Aula 11
Integrando Segurança ao Processo de Desenvolvimento de Software

Prof. Leonardo Lemes Fagundes

A educação faz com que as pessoas sejam fáceis de guiar, mas difíceis de arrastar; fáceis de governar, mas impossíveis de escravizar.

Peter Drucker

Agenda

- ✓ Divulgação das Notas do Grau A
- ✓ Introdução
- ✓ Panorama Geral Sobre Vulnerabilidades
- ✓ Ciclo de Vida de Desenvolvimento Seguro
- ✓ Considerações Finais
- ✓ Referências Bibliográficas

Introdução

- ✓ **Entendendo o Contexto Atual**
  - ✓ Surgimento de vulnerabilidades em larga escala;
  - ✓ Toolkits para ataques disponíveis na Internet;
  - ✓ Busca por conformidade (aspectos legais, padrões e normas);

Introdução

- ✓ **Entendendo o Contexto Atual (continuação ...)**
  - ✓ Alguns Mitos
    - ✓ Desenvolvedores versus Analistas de Segurança;
    - ✓ A linguagem “X” é mais segura que “Y”;
  - ✓ Alguns Fatos
    - ✓ Falta qualificação técnica das equipes;
    - ✓ O processo de desenvolvimento precisa ser revisto;

Panorama Geral Sobre Vulnerabilidades

- ✓ **A natureza das vulnerabilidades**
  - ✓ Bugs de Projeto
  - ✓ Bugs de Implementação
    - ✓ Aplicação
    - ✓ Plataforma / Arquitetura
  - ✓ Bugs de Configuração

Panorama Geral Sobre Vulnerabilidades

OWASP TOP 10 (2010)

1. Code Injection
2. Cross Site Scripting
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Failure to Restrict URL Access
8. Unvalidated Redirects
9. Insecure Cryptographic Storage
10. Insufficient Transport Layer Protection

Fonte: Owasp Top 10 – 2010 rc1

Ciclo de Vida de Desenvolvimento de Software

- ✓ **Software Development Life Cycle (SDLC)**
  - ✓ Framework que descreve as atividades de cada um dos estágios de um projeto de desenvolvimento de software.
  - ✓ Diferentes metodologias de desenvolvimento.
  - ✓ Estágios: Requisitos, Projeto, Implementação, Teste, Manutenção

Ciclo de Vida de Desenvolvimento de Software

- ✓ **Security Development Lifecycle (SDL)**
  - ✓ Conjunto de práticas de segurança da informação incorporados aos estágios de um processo de desenvolvimento de software.

Ciclo de Vida de Desenvolvimento de Software Seguro

Requisitos

- Identificação dos requisitos de segurança;
- Identificar os tipos de informações;
- Identificar a categoria do sistema;
- Análise e avaliação de riscos preliminares;

Ciclo de Vida de Desenvolvimento de Software Seguro

Projeto

- Análise e avaliação de riscos;
- Modelagem de ameaças;
- Seleção de controles;
- Definição da arquitetura de segurança;

Ciclo de Vida de Desenvolvimento de Software Seguro

Codificação

- Análise e avaliação de riscos;
- Análise estática;
- Lista aprovada de ferramentas;
- Lista de funções e bibliotecas inseguras / proibidas;

Ciclo de Vida de Desenvolvimento de Software Seguro

Verificação

- Análise e avaliação de riscos;
- Definir planos / estratégicas de teste;
  - Análise dinâmica
  - Análise de vulnerabilidades

Ciclo de Vida de Desenvolvimento de Software Seguro

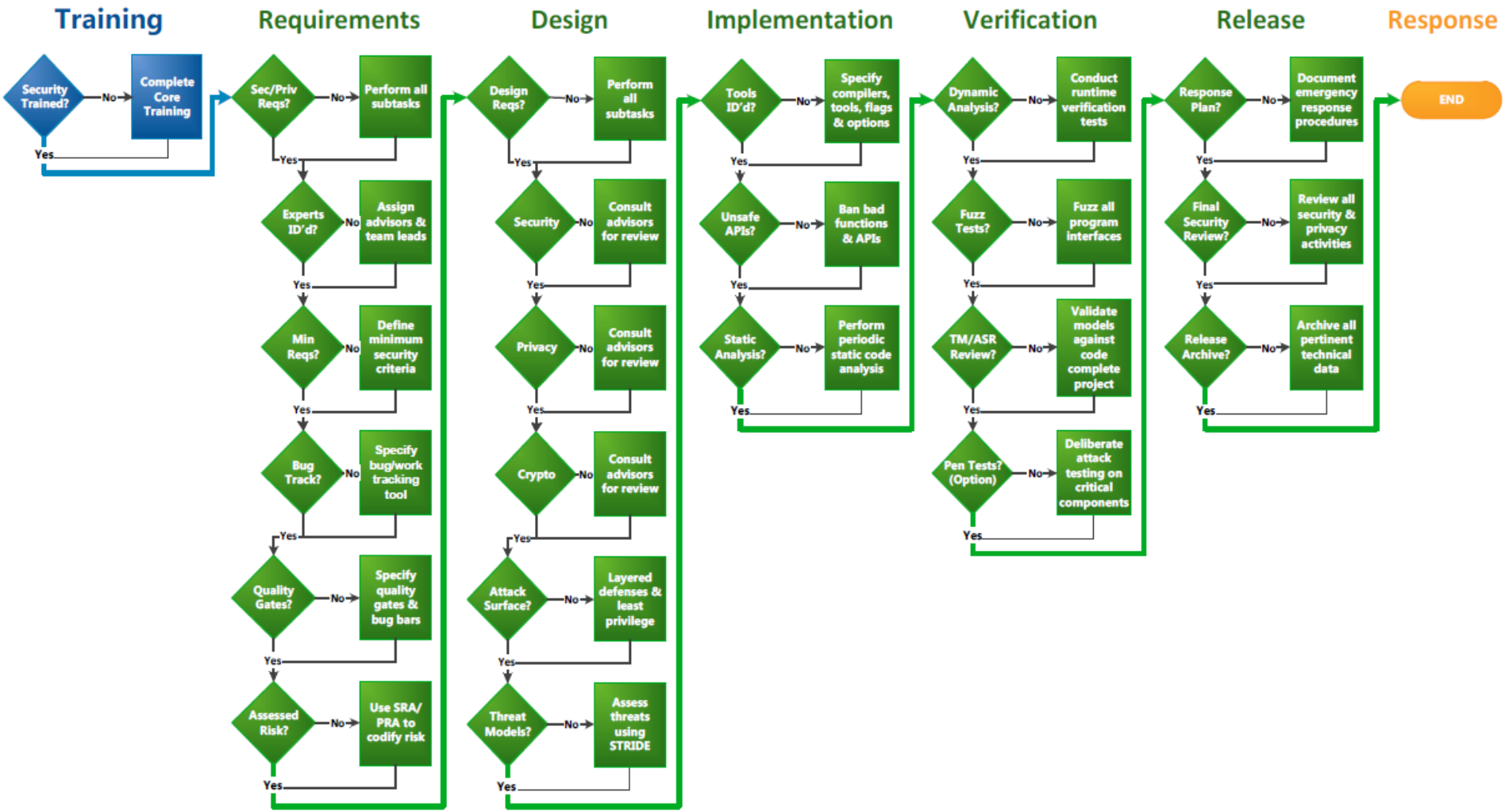
Liberação

- Análise e avaliação de riscos;
- Revisão “Final” de Segurança;
- Planos de resposta a incidentes;

Ciclo de Vida de Desenvolvimento de Software Seguro

Resposta

- Execução do plano de resposta a incidentes;

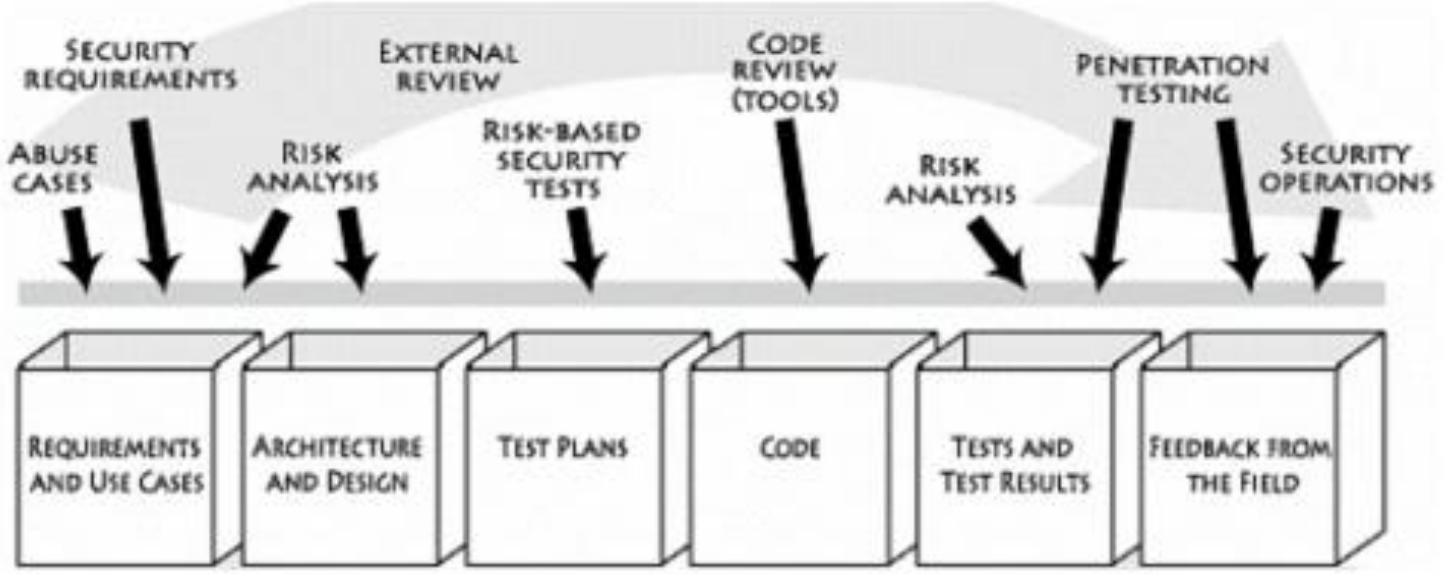

Training
Security Trained?
No
Complete Core Training
Yes
Requirements
Sec/Priv Reqs?
No
Perform all subtasks
Yes
Experts ID'd?
No
Assign advisors & team leads
Yes
Min Reqs?
No
Define minimum security criteria
Yes
Bug Track?
No
Specify bug/work tracking tool
Yes
Quality Gates?
No
Specify quality gates & bug bars
Yes
Assessed Risk?
No
Use SRA/ PRA to codify risk
Yes
Design
Design Reqs?
No
Perform all subtasks
Yes
Security
No
Consult advisors for review
Yes
Privacy
No
Consult advisors for review
Yes
Crypto
No
Consult advisors for review
Yes
Attack Surface?
No
Layered defenses & least privilege
Yes
Threat Models?
No
Assess threats using STRIDE
Yes
Implementation
Tools ID'd?
No
Specify compilers, tools, flags & options
Yes
Unsafe APIs?
No
Ban bad functions & APIs
Yes
Static Analysis?
No
Perform periodic static code analysis
Yes
Verification
Dynamic Analysis?
No
Conduct runtime verification tests
Yes
Fuzz Tests?
No
Fuzz all program interfaces
Yes
TM/ASR Review?
No
Validate models against code complete project
Yes
Pen Tests? (Option)
No
Deliberate attack testing on critical components
Yes
Release
Response Plan?
No
Document emergency response procedures
Yes
Final Security Review?
No
Review all security & privacy activities
Yes
Release Archive?
No
Archive all pertinent technical data
Yes
Response
END

Considerações Finais

- ✓ Quanto mais cedo forem incorporados os controles de segurança ao processo de desenvolvimento de software, menores os riscos de incidentes, melhor será o processo;

- ✓ Convém que o processo de integração entre segurança da informação e o desenvolvimento de software seja flexível.

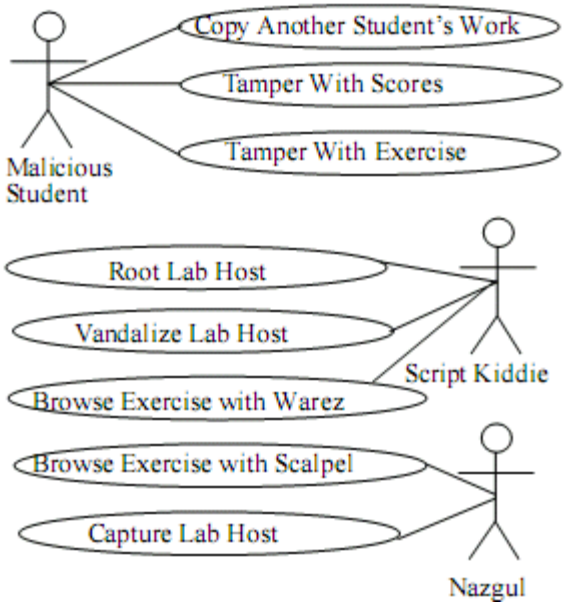
Considerações Finais



Considerações Finais

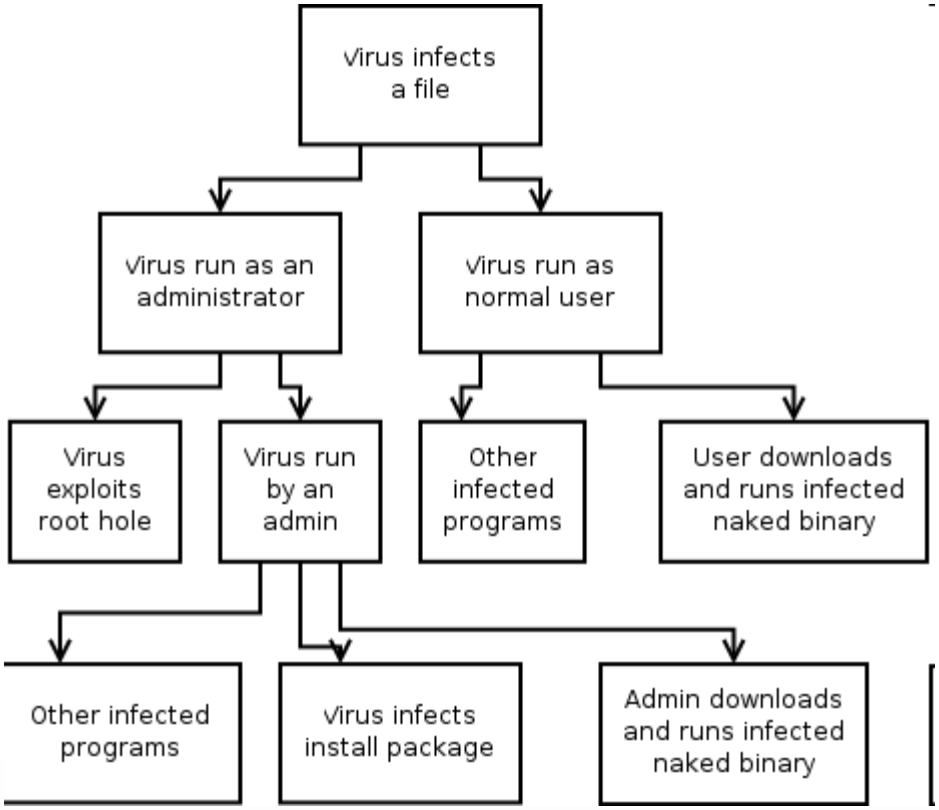
- ✓ Conjunto de práticas que podem auxiliar ao ciclo de vida de desenvolvimento de software seguro:
  - ✓ Systems Security Engineering Capability Maturity Model (SSE-CMM);
  - ✓ Software Assurance Forum for Excellence in Code (SAFEcode);
  - ✓ Security Quality Requirements Engineering (SQUARE);
  - ✓ Security Requirements Engineering Process (SREP);
  - ✓ Common Criteria (CC) ;

Anexo A – Abuse Case

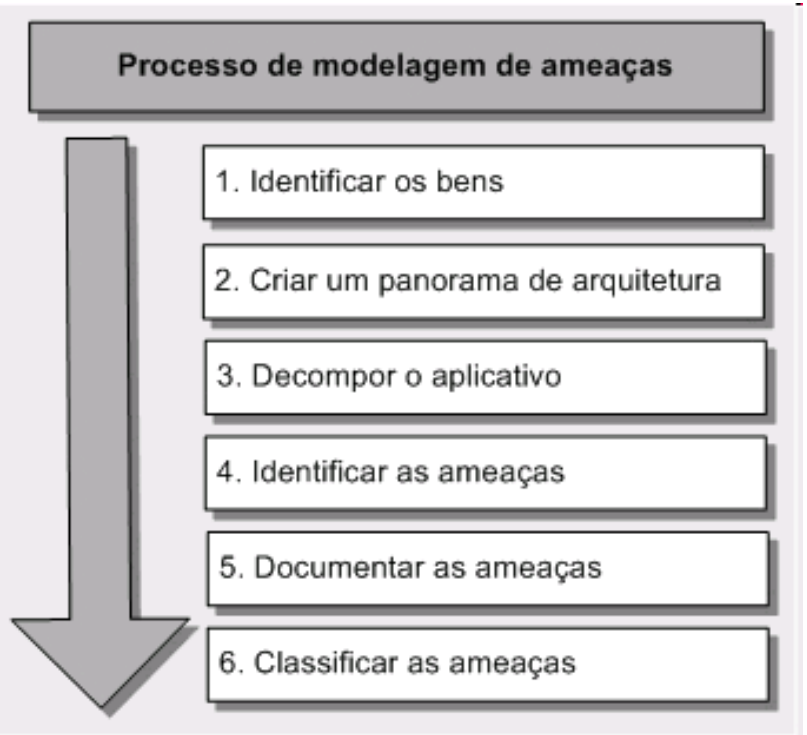


Generic Misuse Case: Spoof User Access	
Summary: The misuser successfully makes the system (physical / human / computerized) believe he is a legitimate user, thus gaining access to a restricted system / service / resource / building.	
Preconditions: 1) The misuser has a legitimate user's valid means to identify and authenticate ***OR*** 2) The misuser has invalid means to identify and authenticate, but so similar to valid means that the system is unable to distinguish (even if operating at its normal capabilities) ***OR*** 3) The system is corrupted to accept means of identification and authentication that would normally have been rejected. The misuser may previously have performed misuse case "Tamper with system" to corrupt the system.	
Misuser interactions	**System interactions**
Request access / service	
	Request identification and authentication
Misidentify and misauthenticate	
	Grant access / provide service
Postconditions: 1) The misuser can do anything the legitimate user could have done within one access session ***AND*** 2) In the system's log (if any), it will appear that the system was accessed by the legitimate user.	

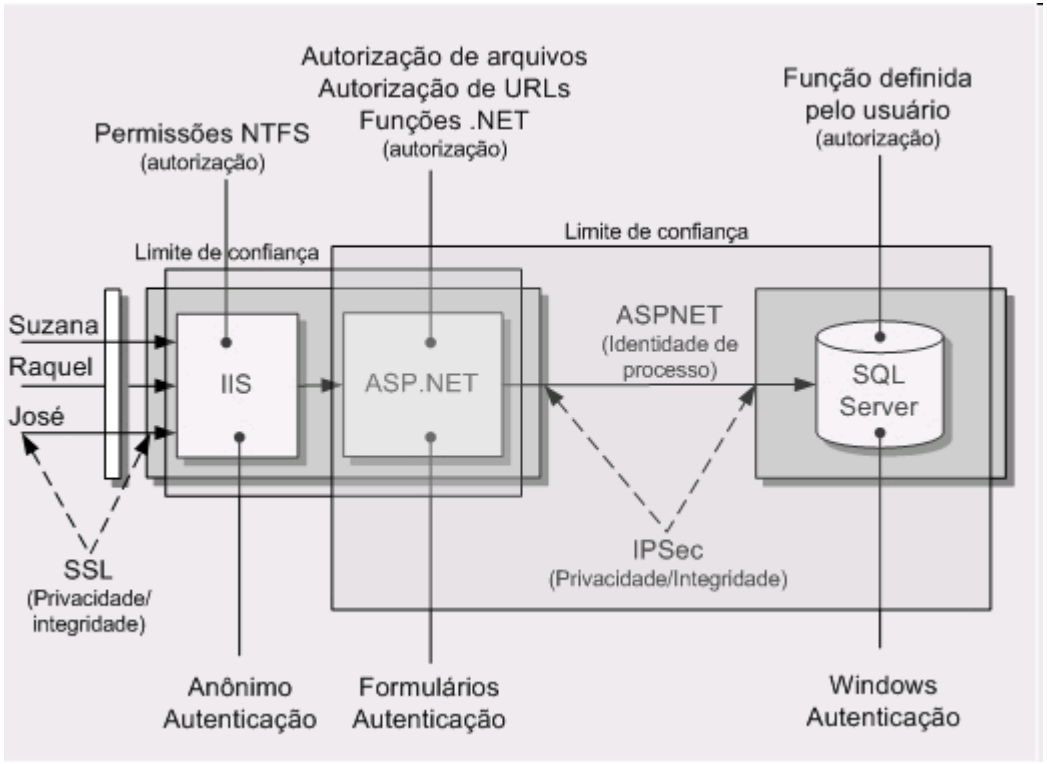
Anexo B – Attack Trees



Anexo C – Modelagem de Ameaças



Anexo D – Modelagem de Ameaças



Referências