Gestão de Vulnerabilidades em Aplicações

AULA 07 – SEGURANÇA DA INFORMAÇÃO

Conceito de Vulnerabilidade

- Vulnerabilidade: falha ou fraqueza em software/sistema

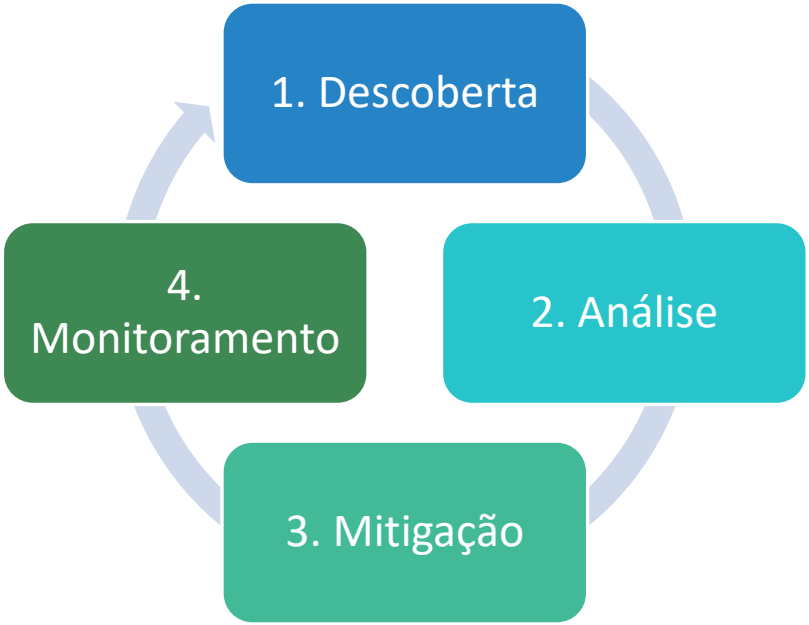
- Exploit: técnica/código que explora a falha

- Risco: impacto + probabilidade de exploração

Exemplo Prático

- Vulnerabilidade: formulário sem validação
- Exploit: ataque de SQL Injection
- Impacto: roubo de credenciais e dados

Ciclo de Gestão de Vulnerabilidades



Processo contínuo e cíclico

Etapa 1: Descoberta

- Scanners automáticos (ZAP, Nessus, Nikto)
- Pentests manuais
- Relatórios de fornecedores
- Programas de bug bounty

Etapa 2: Análise

- Classificação pelo CVSS (Common Vulnerability Scoring System)

- Avaliação de criticidade

- Impacto no negócio

- Definição de prioridades

Etapa 3: Mitigação

- Aplicar patches e updates

- Corrigir configurações

- Usar WAFs (Web Application Firewall) ou controles temporários

- Políticas de segurança

Etapa 4: Monitoramento

- Revisar vulnerabilidades periodicamente

- Acompanhar novos CVEs (Common Vulnerabilities and Exposures)

- Automação de alertas

- Auditorias regulares

OWASP Top 10

- A01: Quebra de Controle de Acesso
- A02: Falhas Criptográficas
- A03: Injeção
- A05: Configuração Incorreta
- A07: Falhas de Autenticação

Exemplo OWASP

A03 – Injeção

- SQL Injection

- LDAP Injection

- Command Injection

Impacto: acesso indevido, roubo de dados, execução de comandos

Ferramentas de Análise

- OWASP ZAP – open source

- Burp Suite – testes avançados

- Syhunt Hybrid – web/mobile/APIs

- Nikto – servidores web

Hardening em Aplicações (1/2)

- Validação de entradas
- Uso de prepared statements
- Configurações seguras em servidores
- Senhas seguras (bcrypt, Argon2)

Hardening em Aplicações (2/2)

- Autenticação multifator (MFA)
- Least privilege
- Monitoramento de logs
- Rotação de chaves e certificados

Estudo de Caso: SQL Injection

Consulta vulnerável:

```
SELECT * FROM usuarios WHERE usuario = 'admin' AND senha = '1234';
```

Exploit: ' OR '1'='1

Resultado: acesso não autorizado

Mitigação de SQL Injection

- Prepared statements
- ORMs seguros
- Contas de banco com permissões mínimas
- Firewall de Aplicação Web (WAF)

Checklist de Segurança

- ❑ Revisão contra OWASP Top 10
- ❑ Entradas validadas
- ❑ Hash seguro de senhas
- ❑ Frameworks atualizados
- ❑ Logs e monitoramento
- ❑ MFA para admins

Atividade em Grupo

Cenário: API REST desenvolvida por um dev júnior

- Usa bibliotecas pouco conhecidas e um framework desatualizado

- Endpoints públicos sem autenticação/validação adequada

- Dependências sem versionamento ou auditoria

Tarefas:

- 1. Identificar pelo menos 4 vulnerabilidades específicas.
- 2. Sugerir ferramentas e métodos para descoberta.
- 3. Propor plano de mitigação.
- 4. Criar checklist de segurança para APIs.

Entregável: documento curto (1 página) com vulnerabilidades encontradas, evidências simples e ações priorizadas.

Leituras Recomendadas

- OWASP Top 10 – https://owasp.org
- OWASP Testing Guide
- The Web Application Hacker’s Handbook
- Documentação: Burp Suite e ZAP
- Banco CVE: https://cve.mitre.org