

Software Security

Changing the Status Quo
by Daniel Cuthbert

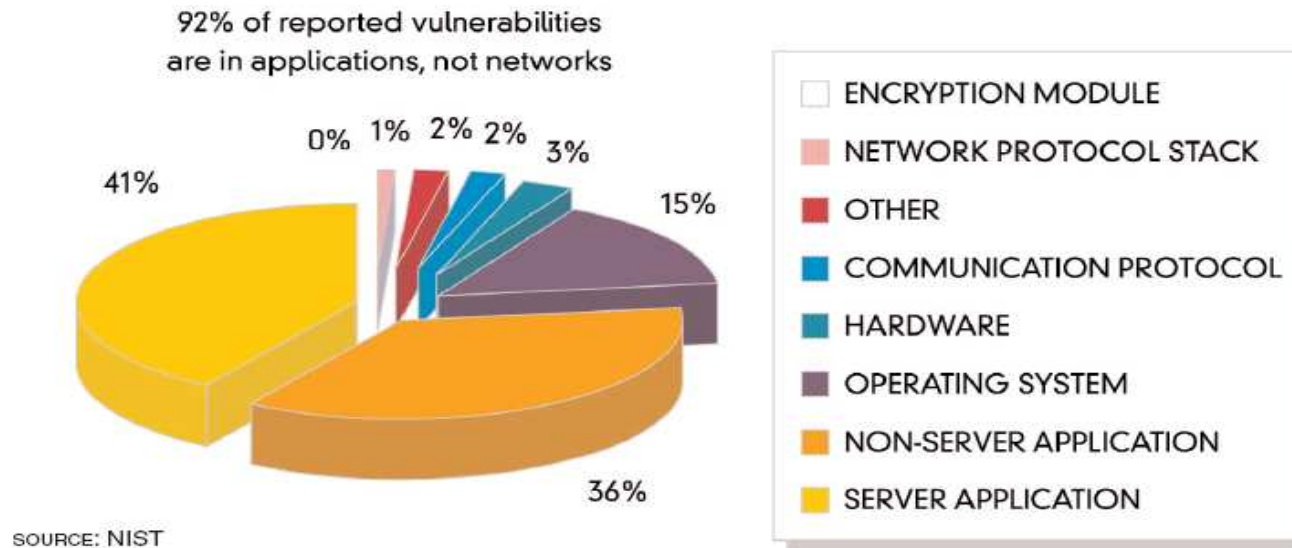


Introduction

Have you tested your software for bugs, scoured it for vulnerabilities of every shape and form and then eliminated them? If not, then your product is not ready to ship!! (*Steven B. Lipner, Senior Director of Security Engineering Strategy at Microsoft*)

The hard facts

- How Many Vulnerabilities Are Application Security Related?



The band-aid approach

- Effective security requires not only prioritising security at the beginning of a project, but also increasing its visibility throughout the lifecycle of the project.
- Application firewalls DO NOT SOLVE THE ISSUE!
- Solving security issues at the end of the QA period isn't the best way forward.
- Developers and architects need to design and understand secure coding techniques and how attackers exploit applications.

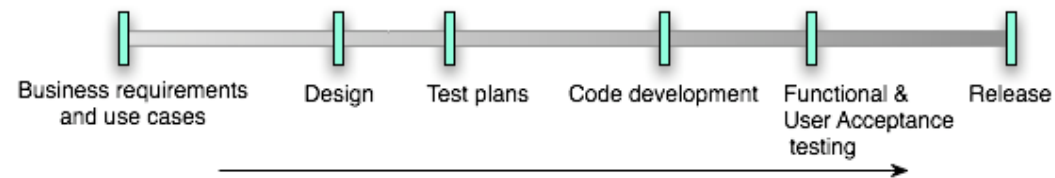
Microsoft's approach

- Microsoft applied a dramatic security overhaul when developing Windows 2003
- Development was stopped and all focus was shifted to security
- What followed was 2 months of:
 - Threat Modeling
 - Risk analysis
 - Penetration testing
 - Source code reviews
 - Production not resuming until all of the bugs were found and eliminated

Why security fails on projects

- Increase of programmer's job scope:
 - Traditional programming discipline has been driven by the challenge of making products optimally efficient.
 - Developers have been focused on providing robust functionality to their customers.
 - Attackers leverage this way of development to perform their malicious activity.
- Separation of security as a new domain discipline:
 - Security isn't a new problem, it's an old problem that now has global media coverage.

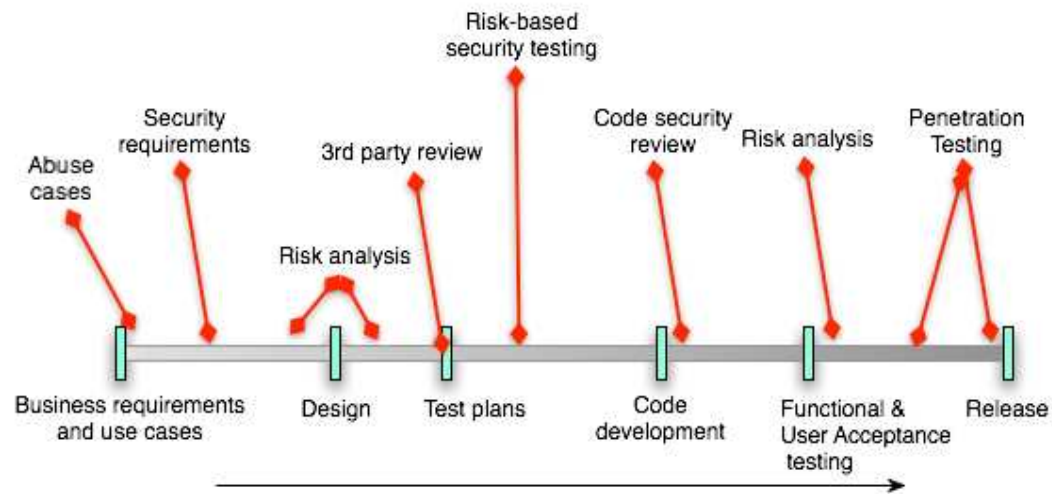
Typical development life cycle



Current development life-cycle problems

- Security is often introduced towards the end of the project.
- Universities have been slow to address the problem of software security.
- Any assessment on the application, or code, is normally performed days before the intended go live date.
- Only 30% of possible security vulnerabilities can be discovered this way.
- There is no stage to include the web application security standards, if any, to the project design.
- Developers are not made aware of any insecure practices they are doing.

Secure Development Life Cycle (SDLC)



Introducing security into the SDLC

- Security is included as part of the design and implementation phase.
- Each stage now has to include some form of security testing or thought process.
- Developers are constantly thinking of how the code/application could be manipulated.
- The end result is an application which has far fewer security vulnerabilities than previous efforts.

Adjustments needed to the current SDLC

- Include security objectives at the start of the project and throughout the lifecycle of the project.
- Make Project Managers/Management accountable for assuring the priority of security on the project.
- Include a mandatory risk assessment and threat modeling exercise before finalising the design.
- Ensure application security standards are met and followed throughout.
- Establish a final accreditation process for the project.



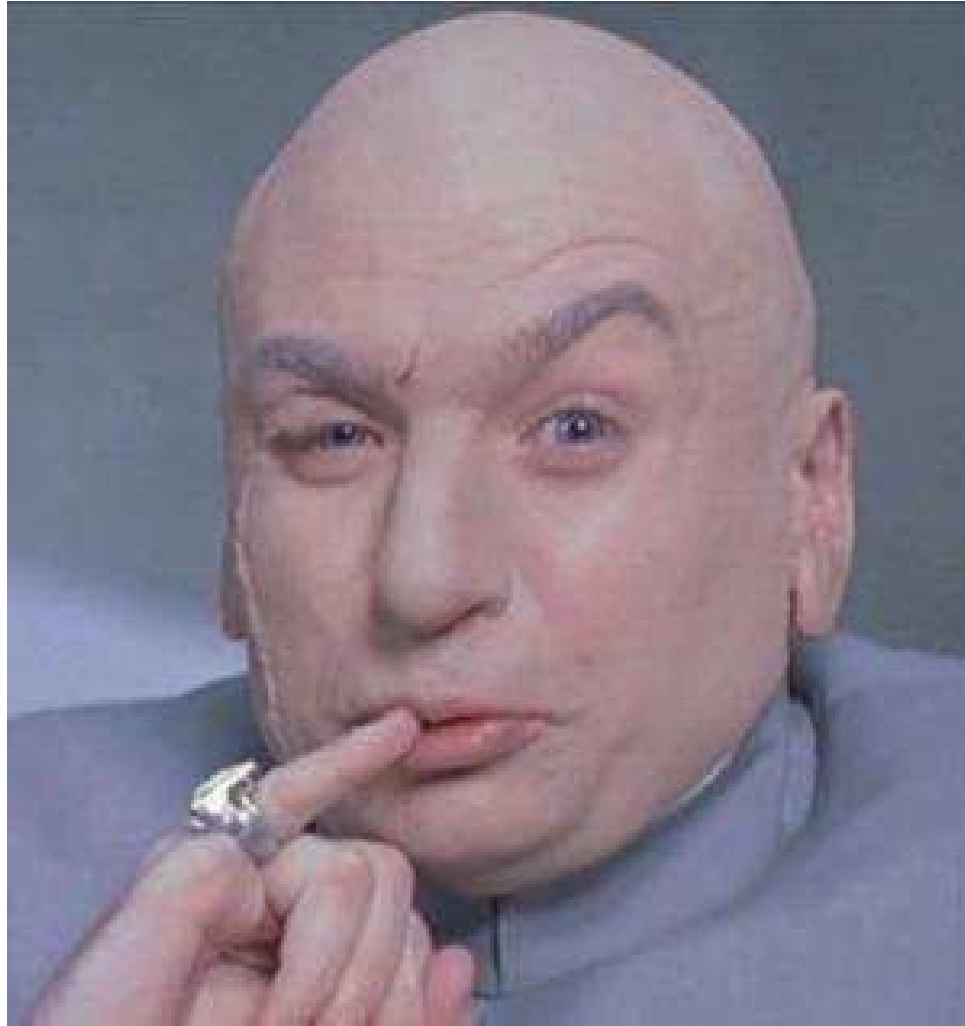
Education and awareness

- Educating developers about current secure development guidelines and good practice:
 - OWASP Guide to building secure web applications.
 - IEEE P1074 Standard for Developing Project Life Cycle Processes.
- Ensuring developers are familiar with ISO 15408 Common Criteria Threat Categories:
 - Allowing for research into latest vulnerabilities and why they happened.
 - Allowing developers to give input on any possible issues found during the development process.
- Making everyone aware that security is no longer a add-on:
 - Using security as one of the selling points of the application/product.

Corsaire training

- Corsaire has developed a series of training courses which aid developers and management in producing secure web applications.
- The courses are based around modules and allow flexibility for developers and companies restricted on training time.
- Ensure that developers are taught why security is important and how to implement security into each of the development stages.

Use cases and abuse cases



CORSAIRE

- Use cases
 - Expected behavior
 - Normal input
 - Functional requirements

- Abuse cases
 - Unexpected behavior
 - By malicious agents
 - Derived from risk assessment
 - Developers should **think evil!**

Thorough security assessment

- Ensure that the people, or 3rd party, performing the assessment have a proven track record in application security assessments.
- Ensure that a methodical approach is taken:
 - e.g OWASP Web Application Penetration Testing Checklist.

Questions?

CORSAIRE