Relatório da Atividade de Laboratório 02 - Sistema de detecção de intrusão para redes CAN

Karen S. B. Silva[1], Luiz Henrique B. A. Silva[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732.970 – Recife – PE – Brazil

{ksbs@cin.ufpe.br, lhbas@cin.ufpe.br}

Abstract. *In this work, we propose an Intrusion Detection System (IDS) based on Long Short-Term Memory (LSTM) neural networks to analyze Controller Area Network (CAN) messages and identify potential attacks. Our system is designed to detect and classify four main types of attacks: Denial of Service (DoS), message spoofing with zero payload, message spoofing with minimal payload, and message replay attacks. The IDS is implemented on a Raspberry Pi 3, capable of real-time traffic classification on the CAN bus. Our contributions include the development of a communication-enabled ECU, the construction of a CAN traffic dataset, and the evaluation and deployment of the LSTM-based IDS, demonstrating its effectiveness in a real attack scenario.*

Resumo. *O foco deste trabalho é o desenvolvimento de um Sistema de Detecção de Intrusões baseado em redes neurais LSTM (Long Short-Term Memory) para analisar mensagens da rede CAN e identificar possíveis ataques. Nosso sistema é projetado para detectar e classificar quatro tipos principais de ataques: Negação de Serviço (DoS), spoofing de mensagens com payload zero, spoofing de mensagens com payload mínimo e ataques de repetição de mensagens. O IDS foi implementado em uma Raspberry Pi 3, capaz de classificar o tráfego em tempo real no barramento CAN. Nossas contribuições incluem o desenvolvimento de uma ECU comunicável, a construção de um conjunto de dados de tráfego CAN, e a avaliação e implantação do IDS baseado em LSTM, demonstrando sua eficácia em um cenário real de ataque.*

1. Introdução

A segurança nas redes de comunicação intra-veiculares, especialmente nas redes CAN (Controller Area Network), tem se tornado uma preocupação cada vez maior devido ao crescimento da conectividade e à complexidade dos sistemas automotivos. Uma das características centrais da rede CAN é sua forma de transmissão broadcast, onde todas as mensagens enviadas na rede são recebidas por todos os dispositivos ou ECUs conectados, o que permite uma comunicação eficiente entre esses dispositivos.

No entanto, essa conectividade também introduz vulnerabilidades. A transmissão broadcast torna a rede CAN suscetível a ataques maliciosos, como a injeção de mensagens falsas ou a interceptação de informações sensíveis. Com a maior adesão de carros conectados a redes externas e carros autônomos, a superfície de ataque potencial também aumenta, tornando os sistemas automotivos alvos atrativos para cibercriminosos. Portanto, garantir a segurança da rede CAN é essencial para que os veículos permaneçam protegidos contra ameaças externas.

Os sistemas de Detecção de Intrusão (IDS) são ferramentas fundamentais para proteger redes de comunicação, detectando atividades anômalas e possíveis ataques.

Estas IDS podem ser baseados em assinaturas, onde procuram padrões conhecidos de ataques, ou baseados em anomalias, onde identificam desvios do comportamento normal da rede. Recentemente, abordagens baseadas em aprendizagem de máquinas, que incluem redes neurais têm ganhado destaque por sua capacidade de detectar ataques desconhecidos e adaptar-se a novos padrões de ameaça.

Neste contexto, nosso trabalho propõe um Sistema de Detecção de Intrusão (IDS) baseado em redes LSTM (Long Short-Term Memory), uma técnica de rede neural recorrente particularmente eficaz para lidar com sequências de dados temporais, como as mensagens da rede CAN. Em resumo, as principais contribuições deste trabalho são:

- Desenvolvimento de uma ECU que se comunica com o barramento CAN e permite o deploy de um IDS baseado em aprendizagem de máquina;
- Construção de datasets de tráfego da rede CAN, incluindo tráfego benigno e malicioso, para treinamento do algoritmo de IDS;
- Avaliação de diferentes algoritmos de aprendizagem de máquina para determinar o mais eficaz, considerando métricas de detecção, tempo de execução e requisitos de armazenamento;
- Implementação e demonstração do IDS em um cenário real de ataque no barramento CAN.

2. Trabalhos Relacionados

A detecção de intrusões em redes Controller Area Network (CAN) tem atraído significativa atenção da comunidade de pesquisa devido à crescente conectividade dos veículos modernos e à necessidade de garantir a segurança cibernética automotiva. Desde então, surgiram muitos métodos de detecção de anomalias, com diferentes abordagens, incluindo métodos baseados em assinaturas, estatísticas e aprendizagem de máquina. Por exemplo, Khan et al. (2023) desenvolveram um método de detecção de intrusões baseado em estatística para redes CAN, onde foram otimizados os valores de limiar e o tamanho da janela temporal, visando minimizar as taxas de erro de detecção. O estudo investigou as características estatísticas dos ataques na rede CAN e propôs um modelo que utiliza uma abordagem de janela deslizante para detectar anomalias com alta precisão. No entanto, é importante notar que o método depende do uso de quadros de dados legítimos do barramento CAN para estabelecer valores de referência de normalidade, o que pode limitar sua capacidade de detectar intrusões se o ECU comprometido iniciar um ataque antes da estabilização desses valores.

3. Arquitetura Proposta

Em nosso trabalho, propomos um Sistema de Detecção de Intrusões (IDS) baseado em redes LSTM (Long Short-Term Memory) supervisionadas e não-supervisionadas para analisar as mensagens da rede CAN e identificar possíveis ataques. A arquitetura do sistema é composta pelas seguintes etapas: coleta e pré-processamento de dados, divisão dos dados, construção e treinamento dos modelos LSTM, avaliação dos mesmos e salvamento de ambos os modelos treinados.

Além disso, ele é projetado para detectar e classificar quatro tipos principais de ataques na rede CAN:

- **Random DOS**: O dispositivo atacante envia mensagens com ID e Payload aleatórios(de 1 a 4 bytes de tamanho(DLC)), com o objetivo de causar comportamento estranho nas ECUs.
- **Spoofing Zero Payload**: O dispositivo atacante coleta mensagens e as replicam, com exceção do Payload, pois todos os bytes do payload são transformados em zeros, com o objetivo de neutralizar operações das ECUs, como por exemplo: capturar uma mensagem de pressionar o pedal em 100% e enviar uma de 0% logo em seguida.
- **Zero DOS**: O dispositivo atacante causa entupimento de "zeros" no barramento, ou seja, ID e Payload(4 bytes de tamanho/DLC) iguais a zero, com o objetivo de se aproveitar da prioridade de bit 0 no barramento, fazendo com que nenhum bit 1 seja transmitido com sucesso.
- **Replay**: O dispositivo atacante coleta mensagens e as replica, com o objetivo de deixar o barramento mais cheio que o normal, consequentemente atrasando mensagens verdadeiras, além disso, o ataque também pode causar comportamentos estranhos dependendo de como a comunicação é configurada. Vale salientar que este ataque é especialmente difícil de ser detectado, pois todas as informações da mensagem são verdadeiras, mas a transmissão tem intenção maliciosa.

O sistema proposto segue um fluxo de trabalho bem definido para garantir uma detecção eficaz de intrusões na rede CAN. Este fluxo inclui várias etapas, desde a coleta e pré-processamento dos dados até a implantação dos modelos treinados.

Os dados utilizados para treinar e avaliar o modelo são provenientes de um setup experimental construído durante a atividade de laboratório 1. Antes de serem utilizados no treinamento do modelo, os dados foram submetidos a um processo de pré-processamento, que incluiu normalização e criação de janelas temporais.

Após isso, com os dados pré-processados em mãos, foi possível avançar para a etapa de construção e treinamento dos modelos LSTM supervisionados e não-supervisionados. Essa arquitetura de rede neural recorrente é especialmente adequada para lidar com sequências de dados, como as mensagens da rede CAN, pois utilizam uma estrutura de células de memória que permitem armazenar e acessar informações ao longo de muitas etapas de tempo, mantendo a relevância do contexto passado enquanto processam novas entradas.

Após o treinamento, o modelo é avaliado usando conjuntos de dados separados para validação e teste. Além disso, métricas de desempenho, como acurácia, recall e precisão, são calculadas para determinar a eficácia do modelo na detecção de ataques, sendo essencial para garantir a confiabilidade do sistema em condições reais de operação.

Finalmente, o modelo treinado é salvo e implantado em uma Raspberry Pi de forma a classificar o tráfego que está sendo transmitido no barramento CAN, permitindo uma detecção contínua e eficiente de possíveis ataques.

Com essa abordagem, esperamos fornecer uma solução eficaz para detectar e futuramente mitigar os diferentes tipos de ataques na rede CAN, contribuindo para a segurança dos veículos conectados.

4. Metodologia e Validação Experimental

4.1 Datasets

Para avaliar o sistema de detecção de intrusões (IDS) proposto, geramos dois conjuntos de dados que representam diferentes tipos de tráfego na rede CAN (Controller Area Network), obtidos a partir do setup experimental desenvolvido na atividade de laboratório 1. Esses dados incluem tanto tráfego benigno quanto tráfego malicioso, que foram escolhidos para cobrir uma variedade de cenários de ataque. Os dados maliciosos foram categorizados em quatro tipos de ataques: Random DOS, Spoofing Zero Payload, Zero DOS e Replay. A escolha desses dados se deve à relevância e à representatividade dos tipos de ataques mais comuns em redes CAN.

Os conjuntos de dados foram organizados da seguinte forma:

- **Dados Benignos**: Mensagens CAN normais sem qualquer interferência ou manipulação maliciosa.
- **Dados Maliciosos**: Incluem mensagens CAN que simulam três tipos específicos de ataque (não conseguimos coletar o quarto tipo de ataque, o Replay, por questões de problemas nas raspberrys pi): Random DOS, Spoofing Zero Payload e Zero DOS.

4.2 Pré-processamento e Dados de Treinamento, Validação e Teste

Utilizamos entre 3 e 5 milhões de mensagens em cada treinamento/validação/teste, Os dados foram pré-processados utilizando a técnica de normalização MinMaxScaler para garantir que todas as features tivessem valores dentro do mesmo intervalo, facilitando o treinamento do modelo. Em seguida, os dados foram segmentados em janelas de tempo com tamanho fixo de 150 amostras para capturar a sequência temporal das mensagens CAN.

Os conjuntos de dados foram divididos da seguinte forma:

Para o Modelo Supervisionado:

- Treinamento: 75% dos dados totais, usados para ajustar os parâmetros do modelo.
- Validação: 12.5% dos dados, usados para ajustar hiperparâmetros e evitar overfitting durante o treinamento.
- Teste: 12.5% dos dados usados para avaliar o desempenho final do modelo. A distribuição das classes nos conjuntos de dados foi equilibrada para garantir que o modelo não fosse tendencioso em relação a nenhuma das classes.

Para o Modelo Não-supervisionado:

Foram usadas dois tipos de proporções durante nossos treinamentos(cada treinamento teve uma dessas):

Primeira proporção:

- Treinamento: 40% dos dados benignos
- Validação: 10% dos dados benignos

- Teste: 50% dos dados benignos(diferentes dos anteriores) + 100% dos dados maliciosos

Segunda proporção:

- Treinamento: ~76% dos dados benignos
- Validação: ~19% dos dados benignos
- Teste: ~5% dos dados benignos + 100% dos dados maliciosos

4.3 Estrutura do Modelo e Experimentos Realizados

Implementamos dois tipos de modelos LSTM para detecção de anomalias. O modelo supervisionado inclui uma LSTM de 50 unidades, seguida por Dropout, BatchNormalization e uma camada densa com regularização L2. Enquanto o modelo não supervisionado é um autoencoder LSTM, composto por uma LSTM para codificação, seguida por RepeatVector e uma LSTM para decodificação, além de camadas de Dropout e BatchNormalization. Ambos foram treinados com o otimizador Adam (taxa de aprendizado de 0.004) e a função de perda MSE. Após isso, avaliamos os modelos nos conjuntos de teste usando métricas como acurácia, recall e precisão para uma análise detalhada do desempenho dos modelos em diferentes cenários de ataque.

4.4 Ambiente e Hardware Usados no Treinamento do Modelo

O ambiente onde nossos modelos foram desenvolvidos possuem as seguintes especificações: **Processador** Ryzen 5 7600; 6 Núcleos/12 Threads; Clock(base/max): 3,8 GHz ~ 5,1 GHz; 2 memórias RAM de 16 GB (32 GB total) de espaço; Frequência: 4800 MHz ~ 5600 MHz. **Placa-mãe** Tuf Gaming B650-Plus Wifi. **Armazenamento** Tipo SSD SNV2S/2000G NV2 Formato em M.2 2280 NVMe e Espaço de 2 TB; Versão PCIe: PCIe 4.0; Leitura até 3500MB/s e Gravação até 2800MB/s.
Linguagens/Frameworks: Python 3.12.3; Tensorflow 2.16.1; Keras 3.3.3.
Sistema operacional: Windows 11 Pro 23H2.

5. Resultados e Discussões

Comparado com o trabalho de Khan et al. (2023), que utilizou métodos estatísticos para detecção de intrusões em redes CAN, nosso método baseado em LSTM oferece uma abordagem diferenciada. Embora ambos os métodos apresentem alta precisão na detecção de anomalias, o nosso aborda a complexidade temporal dos dados da rede CAN com técnicas de aprendizado profundo, o que pode oferecer uma capacidade de generalização maior em comparação com abordagens puramente estatísticas. No entanto, nossos modelos também enfrentam desafios, como a necessidade de maior poder computacional e o tratamento de conjuntos de dados desbalanceados.

Após realizar o deploy do IDS em uma Raspberry pi, utilizando a acurácia, avaliamos as classificações de acordo com o comportamento do barramento em 5 situações, 1 para tráfego benigno e 4 para tráfegos com ataques ocorrendo sempre(1 para cada tipo de ataque). O IDS supervisionado estava classificando qualquer tráfego como malicioso, porém, o IDS não-supervisionado, com um threshold de **0,1328086**, conseguiu ter resultados mais satisfatórios:

Tráfego Benigno: Acurácia: 100%

Random DoS: Acurácia: 100%

Zero DoS: Acurácia 100%

Spoofing Zero Payload: Acurácia: 0%

Replay: Acurácia 0%

6. Conclusão e Trabalhos Futuros

Neste trabalho, desenvolvemos e avaliamos dois modelos baseados em LSTM para detectar anomalias em redes CAN, utilizando um conjunto de dados que inclui tráfego benigno e quatro tipos de ataques maliciosos. Os resultados mostraram que o nosso modelo não-supervisionado tem uma precisão de 0,99 para ataques zero e random DoS, mas não é muito eficiente para os demais. Concluímos que, apesar das limitações, os modelos baseados em LSTM são promissores para a detecção de anomalias em redes CAN, e trabalhos futuros podem explorar melhorias na detecção de ataques variados, bem como a combinação de abordagens supervisionadas e não supervisionadas para aumentar a robustez do sistema.

Referências

Pham, M., & Xiong, K. (2021). A survey on security attacks and defense techniques for connected and autonomous vehicles. Computers & Security, 102269. https://doi.org/10.1016/j.cose.2021.102269.

Tanksale, V. Intrusion detection system for controller area network. Cybersecurity 7, 4 (2024). https://doi.org/10.1186/s42400-023-00195-4.

W. Wu et al., "A Survey of Intrusion Detection for In-Vehicle Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 919-933, March 2020, https://doi.org/10.1109/TITS.2019.2908074.

Al-Jarrah, O. Y., Maple, C., Dianati, M., Oxtoby, D., & Mouzakitis, A. (2019). Intrusion detection systems for intra-vehicle networks: A review. IEEE Access, 7, 21266-21289. https://doi.org/10.1109/ACCESS.2019.2894183.

Aziz, Saddam & Faiz, Talib & Adegoke, Muideen & Loo, K.H. & Hasan, Kazi & Xu, Linli & Irshad, Muhammad. (2022). Anomaly Detection in the Internet of Vehicular Networks Using Explainable Neural Networks (xNN). Mathematics. 10. 1267. 10.3390/math10081267.

Aliwa, E., Rana, O., Perera, C., & Burnap, P. (2021). Cyberattacks and countermeasures for in-vehicle networks. ACM Computing Surveys (CSUR), 54(1), 1-37. https://doi.org/10.1145/3431233.

Khan, J.; Lim, D.-W.; Kim, Y.-S. Intrusion Detection System CAN-Bus In-Vehicle Networks Based on the Statistical Characteristics of Attacks. Sensors 2023, 23, 3554. https://doi.org/10.3390/s23073554.