Almost Intractable Application Security Problems… and Solutions

OWASP Dallas

25 May 2021

Danny Harris

SDLC Center of Excellence Technical Lead

Senior Security Consultant

dharris@securityinnovation.com

SECURITY INNOVATION

What This is About

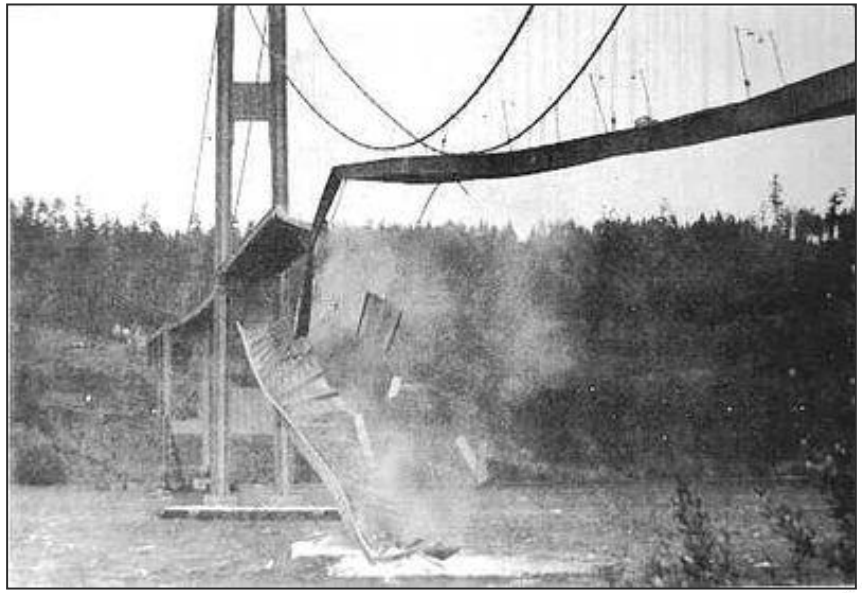
- One person's view into some of the security challenges facing software development teams
- Based on seeing many organizations that build a variety of different types of applications
  - This talk is purely anecdotal and is based on my observations as an application security consultant at many organizations
- There are many commonly occurring big challenges
  - When considered as a whole, these challenges can seem intractable
- We will discuss solutions!

Building Secure Software is Very Hard to Do Well

- Software is easy to criticize, but difficult to build right
- It is easier to break something than it is to design something that can't be broken
  - "Any jackass can kick down a barn, but it takes a good carpenter to build one" – Sam Rayburn
  - Comment: It also is very important to have a good architect to constrain the carpenters and set them up for success, while minimizing the risk...

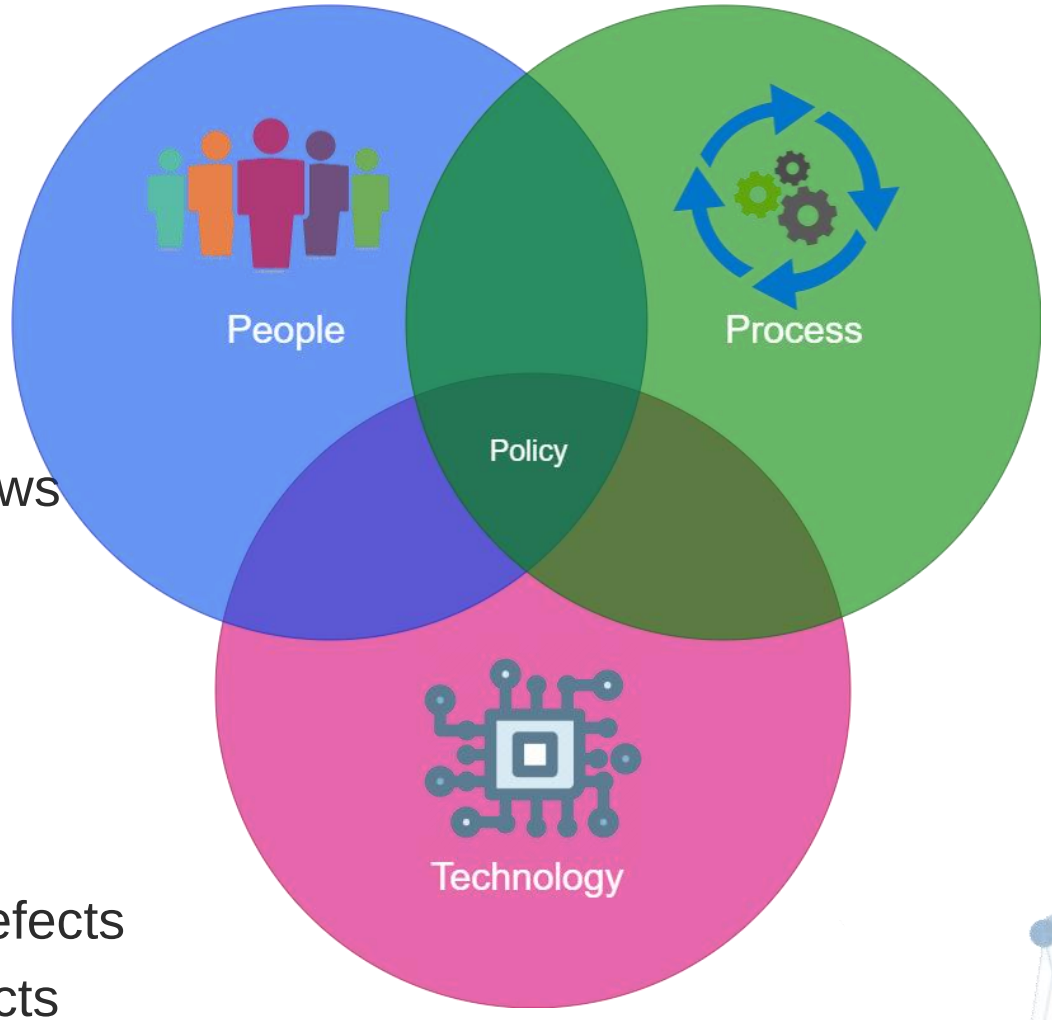
Building Secure Software is Very Hard to Do Well

- Secure software must be designed for the…
  - Success case (proper functionality and operation)
  - Failure case (attacks, stupid users, abuse, misuse) ← This often is neglected
- Learning how to design, build, and test secure software focuses attention to the failure case and how to handle it gracefully by failing securely



Agenda

- People
  - Failure to Understand Application Security Risk
  - Failure to Understand the Threat Actors
  - Lack of Awareness of the Threats to Applications
  - Confusing Testing with Security Testing
  - Confusing Code Reviews with Security Code Reviews
- Process
  - Failure to Design Security in From the Outset
  - Failure to Continuously Test Throughout the SDLC
- Technology
  - Relying Primarily on Technology to Find Security Defects
  - Failing to Correctly Use Tools to Find Security Defects
  - Believing that Perimeter Security is Sufficient



Topic Format

- Each topic we cover has the following sections:
  - Observations
    - What we've seen at various organizations
  - Implications
    - The implications of our observations
  - Causes
    - The causes of the problem
  - Solutions
    - Some solutions to prevent or mitigate the problem

Everybody contributes to the challenges of building secure applications

http://dilbert.com/strips/comic/1995-11-13/

SECURITY INNOVATION

Failure to Understand Application Security Risk

- Observations
  - **Business** risk is often confused with **security** risk
    - They are not exactly the same!

28 AUG 2015 NEWS

CEOs Failing to Grasp Information Security Risk

Joe O'Halloran Editor In Chief, Rapid TV News

Despite a continuing string of high-profile information security breaches, many organizations' leadership teams still have a very poor understanding of their own susceptibility to similar failures, asserts a research note from leading analyst Ovum.

Why Not Watch?

9 APR 2015

26 FEB

http://www.infosecurity-magazine.com/news/ceos-failing-to-grasp-full-u/

SECURITY INNOVATION

Failure to Understand Application Security Risk

- Business Risk
  - “The willful, planned exposure of capital and resources to the chance of loss to make a profit or achieve an organization’s objective”
  - Most business leaders understand and are familiar with business risk

- Security Risk
  - “The involuntary exposure to chance that forces or people not under the organization’s control will abuse or misuse the organization's information [or systems or processes] to cause a loss”
  - It is very difficult to predict what an unknown force or malicious person may do to information and systems at some time in the future
  - It is also difficult to predict the loss that may ensue as a result of such activity

Security risk is really a **component** of business risk, just as physical risk and financial risk

Source: Donn B. Parker, “The Alternative to Risk Analysis”, **Computer Security Alert**, number 198, September 1999, p 2.

Failure to Understand Application Security Risk

- Implications
  - Conflating business risk with security risk can mean that real security risks don't receive the attention they should be getting
  - Applications can remain vulnerable to unidentified security risks
- Causes
  - Having a single process for risk management that focuses on traditional business risk
    - "If your only tool is a hammer, then every problem looks like a nail"
    - Business risk is what Management/Business best understand
  - Failing to understand the very real and significant differences between business and security risk
    - Security risk is more complex than business risk
      - It's harder to translate security risk into business risk that executives understand
    - Executives believe that their dev teams are doing everything right
  - Executives may not have enough (any?) visibility into the application security development process
    - Security defects happen at every phase of the Software Development Lifecycle (SDLC)

Failure to Understand Application Security Risk: Solutions

- Educate Management/Business about application security risk and how it differs from business risk
  - Show how security defects are introduced
  - Understand the attackers: well-funded criminals, nation states, activists, script kiddies, etc.
    - How attackers view the executive's enterprise
  - Applications can't be completely security defect-free
    - The best you can do is to have sufficient defense-in-depth and detection controls/processes in place
  - Understand the business risk of not fixing security defects
    - Develop an accurate understanding of the existing security defects with an articulated and agreed upon vulnerability management process
- Application security and development teams need to normalize language so they can translate security risk into business risk
  - Otherwise, the "business" people will never see or fully understand the issues
- Develop an application portfolio risk profile that is visible to the executives
  - They can see how risk varies with application criticality across the entire application portfolio

Failure to Understand the Threat Actors

- Observations
  - Executives don’t necessarily recognize that the cybercriminal ecosystem is large and diverse
  - No one is really interested in what we do. We just do <X>…
- Implications
  - “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*
  - Systems will be designed, built, and deployed in a less robust and secure manner
- Causes
  - When Management/Business think “hacker”, they may think a teenager or maybe a competitor
  - They don’t believe anyone would bother them because of what they do (too insignificant) or what they make
    - “We’re too small”, “We just make widgets”, etc.

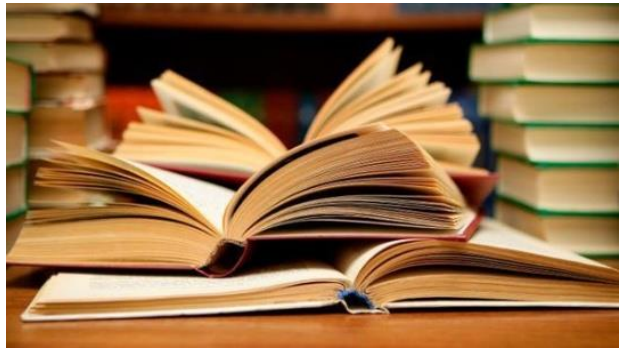
Failure to Understand the Threat Actors: Solutions

- This is an education issue
  - Understand who the attackers are and what their capabilities are
  - Understand the kinds of harm they can do to your specific systems
  - Show examples from related industries, verticals, and even from the same locale
- Have the executives completely work through the implications of few different kinds of attacks as a tabletop exercise (it's not easy)
  - Web defacement
  - Ongoing denial of service
  - Ransomware extortion

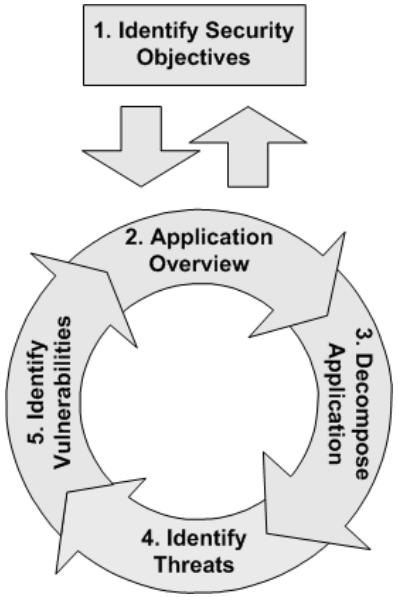
Lack of Awareness of the Threats to Applications

- Observations
  - The business and development team don't have a comprehensive understanding of the threats facing applications
- Implications
  - If the business and development team are unaware of the full range of threats, their ability to build software that proactively defends against those unknown threats will be limited
    - "The worst thing that can happen to a good cause is, not to be skillfully attacked, but to be ineptly defended." – Frédéric Bastiat
  - If threats are not considered, how can the application be designed and built to defend against those threats?
  - Without knowing the threats, it is not possible to validate that the actual implementation is effective against the likely threats
- Causes
  - Ignorance of the threats
    - Most development teams don't have formal exposure to application security threats (and defenses)

Lack of Awareness of the Threats to Applications: Solutions



- Get everyone educated about the common application threats
  - Understand who the attackers are and their motivations
  - Understand what kinds of harm attackers can do to the application and data
  - Show the team common kinds of attacks against a real application
  - Let the team try these attacks themselves
    - It makes a big impression!
- Get the team involved with threat modeling the application
  - This will get them thinking about the threats before coding
    - Ensures that the application will be developed to adequately defend against them
    - Threat model throughout the SDLC
      - It is the most effective way to reduce risk by taking a threat/risk-based approach, not just counting vulnerabilities
  - Use the threat model during testing to validate that the application is protected against the threats enumerated by the threat model



SECURITY INNOVATION

Confusing Testing with Security Testing

- Observations
  - There is often confusion between functional testing and security testing
    - **Functional** testing is focused on what the application **should** be doing
    - **Security** testing focuses on what the application **should not** be doing
    - Everybody does functional testing
    - Many organization do little or no security testing or don't know how to do it well

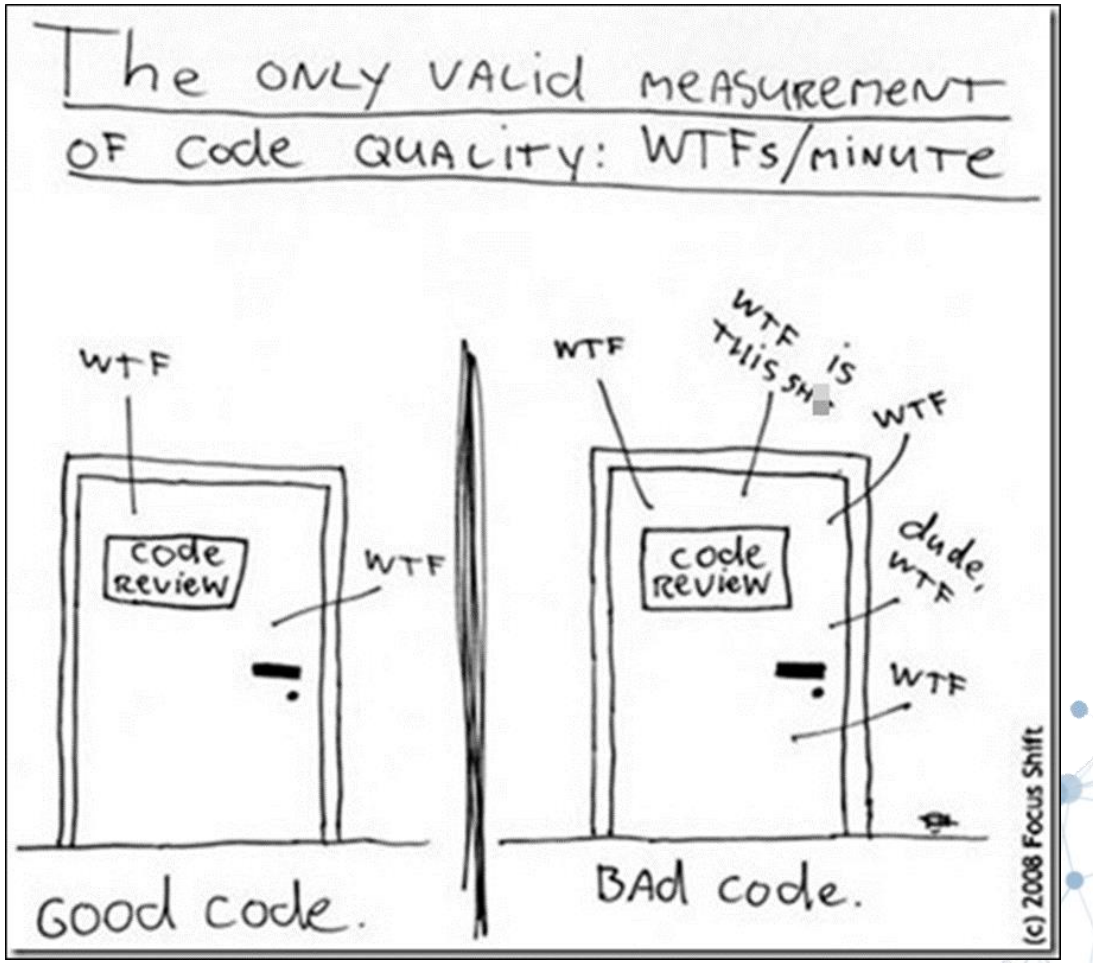
- Implications
  - The application may not get any security testing or may not get sufficient security testing
- Causes
  - People conflate "testing" with security testing
  - The business and developers assume that the testers and QA do whatever testing is appropriate
    - Unfortunately, that does not always include security testing

Confusing Testing with Security Testing: Solutions

- This is an education issue:
  - The business and development teams need to understand the significant differences between functional and security testing
  - Testing teams need to learn how to do application security testing if they don't already do it
- Application security testing requires additional skillsets and tools
  - Ensure that QA/testers develop additional application security testing skills to handle the variety of technologies and platforms
- Expose QA/testers (and the rest of the development team) to application security testing
  - This will help everyone understand what attackers do and how they do it
  - Developers should also conduct lightweight security testing within the context of coding to find obvious security defects while they are coding
  - It allows the development team to find and fix security defects during coding

Confusing Code Reviews with Security Code Reviews

- Observations
  - Many development organizations say that they do code reviews
  - Upon closer examination, they really do functional code reviews, as opposed to security code reviews
- Implications
  - Security code reviews are not being done
    - They require advanced knowledge and practice
      - They can be hard to do
  - Security defects remain in the code
- Causes
  - People confuse code reviews with security code reviews
  - The traditional code review is a functional code review and doesn't necessarily focus on security issues



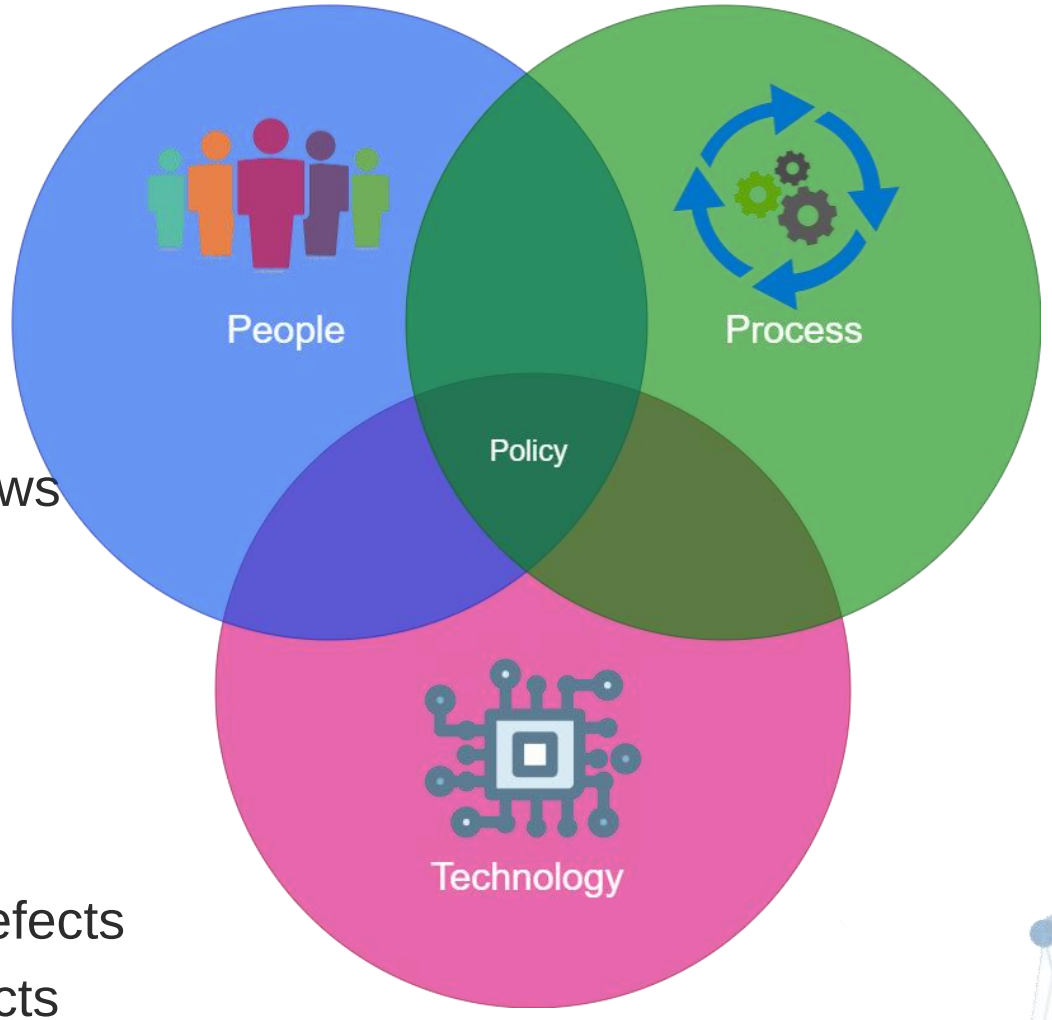
http://www.osnews.com/story/19266/WTFs_m

Confusing Code Reviews with Security Code Reviews: Solutions

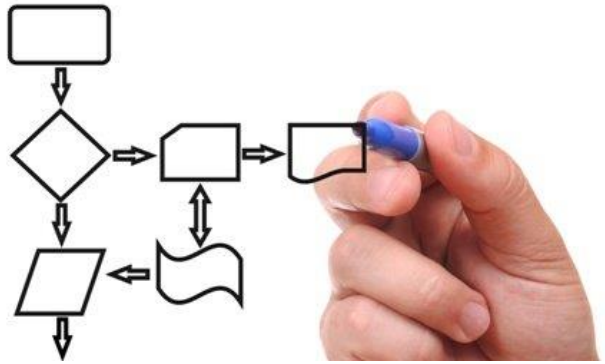
- Create/update a list of code inspection questions
    - These are questions to ask during the code review to help focus the review efforts on security
    - The inspection questions are technology specific
    - They represent patterns of problems that may occur in your application
        - What problems are possible in the technology?
            - E.g., Should you be looking for buffer overflows?
        - What problems are possible due to the design?
            - Are there threat mitigations already in place?
- Identify the security defects
    - Where in the code are they found
    - Under what conditions
- Document a fix
    - How can the security defects be resolved

Agenda

- People
  - Failure to Understand Application Security Risk
  - Failure to Understand the Threat Actors
  - Lack of Awareness of the Threats to Applications
  - Confusing Testing with Security Testing
  - Confusing Code Reviews with Security Code Reviews
- Process
  - Failure to Design Security in From the Outset
  - Failure to Continuously Test Throughout the SDLC
- Technology
  - Relying Primarily on Technology to Find Security Defects
  - Failing to Correctly Use Tools to Find Security Defects
  - Believing that Perimeter Security is Sufficient



Failure To Design Security In From The Outset



- Observations
  - Application security is often not given the due consideration it requires
    - “We will do security at the end…”
    - “We are using TLS…what’s the big deal about security…”
    - “We’ve got a firewall, IDS/IPS, even a WAF…”
- Implications
  - Applications that are built without the security considerations during design are more likely to have significant security issues
    - Developers are putting in features as specified
    - But if security controls aren’t on the “features” list, they won’t necessarily be integrated into the application
  - Design defects are significantly more difficult and more expensive and take longer to fix later in the SDLC
- Causes
  - There is tremendous pressure from the business to produce software with the latest and greatest features
    - Security may not be considered as adding sufficient value to warrant adding to some future release

Failure To Design Security In From The Outset: Solutions

- Thinking about security requirements at the earliest stages of the SDLC is one of most powerful ways of ensuring that the application will be reasonably secure

- Develop a list of common security requirements and security design principles to help development teams get started
  - Help architects/developers/testers to avoid mistakes in during design, implementation, and testing
- Threat modeling and architecture design reviews are great tools to help validate security requirements

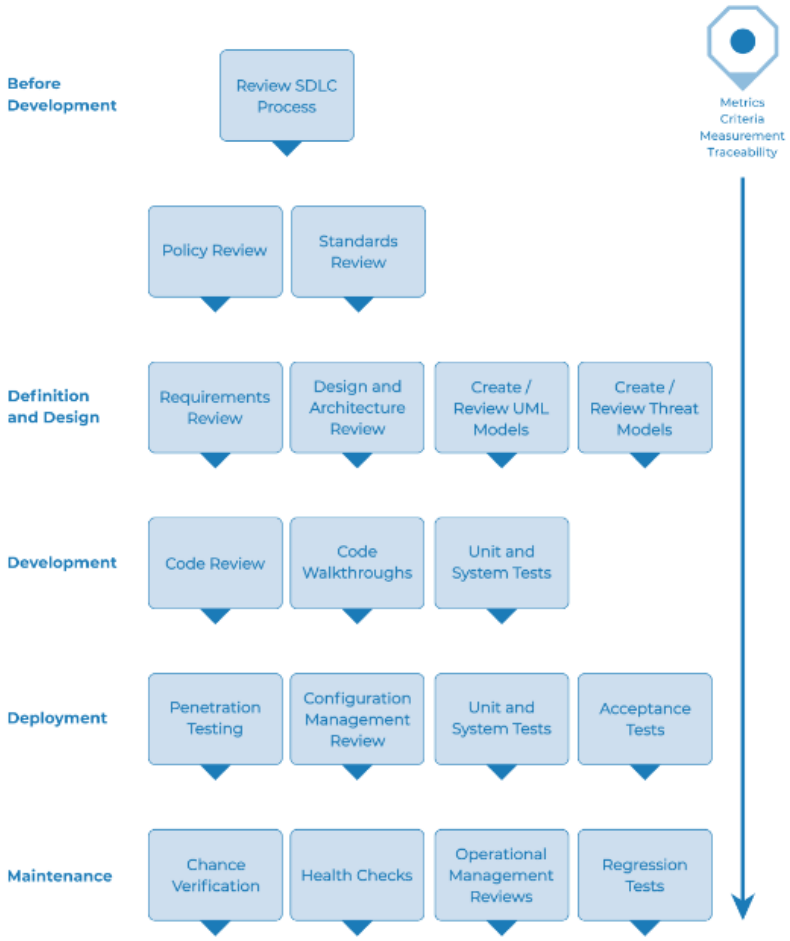
Failure to Continuously Test Throughout the SDLC

- Observations
  - A commonly held belief is that testing is just done at the testing phase, after the coding/implementation milestone has been completed
  - QA is exclusively responsible for testing

- Implications
  - Just testing at the verification/testing phase finds security defects from all the previous phases
  - The later in the lifecycle a defect is found, the harder, more expensive, and longer it takes to fix
- Causes
  - False belief: The time to test is during the verification/testing phase
  - QA is viewed as being responsible for security testing, if any
  - We don’t have the resources to test except during the testing phase

Failure to Continuously Test Throughout the SDLC: Solutions

- It is far easier, faster, and less expensive to integrate security testing into the SDLC earlier, rather than to wait until a security defect is discovered
- If we broaden the definition of security “testing” to include security reviews, doing testing throughout the SDLC can result in a more secure application
- Provide training and tools (based on role) to the entire development team to ensure that security testing happens during all phases of the SDLC
- Fix the issues in a timely fashion
  - Don’t just push all of the security issues to an ever-growing issues backlog

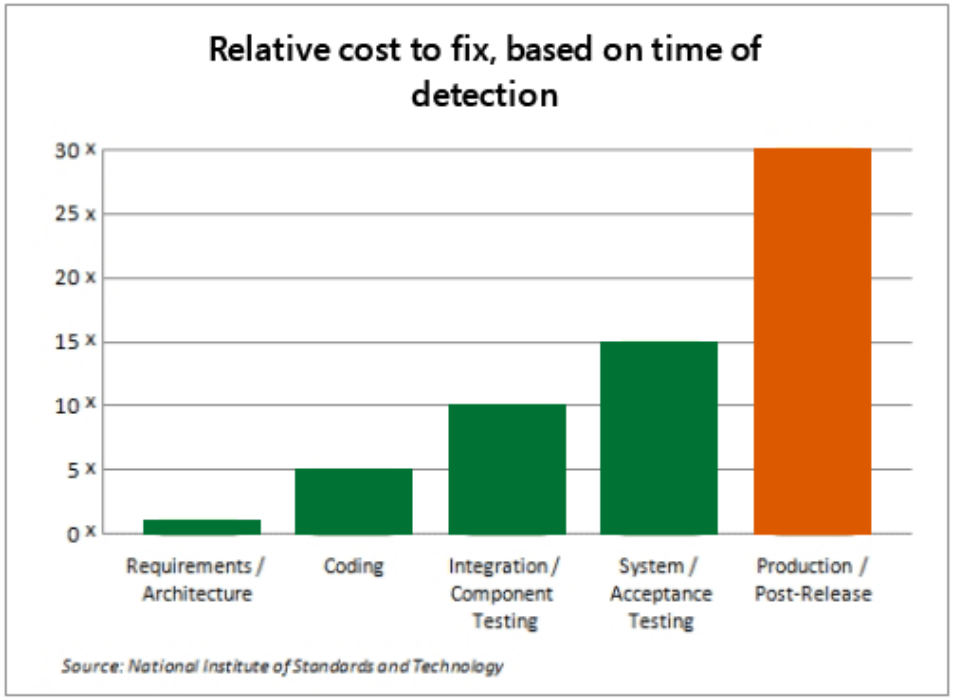


Source: [OWASP Testing Guide](), v 4.2 p. 40

SECURITY INNOVATION

Failure to Continuously Test Throughout the SDLC: Solutions

- It can cost 30 times more to fix a bug in production than in the earlier stages of the SDLC

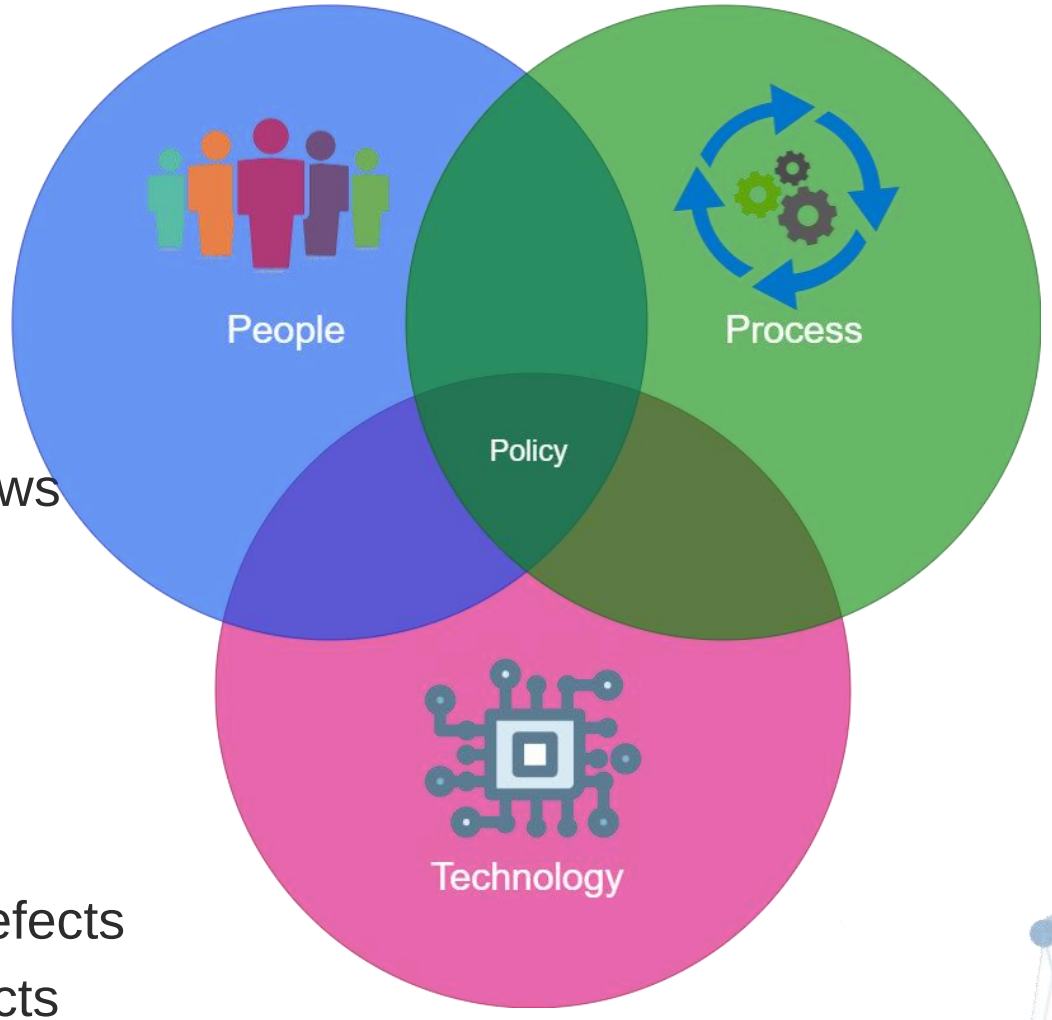


- Security defects will occur
- Conducting security testing throughout the software development lifecycle enables the development team to find and fix defects **earlier in the lifecycle,** avoiding the exponentially increasing costs of late remediation
- A secure software development lifecycle can help achieve this

Source: National Institute for Standards and Technology: *The Economic Impacts of Inadequate Infrastructure for Software Testing*, p 5-4
http://www.nist.gov/director/planning/upload/report02-3.pdf

Agenda

- People
  - Failure to Understand Application Security Risk
  - Failure to Understand the Threat Actors
  - Lack of Awareness of the Threats to Applications
  - Confusing Testing with Security Testing
  - Confusing Code Reviews with Security Code Reviews
- Process
  - Failure to Design Security in From the Outset
  - Failure to Continuously Test Throughout the SDLC
- Technology
  - Relying Primarily on Technology to Find Security Defects
  - Failing to Correctly Use Tools to Find Security Defects
  - Believing that Perimeter Security is Sufficient



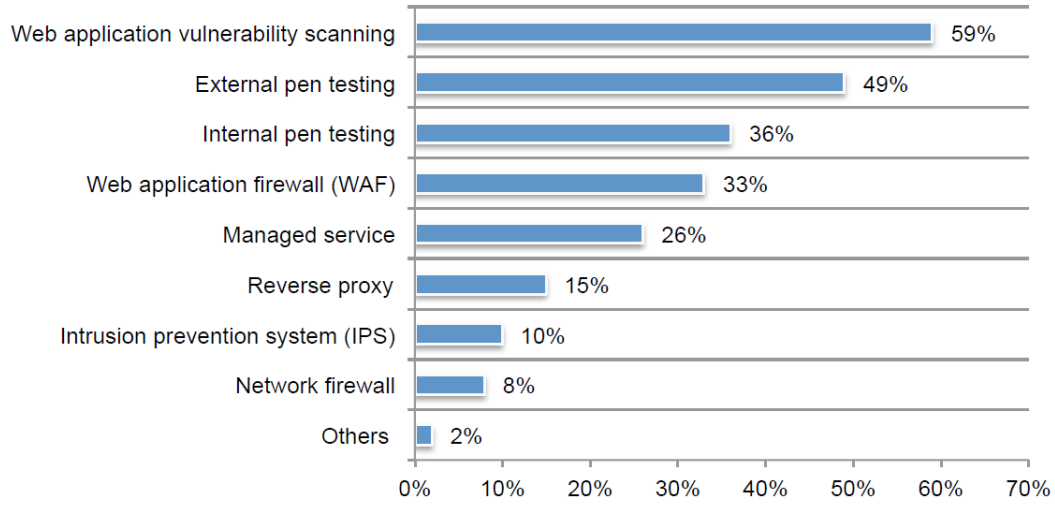
Relying Primarily on Technology to Find Security Defects

- Observations
  - Technological solutions are seen as the **primary** way of finding security defects
- Implications
  - There is no technology magic bullet
  - Parts of the application are simply not covered with just using technology alone to find security defects
  - Development teams often tend to rely exclusively on technology solutions
  - Certain classes of security defects are more likely to be found when a human is involved in the process
    - Business logic errors, complex and multi-stage attacks
- Causes
  - Technology solutions can do a good job of finding many (but not all) types of security defects
    - These solutions can be very cost effective and often scale nicely
  - There is a failure to recognize the importance of human thinking and problem solving necessary for finding security defects in complex systems

Relying Primarily on Technology to Find Security Defects: Solutions

- Use technology solutions to find security defects **in conjunction with manual efforts**
- Manual or human solutions are necessary for complete testing coverage
  - It may not be possible to test some parts of an application with just automated tools
  - Be sure the tools are capable of adequately testing the application
- At a minimum, include manual testing for high-risk and security-sensitive components of an application
  - This includes also conducting manual security code reviews for those parts of the application

Figure 5. What are your primary means of securing applications?
More than one response permitted



Ponemon – *The Increasing Risk to Enterprise Applications*

SECURITY INNOVATION

Failing to Correctly Use Tools to Find Security Defects

- Observations
  - Development teams will often use whatever tools they have to test the applications
  - Using the tool incorrectly or on a system where it can't effectively work will generate poor results
- Implications
  - Not using the existing set of tools correctly can lead to false sense of security
    - "No bugs found! Let's ship it!"
- Causes
  - Lack of training
  - Failing to understand limits of what the tools can do and where they work best
  - Not having the right toolset for the job at hand

Failing to Correctly Use Tools to Find Security Defects: Solutions

- Make sure the team knows how to use the tools effectively
  - When they work best
  - Understand the situations where they don't or can't work well
  - Understand when to do manual assessments to help get complete testing coverage
- Provide…
  - The right set of tools for everyone that needs them
    - Figure out a way to permit "hacker" tools despite the policy against them
      - Criminals will use them against your systems
      - Shouldn't you also try to see if your systems are robust enough to withstand the attacks?
  - Tool training
  - "How to" guidance
  - Cheat sheets
  - Default configurations that reduce false positives

Believing that Perimeter Security is Sufficient

- Observations
  - Perimeter security systems such as firewalls (and even WAFs) and intrusion detection/prevention systems are believed to provide sufficient security for applications
- Implications
  - Perimeter security solutions alone are not sufficient to protect against application attacks
  - Many application attacks easily slip by perimeter defenses
- Causes
  - Most perimeter defenses were never designed to stop application attacks
    - Access to the server through ports 80 and 443 makes web servers part of the external perimeter defenses
    - Attacks flow through network firewalls unimpeded!
  - Security defects in software and applications may allow access to internal network resources

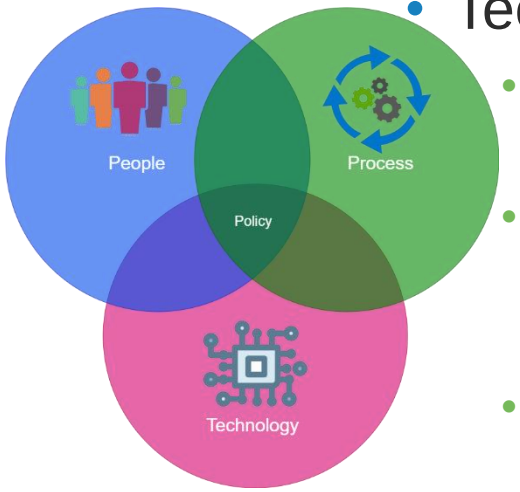
Believing that Perimeter Security is Sufficient: Solutions

- Having strong perimeter security is an important component of an overall good set of security controls, but having a secure application is also important from a defense-in-depth perspective
- Educate the team to understand the necessity of defense-in-depth
  - Perimeter defense is great for stopping some kinds of attacks, but not all
  - Designing, building, testing, and securely deploying the application provides another significant layer of protection when perimeter defenses fail or simply allow the attack through

Summary

- Because applications are complex, there are a number of high-level solutions to deal with the very hard problem of building secure software systems

- People
  - Understand application security risk and how it differs from traditional business risk
  - Understand the attackers who seek to harm systems and data
  - Develop an awareness of application threats
  - Understand the differences between functional and security testing
  - Understand the differences between code reviews and security code reviews



- Process
  - Design security in from the outset
  - Continuously test throughout the SDLC
- Technology
  - Don't rely exclusively on technology to find security defects
  - Be sure to know how to use the tools and technology correctly to maximize the security
  - Understand the limitations of perimeter security and build secure systems using defense-in-depth

References

- Application Security Guide For CISOs
  - https://www.owasp.org/images/d/d6/Owasp-ciso-guide.pdf
- OWASP Cheat Sheet Series
  - https://cheatsheetseries.owasp.org/
- OWASP Code Review Guide v 2.0
  - https://owasp.org/www-pdf-archive/OWASP_Code_Review_Guide_v2.pdf
- OWASP Application Security Verification Standard Project (ASVS)
  - https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project
- OWASP Web Security Testing Guide
  - https://owasp.org/www-project-web-security-testing-guide/
- OWASP Vulnerability Management Guide (OVMG)
  - https://owasp.org/www-project-vulnerability-management-guide/OWASP-Vuln-Mgm-Guide-Jul23-2020.pdf