Insper

Avaliação Final Tecnologias Hackers

Rodolfo Avelino e João Eduardo

Opção 1: Projeto de Hospedagem de Ambiente de Desenvolvimento Seguro na AWS com Ênfase na Segurança e CI/CD.

Objetivo:

Avaliar a capacidade dos alunos em projetar, implementar e configurar um ambiente de desenvolvimento seguro, demonstrando conhecimento em arquitetura de sistemas distribuídos, segurança da informação e automação de infraestrutura.

Critérios de Avaliação:

A avaliação será realizada com base nos seguintes critérios:

- **Arquitetura:**
  - Separação clara entre ambientes (desenvolvimento, teste e produção).
  - Utilização adequada de serviços AWS (EC2, RDS, S3, IAM, CloudFormation, Terraform, entre outros).
  - Implementação de mecanismos de balanceamento de carga e alta disponibilidade.
- **Segurança:**
  - Implementação de controles de acesso baseados em papéis (IAM).
  - Criptografia de dados em trânsito e em repouso.
  - Configuração de grupos de segurança e políticas de rede.
  - Implementação de mecanismos de detecção de intrusão (IDS/IPS).
  - Realização de testes de penetração para identificar vulnerabilidades.
- **CI/CD:**
  - Automação do processo de build, testes e deploy.
  - Utilização de ferramentas de CI/CD (Jenkins, GitHub Actions, etc.).
  - Implementação de testes unitários e de integração.

- **Infraestrutura como Código:**
  - Utilização de ferramentas de IaC (Terraform, CloudFormation).
  - Modularização e versionamento do código de infraestrutura.
- **Documentação:**
  - Documentação clara e concisa da arquitetura, configurações e procedimentos.
  - Diagramas de arquitetura e fluxogramas.
  - Procedimentos de operação e manutenção.
- **Inovação:**
  - Utilização de tecnologias e ferramentas inovadoras.
  - Soluções criativas para os desafios do projeto.
- **Sustentabilidade:**
  - Otimização de custos e recursos.
  - Consideração do impacto ambiental.

Entrega:

Os alunos deverão entregar um relatório técnico detalhado em seu GitHub, incluindo:

- Descrição da arquitetura do sistema.
- Diagrama de arquitetura.
- Código fonte utilizado.
  - Scripts de configuração da infraestrutura.
  - Pipelines de CI/CD.
- Relatório de testes de segurança.
- Documentação completa do projeto.
- Vídeo demonstrativo do ambiente em funcionamento (até 7 minutos).

Avaliação:

A avaliação será realizada em uma escala de 0 a 10, considerando os seguintes aspectos:

- **Completude:** Todos os requisitos do projeto foram atendidos.

- **Qualidade:** A solução apresentada é de alta qualidade, bem estruturada e eficiente.
- **Inovação:** A solução apresenta elementos inovadores e criativos.
- **Documentação:** A documentação é clara, completa e bem organizada.
- **Apresentação:** A apresentação do projeto é clara e concisa.

Critério	Nota 0 - 4	Nota 5 - 6	Nota 7 - 8	Nota 9 - 10
Arquitetura	Projeto abaixo do esperado ou não entregue.	Instâncias separadas	Arquitetura modular, balanceamento de carga	Arquitetura resiliente, auto-escalável
Segurança	Projeto abaixo do esperado ou não entregue.	Configurações básicas	Implementação de controles de segurança avançados (WAF, IDS)	Testes de penetração bem-sucedidos
CI/CD	Projeto abaixo do esperado ou não entregue.	Pipeline básico	Pipeline automatizado com testes unitários e de integração	Pipeline com deploy contínuo e blue-green deployment
Infraestrutura como Código	Projeto abaixo do esperado ou não entregue.	Uso básico de uma ferramenta	Modularização e versionamento do código	Utilização de módulos e variáveis para customização
Documentação	Projeto abaixo do esperado ou não entregue.	Documentação básica	Documentação detalhada e organizada	Documentação automatizada com ferramentas como Sphinx
Inovação	Projeto abaixo do esperado ou não entregue.	Solução padrão	Utilização de tecnologias emergentes	Solução inovadora e disruptiva
Sustentabilidade	Projeto abaixo do esperado ou não entregue.	Consideração básica de custos	Otimização de custos e recursos	Análise de custo-benefício detalhada

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):

- Análise de Dados Avançada com Modelos Preditivos:
  Além das funcionalidades dos conceitos C e B, o sistema deve implementar um modelo de machine learning treinado com um conjunto robusto de dados de ataques cibernéticos. O modelo deve ser capaz de classificar requisições normais e maliciosas com um alto nível de acurácia.
- Geração de Alertas em Tempo Real e Visualização:
  O sistema deve gerar alertas em tempo real com base na análise de dados e fornecer uma interface visual detalhada, com dashboards que permitam monitorar a segurança dos servidores e identificar ameaças de forma intuitiva.

Insper

Opção 3: Plugin para Navegador Web Firefox para Bloqueio de Rastreadores

Exemplo Prático:

- **Identificação de rastreadores:** Utilizar uma lista de domínios de rastreamento conhecida (ex: EasyList) e comparar com os hosts das requisições.
- **Bloqueio de conteúdo:** Interceptar as requisições para os domínios de rastreamento e cancelar o envio.
- **Personalização:** Permitir que o usuário crie listas personalizadas de domínios a serem bloqueados.
- **Interface:** Criar uma interface simples e intuitiva para o usuário configurar o plugin.

Conceito C (Nota C):

- Identificação Básica de Rastreadores:
  O plugin deve ser capaz de identificar e listar rastreadores utilizando uma lista de domínios conhecida, comparando as requisições feitas pelo navegador com essa lista para detectar os domínios de rastreamento.
- Bloqueio de Conteúdo Básico:
  Implementar o bloqueio de requisições para domínios identificados como rastreadores, interrompendo a comunicação com esses hosts durante a navegação.

Conceito B (Nota B):

- Identificação e Bloqueio Avançados:
  Além de identificar e bloquear rastreadores utilizando uma lista de domínios conhecida, o plugin deve permitir a adição de listas

personalizadas, onde o usuário pode inserir ou remover domínios de rastreamento de forma manual.

- Interface Simples:
  Criar uma interface simples e intuitiva que permita ao usuário ativar/desativar o bloqueio de rastreadores e gerenciar suas listas de domínios personalizados.
- Relatório Básico:
  Exibir um relatório simples na interface mostrando os rastreadores bloqueados durante a navegação.

Conceito A (Nota A):

- Identificação, Bloqueio e Personalização Completa:
  Além de cumprir os requisitos dos conceitos C e B, o plugin deve permitir personalização avançada, com a criação e edição de múltiplas listas de bloqueio, e a capacidade de diferenciar entre domínios de rastreadores de primeira e terceira parte.
- Interface Avançada com Estatísticas:
  A interface do plugin deve ser aprimorada para incluir estatísticas detalhadas sobre o número de rastreadores bloqueados, bem como gráficos de uso que ajudem o usuário a entender o impacto do bloqueio sobre a navegação.
- Configurações Avançadas de Privacidade:
  O plugin deve oferecer opções avançadas para que o usuário configure regras específicas de bloqueio, como filtrar por tipos de conteúdo (cookies, scripts, etc.) ou determinar diferentes níveis de bloqueio.