Atividade 2 - Documentação Estruturada para Projeto de IA e Segurança

João Pedro Schmidt Cordeiro - 22100628

INE5448 - Tópicos Especiais em Aplicações Tecnológicas I

25 de setembro de 2025

Sumário

1 Documentos de Requisitos

1.1 Product Requirements Document (PRD)

Título PRD - Sistema de Detecção de Intrusões com IA (IDS-AI)

Versão 1.0

Data 28 de agosto de 2025

1.1.1 Propósito

O IDS-AI é um sistema de segurança de rede que utiliza inteligência artificial e aprendizado de máquina para monitorar, detectar e alertar sobre atividades maliciosas e anômalas em redes corporativas em tempo real. O objetivo é fornecer às equipes de segurança uma ferramenta proativa para identificar ameaças avançadas que contornam as defesas tradicionais.

1.1.2 Público-alvo

Empresas de médio e grande porte que necessitam de proteção robusta para sua infraestrutura de TI, especialmente aquelas que operam com dados sensíveis e estão sujeitas a regulamentações de segurança, como LGPD, GDPR, etc.

1.1.3 Funcionalidades Principais

1. **Análise de Tráfego em Tempo Real:** Monitoramento contínuo de pacotes de dados na rede para identificar padrões suspeitos com baixa latência.
2. **Detecção de Anomalias Baseada em ML:** Utilização de algoritmos de aprendizado de máquina não supervisionado para estabelecer uma linha de base ("baseline") do comportamento normal da rede e detectar desvios que possam indicar ataques *zero-day*.
3. **Alertas Inteligentes e Priorização:** Geração de alertas detalhados com classificação de severidade baseada no risco potencial, reduzindo a fadiga de alertas e permitindo que a equipe de segurança foque nas ameaças mais críticas.
4. **Integração com Fontes de Threat Intelligence:** Enriquecimento das detecções com dados de fontes externas de inteligência de ameaças para identificar Indicadores de Comprometimento (IoCs) conhecidos, como IPs e domínios maliciosos.
5. **Dashboard de Visualização e Relatórios:** Interface gráfica interativa que apresenta o estado da segurança da rede, detalhes dos alertas, e permite a geração de relatórios forenses para análise de incidentes.

1.1.4 Métricas de Sucesso

- **Taxa de Detecção de Ameaças Reais:** Percentual de ataques reais identificados com sucesso pelo sistema ($> 95\%$).
- **Taxa de Falsos Positivos:** Redução do número de alertas benignos classificados como maliciosos ($< 2\%$ do total de alertas).

- **Mean Time to Detect (MTTD):** Tempo médio para detectar uma ameaça desde o seu início (meta: < 5 minutos).
- **Adoção do Produto:** Número de equipes de segurança que integram ativamente o IDS-AI em seu fluxo de trabalho diário.

1.1.5 Considerações de Segurança Específicas

- **Privacidade de Dados:** O sistema deve ser capaz de anonimizar ou pseudoanonimizar dados sensíveis (payloads) para cumprir com a LGPD, sem comprometer a eficácia da detecção.
- **Segurança do Modelo de IA:** Implementar defesas contra ataques adversariais, como envenenamento de dados de treinamento e evasão de detecção.
- **Integridade dos Alertas:** Garantir que os alertas não possam ser manipulados ou suprimidos por agentes mal-intencionados.

1.2 Technical Requirements Document (TRD)

Título TRD - Sistema de Detecção de Intrusões com IA (IDS-AI)

Versão 1.0

Data 28 de agosto de 2025

1.2.1 Arquitetura Geral

O sistema será baseado em uma arquitetura de microsserviços para garantir escalabilidade e resiliência. Os componentes principais são:

- **Coletores de Dados (Sensors):** Agentes leves implantados em pontos estratégicos da rede (SPAN ports, TAPs) para capturar o tráfego de rede.
- **Engine de Processamento de Dados:** Serviço responsável por normalizar, enriquecer e extrair features dos dados de rede brutos.
- **Servidor de Modelos de ML:** API que hospeda os modelos de detecção treinados e realiza inferências em tempo real.
- **Módulo de Alerta:** Serviço que recebe as detecções, as prioriza e as envia para os canais configurados (SIEM, e-mail, Slack).
- **Interface de Usuário (Dashboard):** Aplicação web para visualização dos dados e gerenciamento do sistema.

1.2.2 Requisitos de IA

- **Algoritmos de Detecção de Anomalias:** Utilização de um ensemble de algoritmos, incluindo **Isolation Forest** (para eficiência) e **Autoencoders** (para detecções complexas).
- **Algoritmos para Tráfego Sequencial:** Uso de Redes Neurais Recorrentes (**LSTM**) para analisar sequências de pacotes e identificar ataques multi-etapa.

- **Frameworks:** Python com Scikit-learn, TensorFlow/PyTorch.
- **Retreinamento:** O pipeline de MLOps deve suportar o retreinamento automático dos modelos a cada 30 dias para se adaptar a mudanças no comportamento da rede.

1.2.3 Requisitos de Segurança

- **Autenticação e Autorização:** Acesso ao dashboard e APIs protegido por OAuth 2.0 com Role-Based Access Control (RBAC).
- **Criptografia:** Comunicação entre microsserviços e com sistemas externos via mTLS. Dados em repouso (logs, modelos) criptografados com AES-256.
- **Logs de Auditoria:** Todas as ações administrativas e acessos ao sistema devem ser registrados em logs imutáveis.

1.2.4 Requisitos de Performance

- **Latência de Detecção:** O tempo entre a captura de um pacote e a geração de um alerta deve ser inferior a 200 milissegundos.
- **Throughput:** Os coletores devem suportar uma taxa de captura de até 10 Gbps por sensor.
- **Escalabilidade:** A arquitetura deve suportar escalonamento horizontal para lidar com o aumento do volume de tráfego de rede.

1.2.5 Integrações Necessárias

- **SIEM:** Envio de alertas em formatos padronizados (CEF, LEEF) para plataformas como Splunk, QRadar e ArcSight.
- **Firewalls/SOAR:** Integração via API para permitir ações de resposta automatizadas, como o bloqueio de um endereço IP malicioso.
- **Bases de Dados de Threat Intelligence:** Conexão com APIs de feeds de inteligência (ex: VirusTotal, AbuseIPDB) via protocolo STIX/TAXII.

2 Documentação de Decisões Técnicas

2.1 Architectural Decision Record (ADR-001)

Título ADR-001: Seleção de Algoritmo para Detecção de Anomalias

Status Aceito

Data 28 de agosto de 2025

2.1.1 Contexto

A funcionalidade central do IDS-AI é a detecção de anomalias em tempo real. A escolha do algoritmo de Machine Learning inicial é uma decisão crítica que impactará diretamente a precisão, a performance, a complexidade de implementação e a manutenibilidade do sistema. Precisamos de um modelo que seja eficaz na identificação de desvios do comportamento normal da rede, computacionalmente eficiente para operar em tempo real e relativamente simples de implementar na primeira versão do produto (MVP).

2.1.2 Opções Consideradas

1. **Isolation Forest:**
    - *Descrição:* Um algoritmo de ensemble não supervisionado que isola anomalias em vez de criar um perfil de dados normais. É eficiente em datasets de alta dimensão.
    - *Prós:* Muito rápido para treinar e prever, baixo consumo de memória, bom desempenho em diversos cenários.
    - *Contras:* Menos eficaz na detecção de anomalias contextuais (que dependem de uma sequência de eventos).
2. **LSTM (Long Short-Term Memory) Autoencoder:**
    - *Descrição:* Uma rede neural recorrente usada para aprender o padrão sequencial de dados normais. Anomalias são detectadas quando a reconstrução de uma nova sequência tem um erro alto.
    - *Prós:* Excelente para dados temporais (fluxos de rede), capaz de capturar padrões complexos e ataques multi-etapa.
    - *Contras:* Computacionalmente caro, requer mais dados para treinamento, mais complexo de implementar e otimizar.
3. **One-Class SVM (Support Vector Machine):**
    - *Descrição:* Um algoritmo que aprende uma fronteira ao redor dos dados normais. Qualquer ponto que caia fora dessa fronteira é considerado uma anomalia.
    - *Prós:* Bem estabelecido e com base teórica sólida.
    - *Contras:* Não escala bem com o número de amostras, sensível a hiperparâmetros, pode ter dificuldade com estruturas de dados complexas.

2.1.3 Decisão

A decisão é adotar o **Isolation Forest** como o principal algoritmo para a detecção de anomalias no MVP do IDS-AI.

Justificativa: O Isolation Forest oferece o melhor equilíbrio entre performance e precisão para uma primeira versão. Sua alta velocidade de inferência é crucial para atender ao requisito de detecção em tempo real. A simplicidade de implementação permitirá uma entrega mais rápida de valor, validando a arquitetura central do produto. O LSTM Autoencoder será considerado para uma versão futura (v2.0) para adicionar uma camada de detecção de ameaças temporais mais sofisticada.

2.1.4 Consequências

- **Positivas:**
  - Rápido desenvolvimento e implantação do MVP.
  - Baixa sobrecarga computacional nos servidores de inferência.
  - Estabelece uma base sólida de detecção que pode ser facilmente expandida.
- **Negativas:**
  - O sistema pode não detectar ataques sutis que se desenrolam ao longo do tempo e dependem da ordem dos eventos.
  - A interpretabilidade dos resultados do Isolation Forest é limitada, o que pode dificultar a análise de causa raiz de alguns alertas.
  - Haverá a necessidade de alocar recursos no futuro para pesquisar e implementar a solução baseada em LSTM.

3 Experimentação com IA

Para a elaboração desta atividade, diversas ferramentas de IA foram empregadas como assistentes de desenvolvimento e pesquisa, incluindo Gemini 2.5 Pro [3], ChatGPT 5 [6], NotebookLM [4] e o ambiente de desenvolvimento Cursor [1].

3.1 Engenharia de Prompts para Documentação

3.1.1 Prompt Estruturado Utilizado

O prompt a seguir foi fornecido a uma ferramenta de IA (GPT-4[5]) para gerar a seção "Procedimentos de Resposta a Incidentes"do TRD. O procedimento segue as 6 fases do ciclo de vida de resposta a incidentes do NIST[2].

Listing 1: Prompt para Geração de Seção do TRD

```
# Persona
Aja como um Engenheiro de Segurança Sênior especializado em
   resposta a incidentes e automação (SOAR).

# Contexto
Estou criando o Documento de Requisitos Técnicos (TRD) para um novo
   Sistema de Detecção de Intrusões baseado em IA (IDS-AI). O
   sistema monitora o tráfego de rede em tempo real, detecta
   anomalias usando Machine Learning e se integra a sistemas SIEM e
   firewalls via API. O público-alvo são equipes de segurança de
   médias e grandes empresas.

# Tarefa
Gere uma seção detalhada para o TRD chamada "Procedimentos de
   Resposta a Incidentes". A seção deve descrever um plano de
   resposta padrão que possa ser iniciado a partir de um alerta
   crítico gerado pelo IDS-AI.

# Requisitos do Conteúdo
O procedimento deve seguir as 6 fases do ciclo de vida de resposta
   a incidentes do NIST:
1.  **Preparação:** O que a equipe precisa ter em vigor ANTES de um
   incidente.
2.  **Identificação:** Como confirmar e analisar um alerta do
   IDS-AI.
3.  **Contenção:** Ações imediatas para limitar o dano.
4.  **Erradicação:** Como remover a causa raiz da ameaça.
5.  **Recuperação:** Passos para restaurar os sistemas à operação
   normal.
6.  **Lições Aprendidas (Post-Incident):** Como melhorar o processo
   após o incidente.

# Formato
Use títulos claros para cada fase. Dentro de cada fase, utilize
   bullet points para descrever as ações específicas. Seja objetivo
   e técnico.
```

3.1.2 Resposta Gerada pela IA

— Início da Resposta Gerada —

Procedimentos de Resposta a Incidentes

Esta seção descreve o fluxo de trabalho padrão para responder a um alerta de alta severidade gerado pelo IDS-AI, alinhado com as melhores práticas do setor.

1. **Preparação**
   - Manter uma lista de contatos atualizada da equipe de resposta a incidentes (CSIRT).
   - Garantir que os analistas de segurança tenham acesso e permissões necessárias nas ferramentas integradas (SIEM, Firewall, EDR).
   - Desenvolver e manter playbooks de resposta para os tipos mais comuns de alertas (ex: detecção de malware, exfiltração de dados, varredura de rede).
   - Realizar treinamentos e simulações de incidentes periodicamente.
2. **Identificação**
   - Ao receber um alerta crítico do IDS-AI, o analista de plantão (on-call) inicia a triagem em até 15 minutos.
   - Correlacionar o alerta com outras fontes de dados no SIEM para obter contexto (logs de firewall, autenticação, etc.).
   - Analisar os detalhes do alerta no dashboard do IDS-AI: pacotes capturados, IPs de origem/destino, e a justificativa da anomalia pelo modelo de ML.
   - Validar se o alerta é um verdadeiro positivo. Se confirmado, declarar um incidente formal e escalar para a equipe sênior.
3. **Contenção**
   - Isolar os ativos afetados da rede para prevenir a propagação da ameaça.
   - Utilizar a integração do IDS-AI com o firewall para bloquear os endereços IP de origem maliciosos via API.
   - Desativar contas de usuário comprometidas, se aplicável.
   - Preservar evidências voláteis (memória RAM) e não voláteis (discos) dos sistemas afetados para análise forense.
4. **Erradicação**
   - Identificar e remover a causa raiz do incidente (ex: remover o malware, corrigir a vulnerabilidade explorada).
   - Realizar uma varredura completa nos sistemas afetados para garantir que não há persistência do atacante.
   - Aplicar patches de segurança e fortalecer as configurações dos sistemas para prevenir a reinfecção.
5. **Recuperação**
   - Restaurar os sistemas a partir de backups limpos e verificados.
   - Monitorar de perto os sistemas restaurados para garantir que estão operando normalmente e que a ameaça não retornou.
   - Reintroduzir gradualmente os sistemas na rede de produção após validação.

6. Lições Aprendidas (Post-Incident)

- Conduzir uma reunião de post-mortem em até 5 dias úteis após a resolução do incidente.
- Documentar a linha do tempo do incidente, o impacto, as ações tomadas e as lições aprendidas.
- Criar itens de ação para melhorar as defesas, ferramentas e procedimentos (ex: ajustar a sensibilidade do modelo de ML, criar uma nova regra de detecção no SIEM).
- Atualizar os playbooks de resposta a incidentes com base nos aprendizados.

— Fim da Resposta Gerada —

3.1.3 Análise Crítica

- **O que funcionou bem:** A IA compreendeu perfeitamente a estrutura solicitada (as 6 fases do NIST) e a persona de um engenheiro de segurança. O conteúdo gerado é tecnicamente sólido, relevante e bem organizado. A linguagem é apropriada para um documento técnico. A IA também conseguiu conectar as ações ao contexto do IDS-AI (ex: "analisar os detalhes do alerta no dashboard do IDS-AI").
- **O que precisou ser ajustado:** A resposta é genérica por natureza, o que é esperado. Para um documento real, seria necessário adicionar detalhes específicos da organização, como nomes de ferramentas (ex: "bloquear IPs no Firewall *Palo Alto*"), nomes de equipes e SLAs (Service Level Agreements) exatos (ex: "iniciar triagem em *exatos* 15 minutos"). A resposta é um excelente *template*, mas a customização humana é indispensável para torná-la acionável.

3.2 IA como Base de Conhecimento

3.2.1 Prompt Hipotético para IA

Um desenvolvedor, de posse do PRD e do TRD criados, poderia usar o seguinte prompt em uma ferramenta de IA avançada (como o GitHub Copilot Enterprise, que pode ser contextualizado com a base de código e documentação do projeto) para acelerar a implementação.

Listing 2: Prompt Hipotético de Implementação

```
# Contexto
Estou trabalhando no projeto IDS-AI. Tenha como base o PRD e o TRD
   fornecidos.

# Tarefa
Gere um esqueleto de código em Python para a funcionalidade
   "Alertas Inteligentes e Priorização".

# Requisitos do Código
1. Crie uma classe `AlertManager`.
2. O construtor deve receber a URL da API do SIEM.
3. Implemente um método `process_detection(detection_data)` que
   recebe um dicionário com dados da detecção (ex: `{'source_ip':
   '...', 'threat_type': '...', 'confidence_score': 0.85}`).
```

```
4. Dentro deste método, implemente a lógica de priorização descrita
   no TRD (considere o `confidence_score` e o tipo de ameaça). A
   prioridade deve ser "Baixa", "Média", "Alta" ou "Crítica".
5. Crie um método privado `_send_to_siem(alert_payload)` que
   formate o alerta para o padrão CEF (Common Event Format) e o
   envie para a API do SIEM usando uma requisição POST.
6. Adicione docstrings e type hints ao código.
```

3.2.2 Como a Documentação Facilitaria a Resposta da IA

A documentação estruturada seria fundamental para que a IA gerasse uma resposta precisa e útil.

1. **O PRD forneceria o "porquê":** A seção *"Funcionalidades Principais"* do PRD confirmaria para a IA a importância estratégica dos "Alertas Inteligentes e Priorização", garantindo que a solução gerada esteja alinhada com os objetivos do produto.
2. **O TRD forneceria o "como":**
   - A seção *"Arquitetura Geral"* informaria à IA que existe um "Módulo de Alerta"e uma "Interface de Usuário", ajudando-a a entender onde a classe 'AlertManager' se encaixaria.
   - A seção *"Integrações Necessárias"* seria a mais crítica. Ela especifica a necessidade de integração com **SIEM** e o formato **CEF**. Sem essa informação, a IA poderia gerar um código genérico de log ou usar um formato incorreto.
   - A seção *"Requisitos de IA"* mencionaria os modelos e seus outputs (como um score de confiança), o que daria à IA o contexto necessário para implementar a lógica de priorização solicitada no prompt.

Em resumo, sem a documentação, a IA teria que fazer suposições. Com o PRD e o TRD, a IA age como um desenvolvedor júnior que recebeu especificações claras de um sênior, resultando em um código muito mais alinhado com as necessidades reais do projeto, economizando tempo de desenvolvimento e refatoração.

Referências

[1] Anysphere. Cursor. `https://cursor.sh/`, 2024. Acessado em: 28 de agosto de 2025.

[2] Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. Computer security incident handling guide. Technical Report SP 800-61 Rev. 2, National Institute of Standards and Technology, 2012.

[3] Google. Gemini 2.5 pro. `https://deepmind.google/technologies/gemini/`, 2024. Acessado em: 28 de agosto de 2025.

[4] Google. Notebooklm. `https://notebooklm.google.com/`, 2024. Acessado em: 28 de agosto de 2025.

[5] OpenAI. Gpt-4 technical report. `https://arxiv.org/abs/2303.08774`, 2023. Accessed: 2025-08-28.

[6] OpenAI. Chatgpt 5. https://openai.com/chatgpt, 2024. Acessado em: 28 de agosto de 2025.