have conducted security exercises on behalf of a software vendor?

# Red Team/Blue Team Overview I

- Red Team: attack
  - Emulate real-world adversaries
  - Might be permanent team (large companies)
  - Often consulting service to (medium-sized companies)
  - Small companies may have no idea they have a Red Team (*i.e.*, adversaries)
- Blue Team: defend
  - Defend against real-world adversaries
  - Can be members of specific defense or response teams (large companies)
  - May be working sysadmin team (small- to medium-sized companies)
  - May be fully- or partially-outsourced IT services
  - May be a lone sysadmin

# Red Team/Blue Team Overview II

- Purple Team: communication between Red and Blue Teams
  - Sharing lessons learned, strategies and tactics
  - Often from Red Team to Blue

# Red Team/Blue Team Overview – Goals

- Broadest goal: secure organization to focus on business goals
  - improve security of organization
- Test real-world preparedness
- Identify weaknesses
- Test ability to detect and respond
- Deliver specific recommendations for improving security
- Chief constratint:
  - Breadth vs Depth
  - Focus on a company, business unit, group or function

# Red Team/Blue Team Overview – Types I

- Vulnerability Assessment
  - Broad in scope
  - Find vulnerabilities and suggest remediation and prioritize
- Penetration Test
  - Narrow in scope
  - Test specific systems
  - Attempt to steal or modify data, acquire privileges and/or gain a foothold
  - Document specifically how the systems or bypassed, defeated or compromised
  - Suggest remediation and prioritize
- Audit
  - Broad or narrow
  - Verify compliance against some standard

# Red Team/Blue Team Overview – Types II

  - Audits don't prove an organization is secure
  - Document deviation from standard
- Target
  - Physical assets
  - Technological assets
  - Processes
  - People
- Risk Assessment
  - Identifying and assessing risks (probability and impact)
- Threat Assessment
  - Determining credibility of detected threats
- Threat Modeling
  - Model a given threat actor against a given attach scenario

# Red Team/Blue Team Overview – Developer Response

- Isn't this a special case of the development/QA process?

| Security | Development |
|---|---|
| Blue Team | Developer |
| Red Team | QA |
| Assessment | Code Review |
| Penetration Test | QA Black Box Testing |
| Audit | QA White Box Testing |

# Software Development Overview I

- Product Management: define requirements
  - Proxy for customer
  - Intermediary between developers and customers
  - Define use cases
  - Identify operational and technology limits or requirements
  - Work with business stakeholders to define acceptable costs
- Quality Assurance: compliance and acceptance testing
  - Proxy for product management
  - Validate software against requirements
  - Ensure adherence to coding standards
  - Verify accuracy of documentation
- Software Engineering: design and build
  - Convert product requirements into a technology plan
  - Select best technologies to achieve requirements

# Software Development Overview II

  - Define coding standards
  - Write software
  - Write unit tests
  - Implement Research discoveries as reliable, consistent products
- Documentation: communication
  - Document features as implemented
  - Convert use-case descriptions into task steps
- Research: discovery
  - Discover or invent new processes, algorithms or technologies
  - Focused effort to resolve particularly intractable problems

# Software Development Overview – Goals

- Broadest goal: delivery features and functions to meet customer needs
- Solve business problems
- Chief Constraints:
  - Features vs Time
  - Ease-of-use vs "Power"

# Software Development Overview – Types

- In-house software
  - Develop systems and solutions for internal corporate use
- Retail software (Independent Software Vendors)
  - Develop systems and solutions for sale

# Software Development Overview – InfoSec Response

- Where does security fit in this model?

# Software Development Overview – InfoSec Response

- Where does security fit in this model?
- Here's where we the drop the bomb:

# Software Development Overview – InfoSec Response

- Where does security fit in this model?
- Here's where we the drop the bomb:
  - Security is just another feature vying for developer time

# Software Development Overview – InfoSec Response

- Where does security fit in this model?
- Here's where we the drop the bomb:
    - Security is just another feature vying for developer time
- Put that tar down, step away from the feathers!

# Software Development Overview – InfoSec Response

- Where does security fit in this model?
- Here's where we the drop the bomb:
  - Security is just another feature vying for developer time
- Put that tar down, step away from the feathers!
  - You're going to need them in a few minutes. . .

# Now We See the Impedance Mismatch Inherent in the System I

- What do you mean "Security is just another feature vying for developer time"?
- Security isn't an essential part of the software development process
- If there were no bad actors security *per se* would be unnecessary
- But there is quality and robustness
- Since there are bad actors, security can be seen as a necessary component of quality and robustness
- As InfoSec people are well aware, there is no single technological definition of "secure"
    - Yet we know it when we see it

# Now We See the Impedance Mismatch Inherent in the System II

- "Secure" has to be defined for a given situation:
  - Risks have to be identified
  - Probabilities assessed
  - Threats modeled
- Does that mean InfoSec have to define everything?

# Does InfoSec Have to Define Everything?

- No
- There are baseline security practices
- Many developers know many of them

# Security Among Software Developers – Challenges

- Little or no risk assessment
- Little or no threat modeling
- Little or no formal security training
- Increasingly no operating systems training
- Increasingly no formal software-development training

# Security Among Software Developers – Where is Security Found? I

- Where are developers most focused on security?
- Open Source:
  - OpenBSD
  - HardenedBSD
  - SELinux
- Web-centric businesses:
  - Amazon
  - Facebook
  - Google
  - PayPal
  - Twitter
- Some of the larger ISVs (Independent Software Vendors):

# Security Among Software Developers – Where is Security Found? II

  - Adobe
  - Microsoft
  - Oracle
- ISVs with a security-software focus
- In-house developers and smaller ISVs:
  - Unless in the security space, haphazard
  - Quality of security varies based on individual developer's security experience

# Security Among Software Developers – Quality Assurance I

- Testing usually done in isolation, not as part of organization security
- Focused on compliance with:
  - Feature requirements
  - Operating parameters
- Security testing mostly focused on application requirements:
  - Ensure sufficient permissions of daemon/service user
  - Ensure R/W permissions for necessary files and directories
  - Ensure correct database permissions
  - Applications user roles and permissions (where applicable)
- Input validation focused on:
  - Correctly recognizing and rejecting invalid values

# Security Among Software Developers – Quality Assurance II

  - Graceful recovery and state rollback
- Catastrophic failure testing focused on:
  - Finding catastrophic failure conditions
  - Exploitation not typically a concern
  - Unless extremely obvious

# Security Among Software Developers – Software Developers I

- Focused almost entirely on feature implementation
- Emphasis on correctness of implementation
- Input validation focused on:
    - Rejecting invalid values
    - Graceful recovery and state rollback
- Catastrophic failure focused on:
    - Prevention
    - Root-cause analysis
    - No follow-up to see whether the defect can be exploited
- Security often limited to application layer:
    - Application users, groups and roles
    - Permissions within application

# Security Among Software Developers – Software Developers II

- Encryption:
    - Too often preferred to hashing
    - Algorithms often selected solely on key size
- Requirements often don't include organizational security requirements
- Logging:
    - Mostly focused on debugging
    - Little concern for or knowledge about forensic requirements
    - Often to application-controlled files or localhost logging service
    - Very rarely capable of logging to a central server

# Where To From Here? – InfoSec I

- Get developers involved as early as possible:
  - Developers don't magically know your requirements
  - They must be specified
  - Whether for in-house development
  - Bespoke
  - Or purchased
- Define your use cases
- Specify interactions between software and other systems
- Specify the features you need:
  - Privilege separation
  - Least privilege
  - Specify what "failing securely" looks like
  - Acceptable encryption requirements

# Where To From Here? – InfoSec II

  - What data must be encrypted, hashed or never encrypted
  - What data must never be stored on disk (or other long-term media)
- Specify logging requirements:
  - Central logging
  - What forensically-valuable data looks like
  - Hash chaining
- Specify operational limits:
  - Uptime requirements
  - Discrete crypto-task max times and memory for small-batches
    - Validating a single user's password
    - Encrypting a credit-card number in-flight
  - Discrete crypto-task min times and memory for large-scale batches
    - Cracking a bunch of passwords

# Where To From Here? – InfoSec III

- Specify patch management

# Where To From Here? – Software Developers I

- Education
  - Learn basics of security
  - Understand CIA: confidentiality, integrity and availability
  - Recognize security is a process, not a destination
  - Study SANS Top 20 Critical Security Controls
  - Study cryptography engineering
  - Know your OS and its services
  - Dive deep into security best practices and standards for your industry (e.g., HIPPA, PCI)
- Practice Secure Coding Principles (adapted and extended from OWASP)
  - Minimize attack surface area
  - Establish secure defaults
  - Principle of Least privilege

# Where To From Here? – Software Developers II

  - Principle of Defense in depth
  - Fail securely
  - Don't trust services
  - Privilege separation
  - Avoid security by obscurity
  - Keep security simple
  - Fix security issues correctly
  - Ruthlessly remove unused code
- Requirement Gathering:
  - Ask about security requirements
  - Ask whether risk-assessments have been made and for copies
  - Ask whether threats have been modeled and for copies
  - Ask about logging requirement
- Cross Train

# Where To From Here? – Software Developers III

  - Risk assessment
  - Threat modeling
  - Capture the Flag events
  - Reversing (Radare, IDA)
  - Security Meetups
  - Books, Blogs and Vulnerability publications
  - Conferences
- Get Certified

# Conclusion

- Software developers have a role to play (mostly) on the blue team
  - Though some of us like writing hacking tools
- Requires communication which means looping developers in as early as possible
- It requires software developers to educate themselves more broadly about security just as we do other technologies
  - See Notes on Certification and Resources section at end of presentation

# Questions

- Questions - You have them, I may have answers

# Contact Details

- Aaron Poffenberger
- akp@hypernote.com
- `http://akpoff.com`
- @akpoff
- This presentation, look for blog post on `http://akpoff.com`
- KG5DQJ

# Notes on Certifications

- Not a lot of good secure-software development certifications
    - (Not a lot of good software development certifications, either)
- Avoid ticket farms
    - Certifications should demonstrate working knowledge, not your ability to cram
    - And to some degree, experience though many people with experience don't have certifications
- Can be useful when contracting/working for the government
- Can help organize experience into common domains, terms and grammar
- Some demonstrate mastery of a specific task or domain of tasks

# Resources

- Very select and incomplete list of resources
- Google is your friend
- InfoSec people are your friends

# Introduction to Security

- SANS
  - Twitter: @SANSInstitute
  - SANS Top 20 Critical Security Controls
- Lesley Carhart
  - Twitter: @hacks4pancakes
  - Starting an InfoSec Career – The Megamix – Chapters 1-3
- Daniel Miessler
  - Twitter: @DanielMiessler
  - Security Assessments Types
  - The Difference Between Red, Blue, and Purple Teams
  - The Difference Between a Vulnerability Assessment and a Penetration Test
- No Starch Press

# Secure Coding and Cryptography Engineering

- Apple – Introduction to Secure Coding Guide
- Apple – Security Development Checklists
- Cryptography Engineering: Design Principles and Practical Applications
- Microsoft – Secure Coding Guidelines
- Oracle – What Developers Need to Know About Java Security
- SANS – Top 25 Most Dangerous Software Errors
- OWASP – Secure Coding Principles
- UCal Berkeley – Secure Coding Practice Guidelines

# Vulnerability Resources

- Exploit Database
- Mitre Corp.
    - CVE: Common Vulnerability Enumeration
    - CWE: Common Weakness Enumeration
    - Mitre Quick Reference
- SANS Internet Storm Center
- Securityfocus BugTraq (BID) Database

# Certifications and Training

- CompTIA
  - CompTIA Security+
  - CompTIA Advanced Security Practitioner
- ISC$^2$ (International Information Systems Security Certification Consortium)
  - Certified Information Systems Security Professional (CISSP)
  - Certified Secure Software Lifecycle Professional (CSSLP)
- Offensive Security Certified Professional
- SANS Institute
  - Information Security Training
  - Global Information Assurance Certification