Insper

Avaliação Final Tecnologias Hackers

Rodolfo Avelino e João Eduardo

Opção 1: Ferramenta de Avaliação de Segurança de Aplicações Web.

Objetivo:

Desenvolver uma ferramenta capaz de realizar a avaliação de segurança automatizada em aplicações web, identificando vulnerabilidades comuns descritas no OWASP Top 10.

O projeto deve demonstrar conhecimento prático sobre segurança da informação, testes de penetração automatizados, desenvolvimento seguro e documentação técnica.

Descrição Geral

A ferramenta deverá permitir a varredura de URLs de aplicações web para detecção de vulnerabilidades, como:

- Injeções (SQLi, Command Injection)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Directory Traversal e File Inclusion
- Exposure de informações sensíveis

Os alunos deverão desenvolver scripts próprios ou integrar APIs open-source (como OWASP ZAP, Nmap, Wfuzz, ou Nikto), além de apresentar um relatório técnico e uma interface de visualização dos resultados.

Insper

Critérios de Avaliação

Conceito C (Nota C) – Ferramenta Básica

- Implementa varredura simples sobre URLs e parâmetros.
- Identifica e relata vulnerabilidades básicas (ex: XSS ou SQL Injection simples).
- Interface textual ou script de linha de comando.
- Relatório básico com logs e resumo dos testes realizados.

Conceito B (Nota B) – Automação e Integração

- Inclui detecção de múltiplas vulnerabilidades (mínimo 4 das OWASP Top 10).
- Automação da varredura com execução por linha de comando ou interface web simples.
- Relatórios automáticos em formato JSON, CSV ou Markdown.
- Uso de ferramentas auxiliares (ZAP API, Nikto, Nmap, etc.).

Conceito A (Nota A) – Análise Avançada e Dashboard

- Implementa análise heurística e priorização de vulnerabilidades com base em severidade.
- Interface web interativa com dashboard (gráficos, score de risco, filtros).
- Geração de relatórios detalhados com recomendações de mitigação.
- Integração com sistema de autenticação (usuários/empresas simuladas).
- Utilização de containerização (Docker).
- Demonstração em vídeo funcional da ferramenta analisando uma aplicação vulnerável.

Insper

Entregáveis

Os alunos deverão disponibilizar no GitHub:

1. Código-fonte completo da ferramenta.
2. Scripts de automação e integração (CI/CD).
3. Relatório técnico contendo:
    - Descrição do sistema e da arquitetura utilizada.
    - Metodologia de testes.
    - Resultados obtidos e exemplos de vulnerabilidades detectadas.
    - Sugestões de mitigação.
4. Diagrama de arquitetura e fluxograma de funcionamento.
5. Vídeo demonstrativo (até 7 minutos) mostrando a execução da ferramenta em um caso real.

Estrutura esperada

```
/src
  ├── scanner.py
  ├── report_generator.py
  ├── utils/
  ├── tests/
  └── requirements.txt
/docs
  ├── architecture_diagram.png
  ├── flowchart.pdf
.github/workflows
  └── security_scan.yml
README.md
Dockerfile
```

Insper

Sugestões de Ferramentas e Tecnologias

- **Linguagens:** Python, Go, Node.js
- **Frameworks:** Flask, FastAPI, Express.js
- **Ferramentas:** OWASP ZAP, Nikto, Wfuzz, Nmap, Sublist3r
- **Visualização:** Grafana, Streamlit, Dash, Chart.js
- **Infraestrutura:** Docker, GitHub Actions, Terraform

Insper

Opção 2: Sistema de Detecção de Ameaças Cibernéticas em Servidores Web

O objetivo desta tarefa é desenvolver um sistema capaz de identificar e classificar ameaças cibernéticas em servidores web a partir de dados coletados. O sistema deverá passar por diferentes etapas, como coleta de dados, pré-processamento, análise e geração de alertas. Além disso, o projeto deverá incluir uma interface web para a visualização clara e intuitiva dos dados processados e dos alertas gerados.

- Coleta de dados: Capturar logs detalhados de acesso ao servidor web, que fornecerão informações cruciais sobre as requisições feitas ao sistema.
- Pré-processamento: Aplicar técnicas para limpar e preparar os dados, removendo valores anômalos e gerando atributos relevantes, como o tamanho das requisições, o número de parâmetros e a presença de padrões potencialmente suspeitos.
- Análise de dados: Implementar métodos de classificação para identificar requisições normais ou maliciosas, utilizando um modelo baseado em dados de ataques conhecidos.
- Geração de alertas: Desenvolver uma solução para o armazenamento dos dados processados e a geração de alertas em tempo real, com a criação de uma interface web para visualização dos relatórios e atividades suspeitas de forma acessível.

Conceito C (Nota C):

- Coleta de Dados Básica:
  O sistema deve ser capaz de coletar logs de acesso do servidor web de maneira eficaz, garantindo a captura de dados relevantes como endereço IP, requisições HTTP e status de resposta.
- Pré-processamento de Dados Simples:
  Implementar um processo básico de limpeza de dados, removendo ruídos e

valores ausentes, e gerando pelo menos um atributo relevante para a análise (por exemplo, tamanho da requisição).

Conceito B (Nota B):

- Coleta e Pré-processamento Avançado:
  Além de capturar logs detalhados, o sistema deve realizar um pré-processamento mais robusto, como a remoção de outliers e a criação de múltiplos atributos relevantes (por exemplo, número de parâmetros na requisição e presença de padrões suspeitos nas URLs).
- Análise de Dados com Classificação Básica:
  Implementar uma análise que use métodos de classificação simples para identificar requisições maliciosas, baseando-se em um conjunto de regras ou um modelo básico treinado com dados de ataques conhecidos.

Conceito A (Nota A):

Além das funcionalidades dos conceitos C e B, o sistema deve:

1. Base de Dados (Logs de Servidor Web):
   O sistema deve utilizar logs ou simulados de um servidor web (como Apache, Nginx ou IIS) contendo dados como: endereço IP de origem, requisições HTTP e parâmetros, códigos de status (200, 403, 404, 500 etc.), data e hora de acesso e user agent (quando disponível). Pode ser utilizado o gerador de log Apache Log Generator ou scripts Python.
2. Implementar Regras de Correlação e Priorização de Eventos:
   Detectar comportamentos suspeitos combinando múltiplos fatores, como um mesmo IP com alta frequência de requisições e erros 403/404 repetidos.
   Classificar automaticamente os alertas por nível de criticidade (baixo, médio, alto).

3. Interface de Monitoramento Avançada:
   Desenvolver uma interface web com dashboard interativo exibindo volume de requisições por IP, tipos de ameaças detectadas e histórico de alertas.
   A interface deve permitir filtragem e busca por IP, endpoint ou tipo de ataque.
4. Documentação e Demonstração:
   Documentar as regras de detecção, o formato dos logs utilizados e a arquitetura do sistema.
   Apresentar por meio de vídeo o funcionamento do sistema com logs reais ou simulados, demonstrando a geração de alertas e o uso do painel de monitoramento.

Insper

Opção 3: Ferramenta para detecção de Phishing

Exemplo Prático:

- **Análise de URLs e páginas web:** Examinar URLs e conteúdo de páginas para identificar sinais de phishing (domínios falsos, certificados SSL inválidos, conteúdo clonado)
- **Comparação com bases de dados:** Verificar URLs contra listas de domínios maliciosos conhecidos (ex: PhishTank, Google Safe Browsing API)
- **Avaliação de características técnicas:** Analisar metadados como idade do domínio, informações de DNS, registros WHOIS, certificados SSL
- **Análise de conteúdo:** Verificar presença de formulários de login, solicitações de informações sensíveis, logos e imagens de marcas conhecidas

Conceito C (Nota C):

Verificação Básica de URLs:

- Verificar se o domínio está em listas de phishing conhecidas (PhishTank, OpenPhish)
- Identificar características básicas suspeitas como:
    - Presença de números em substituição a letras no domínio
    - Uso excessivo de subdomínios
    - Presença de caracteres especiais na URL

Exibição Web Simples:

- Página web básica que exibe resultados da análise em formato de tabela simples
- Interface para inserção de URLs a serem verificadas
- Indicador visual básico (verde/vermelho) para URLs seguras/maliciosas

Insper

Conceito B (Nota B):

Análise Heurística Avançada:

- Implementar todas as verificações do conceito C
- Análise de idade do domínio através de consultas WHOIS
- Verificação de uso de DNS dinâmico (ex: domínios no-ip, dyndns)
- Análise de certificados SSL (emissor, data de expiração, coincidência entre domínio e certificado)
- Detecção de redirecionamentos suspeitos
- Verificação de similaridade com domínios de marcas conhecidas usando distância de Levenshtein
- Análise básica de conteúdo para detectar formulários de login e solicitações de informações sensíveis

Interface Web Interativa:

- Dashboard com visualização detalhada dos resultados da análise
- Histórico de URLs verificadas com opção de exportação
- Gráficos simples mostrando distribuição de características suspeitas
- Explicações sobre cada característica analisada e por que representa um risco

Conceito A (Nota A):

Plugin para Navegador Web com Verificação em Tempo Real:

- **Integração com navegador:** Plugin para Firefox que verifica URLs enquanto o usuário navega
- **Monitoramento ativo:** Verificação de todas as páginas visitadas e links clicáveis em tempo real

- **Notificações em tempo real:** Alertas visuais quando uma página suspeita é detectada
- **Bloqueio preventivo:** Opção para bloquear automaticamente o acesso a páginas identificadas como phishing
- **Personalização:** Permitir que o usuário defina o nível de sensibilidade das análises e crie listas de sites confiáveis (whitelist)
- **Análise de links em e-mails:** Verificação de links ao passar o mouse sobre eles antes mesmo de clicar

Sistema Web Avançado com Machine Learning:

- **Dashboard analítico:** Interface web completa com visualizações detalhadas e relatórios
- **Análise de screenshots:** Captura de screenshots das páginas para comparação visual com sites legítimos
- **Machine learning:** Uso de modelos pré-treinados para classificação de URLs com base em múltiplas características:
    - Comprimento da URL
    - Número de subdomínios
    - Presença de caracteres especiais
    - Idade do domínio
    - Informações de registro WHOIS
    - Presença de palavras-chave associadas a phishing
    - País de hospedagem (verificado por geolocalização IP)
- **Análise da reputação do host:** Verificar histórico do servidor/IP em listas de reputação
- **Verificação de legitimidade de serviços OAuth:** Detectar aplicações OAuth falsas que solicitam permissões excessivas
- **Modelo de decisão explicável:** Apresentação clara dos fatores que contribuíram para a classificação da URL

- **Avaliação de risco quantitativa:** Score de risco de 0 a 100 com detalhamento dos fatores contribuintes

Características Adicionais Avançadas:

- **Análise do código-fonte:** Detectar scripts maliciosos, técnicas de ofuscação ou obfuscação de código
- **Verificação de práticas SEO maliciosas:** Detectar técnicas como keyword stuffing, cloaking ou doorway pages
- **Análise de URLs encurtadas:** Expandir e analisar URLs encurtadas antes que o usuário acesse
- **Verificação de presença em listas negras de e-mail:** Consultar se o domínio está em blacklists de spam
- **Detecção de páginas de phishing zero-day:** Identificar novas ameaças através de aprendizado de máquina
- **Análise de comportamento do site:** Verificar se o site tenta:
    - Bloquear cliques direito do mouse
    - Ocultar a URL real
    - Utilizar técnicas de manipulação de DOM para falsificar elementos da interface
    - Utilizar temporizadores de urgência para pressionar o usuário