Ferramenta de Detecção de Anomalias em Tráfego de Rede

Este projeto é uma ferramenta desenvolvida em Python para detectar anomalias em tráfego de rede, utilizando técnicas de Machine Learning (Isolation Forest). É ideal para análise de segurança e pode ser usado como base para um Trabalho de Conclusão de Curso (TCC) em Engenharia da Computação.

Funcionalidades

- **Captura e Extração:** Converte arquivos `.pcap` (gerados por Wireshark/TShark) em formato tabular (`.csv`) com features relevantes.
- **Pré-processamento:** Limpa, unifica e normaliza os dados para modelagem.
- **Detecção de Anomalias:** Aplica o algoritmo Isolation Forest para identificar padrões de tráfego anômalos.
- **Visualização:** Apresenta os resultados em um dashboard interativo no terminal, destacando as anomalias detectadas.
- **Sistema de Alertas Automatizados:** Envio automático de emails quando anomalias com alto risco são detectadas, configurável via arquivo `.env`.
- **Detalhamento e Contextualização das Anomalias:** Classificação enriquecida com heurísticas para facilitar a interpretação e a resposta do analista.

Estrutura do Projeto

```
TCC_Network_Anomaly_Detector/
├── data/
│   ├── simulated_traffic.csv       # Dados de exemplo para teste
│   └── ...                         # Local para arquivos .pcap e .csv
├── pcap_converter.py               # Módulo extração/conversão .pcap para .csv
├── anomaly_detector.py             # Módulo pré-processamento e detecção
├── main.py                         # Script principal e interface do usuário
├── requirements.txt                # Dependências do projeto
└── README.md                       # Este arquivo
```

Pré-requisitos

- Python 3.8 ou superior
- Wireshark (que inclui o utilitário `tshark`) com o executável disponível no PATH do sistema
- Acesso à internet para envio dos alertas por email

No Windows, certifique-se de adicionar o diretório do Wireshark (exemplo: `C:\Program Files\Wireshark`) às variáveis de ambiente do sistema.
No Linux (Debian/Ubuntu), instale o `tshark` com:

```
sudo apt install tshark
```

Instalação

1. Clone ou baixe o repositório.
2. Navegue até o diretório do projeto: `bash cd TCC_Network_Anomaly_Detector`
3. Instale as dependências Python: `bash pip install -r requirements.txt`

Configuração do Sistema de Alertas via Arquivo `.env`

Para enviar alertas automáticos por email, configure um arquivo `.env` na raiz do projeto com o seguinte conteúdo:

```
 EMAIL_SENDER=seu_email@gmail.com
EMAIL_PASSWORD=sua_senha_do_app
EMAIL_RECEIVER=email_destino@exemplo.com
SMTP_SERVER=smtp.gmail.com
SMTP_PORT=465
ALERT_SCORE_THRESHOLD=-0.19
```

- `EMAIL_SENDER`: Email remetente para envio dos alertas (Gmail recomendado).
- `EMAIL_PASSWORD`: Senha do app para o email remetente (especialmente se 2FA estiver habilitado).
- `EMAIL_RECEIVER`: Email destinatário que recebe os alertas.
- `SMTP_SERVER` e `SMTP_PORT`: Servidor SMTP e porta (padrão Gmail indicados).
- `ALERT_SCORE_THRESHOLD`: Limiar de score para disparar alertas (valores mais negativos indicam maior risco).

Como Usar

Preparar os Dados

- Coloque um arquivo `.pcap` (captura de rede via Wireshark/Tshark) ou `.csv` já processado na pasta `data/`.
- Certifique-se de que arquivos `.csv` contenham as colunas esperadas: `timestamp`, `src_ip, dst_ip, src_port, dst_port, protocol, packet_length`

Executar a Ferramenta

Execute o script principal:

```
 python main.py data/seuarquivo.pcap
# Ou
python main.py data/seuarquivo.csv
```

Teste com Dados Simulados

O projeto inclui o arquivo `simulated_traffic.csv` para testes rápidos:

```
 python main.py data/simulated_traffic.csv
```

Funcionamento Resumido

1. O arquivo é convertido e pré-processado para extração das características.
2. O algoritmo Isolation Forest realiza a detecção de anomalias no tráfego.
3. Cada anomalia recebe uma classificação detalhada por risco e contexto (tais como portas incomuns, IPs externos, pacotes grandes).
4. Resultado exibido em dashboard no terminal, incluindo as classificações.
5. Caso existam anomalias críticas (acima do limiar configurado), um email é enviado automaticamente para o destinatário configurado.

Desenvolvimento Futuro

- Análise em tempo real com monitoramento contínuo.
- Comparação e integração com outros algoritmos de Machine Learning.
- Desenvolvimento de interface gráfica (GUI/web).

- Ampliação dos canais de alerta (SMS, WhatsApp, sistemas de ticket).

Licença

Este projeto está licenciado sob a licença MIT. Consulte o arquivo LICENSE para mais detalhes.
