Detecção de Ataques de Injeção de Dados no Tráfego de Rede de Sistemas ROS

Rodrigo Antunes, Bruno L. Dalmazo, Paulo Drews

[1] Centro de Ciências Computacionais
Universidade Federal do Rio Grande (FURG)

rodrigoantunes@furg.br, dalmazo@furg.br, paulodrews@furg.br

Abstract. *The Robot Operating System (ROS) is one of the most popular softwares for robotics development and research. However, studies are demonstrating several security problems in its structure. This work evaluates the application of intrusion detection techniques in the recognition of data injection attacks in these systems. A model was proposed, using the support vector machine algorithm, which was trained from the network traffic characteristics of a ROS application. Preliminary results showed an accuracy of about 92%.*

Resumo. *O Robot Operating System (ROS) é um dos mais populares softwares voltados ao desenvolvimento e pesquisa em robótica. Entretanto, estudos têm demonstrado diversos problemas de segurança em sua estrutura. Este trabalho avalia a aplicação de técnicas de detecção de intrusão no reconhecimento de ataques de injeção de dados nesses sistemas. Um modelo foi proposto, empregando um algoritmo de máquinas de vetores de suporte, o qual foi treinado a partir de características do tráfego de rede de uma aplicação ROS. Resultados preliminares mostraram uma acurácia de cerca de 92%.*

1. Introdução

O *Robot Operating System* (ROS) é um dos mais populares softwares voltados ao desenvolvimento e pesquisa em robótica. Ele é um *middleware* gratuito de código aberto que fornece abstração de hardware, controle de baixo nível de dispositivos, troca de mensagens entre processos, dentre outros. Ele estende as funcionalidades de um sistema operacional fornecendo serviços para as aplicações, sendo assim, uma camada de software acima do sistema operacional mas abaixo da aplicação que fornece uma abstração de programação comum entre diversos sistemas.

A representação fundamental de um sistema ROS é a de um grafo (ROS *graph*): uma rede ponto a ponto de processos composta pelo ROS master, nodos ROS, mensagens, tópicos, entre outros. Os nodos são processos que executam computação. Por exemplo, um nodo controla os motores das rodas, outro executa a localização, outro executa o planejamento do caminho, e assim por diante. Eles se comunicam entre si enviando mensagens para um tópico, podendo ser tanto assinantes (*subscribers*) quanto editores (*publishers*). Os nodos que assinam um tópico solicitam conexões de nodos que publicam esse tópico e estabelecem essa conexão por meio de um protocolo de conexão acordado. O protocolo mais utilizado no ROS é chamado TCPROS, que usa soquetes TCP/IP padrão. Já o ROS *master* armazena informações sobre tópicos e nodos disponíveis. Sem ele, os nodos não seriam capazes de se encontrar, trocar mensagens ou invocar serviços.

Atualmente, o uso de robôs na academia é dominado pelo ROS e sua adoção continua crescendo [Mayoral-Vilches et al. 2020]. Entretanto, diversos trabalhos na literatura apontam vulnerabilidades significativas em sua arquitetura, expondo usuários e ambiente ao risco de ataques com inúmeros efeitos colaterais, inclusive danos físicos [Amrouche et al. 2020], [Teixeira et al. 2020], [Rivera and State 2021]. Um ataque relevante nesse contexto é a injeção de dados, explorado pelo presente trabalho: um nodo pode publicar dados em um tópico qualquer sem autorização prévia, o que pode ser utilizado para injetar dados ou comandos na aplicação de forma a interferir no seu funcionamento. Em *Security for the Robot Operating System* [Dieber et al. 2017], o autor denomina esse ataque de "*Unauthorized Publishing*" (Publicação não Autorizada), o qual consiste em utilizar a API XML-RPC do ROS para inserir dados falsos na aplicação através do uso indevido de canais de comunicação. Em um trabalho correlato, com redes elétricas inteligentes, Rouzbahani *et al.* demonstraram que abordagens de aprendizado de máquina, mais especificamente SVM (*Support Vector Machine*), podem ser eficientes na classificação de ataques de injeção de dados em sistemas ciberfísicos [Mohammadi Rouzbahani et al. 2020].

Neste trabalho, é proposto um modelo para detectar anomalias e dar suporte ao reconhecimento do ataque de injeção de dados. A proposta utiliza um algoritmo de SVM, o qual foi treinado a partir de características do tráfego de rede de uma aplicação ROS. Resultados preliminares obtidos por meio de experimentos conduzidos em ambiente simulado mostraram a viabilidade da proposta.

2. Trabalhos Relacionados

Em [Dieber et al. 2017], os autores propuseram um servidor de autenticação (AS) dedicado garantindo que apenas nodos válidos façam parte da aplicação, os *publishers* e *subscribers* se registram no AS utilizando chaves públicas e privadas enquanto o AS autoriza ou desautoriza com base nesses registros. De maneira semelhante, [Breiling et al. 2017] propuseram um canal de comunicação seguro entre dois nodos que estende os canais de comunicação TCP e UDP existentes implementando autenticação, autorização, integridade e confidencialidade. Já em [Rivera and State 2021], foi apresentado um método que identifica vulnerabilidades em aplicações ROS e as codifica em regras de *firewall*, esse *firewall* então filtra os pacotes com base nestas regras. O presente trabalho complementa essas abordagens e pode se beneficiar dos mecanismos de segurança propostos nelas.

Outra abordagem é a aplicação de técnicas de detecção de intrusão. Uma proposta nesse sentido, focada em braços robóticos industriais com ROS, foi apresentada em [Narayanan and Bobba 2018], onde foi desenvolvido um sistema de detecção de anomalias empregando SVM. O sistema aprende o comportamento benigno a partir do estado das juntas, o que foge a este comportamento é considerado anomalia. Em [Amrouche et al. 2020], a proposta consiste na análise do fluxo de vídeo de carros autônomos com ROS utilizando redes neurais, reconhecendo imagens manipuladas por um atacante. Esses trabalhos focam em analisar o conteúdo das mensagens ROS (estado das juntas e fluxo de imagens), sendo aplicável a esses tipos de dados apenas. Por outro lado, o presente trabalho propõe um método mais abrangente, uma vez que foca na análise do tráfego de rede e não de um tipo de dado específico, podendo ser aplicada numa maior variedade de cenários e anomalias.

3. Arquitetura para Detecção de Anomalias em Robot Operating System

Neste trabalho, é proposta uma arquitetura utilizando detecção de intrusão baseada em anomalias no tráfego de rede. Esse tipo de abordagem tem se mostrado eficiente quando aplicado em sistemas ciber-físicos [Vasquez et al. 2017], podendo também ser aplicado em sistemas ROS. A Figura 1 ilustra uma visão geral da arquitetura proposta.

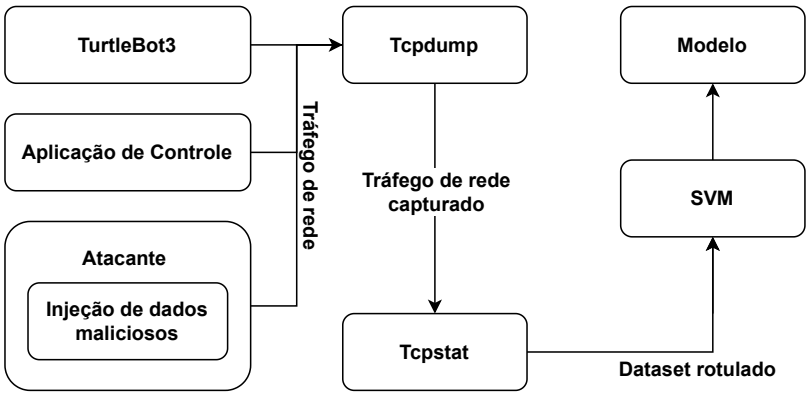

Figura 1. Arquitetura para Detecção de Anomalias em Robot Operating System.

A proposta consiste em treinar um modelo SVM a partir do tráfego de rede de uma aplicação de localização e mapeamento simultâneos (SLAM), aspecto fundamental da navegação autônoma de robôs móveis. O cenário é composto por um robô móvel *Turtlebot3* Burger[1] que se comunica via rede com sua *aplicação de controle* (responsável pela localização, deslocamento, desvio de obstáculos, mapeamento, etc). Incluído nesse cenário está o *atacante*, que injeta dados maliciosos nesse tráfego. Toda a comunicação envolvida é capturada, organizada e transformada em um *conjunto de dados rotulado*, o qual é utilizado para treinar o algoritmo SVM e gerar um *modelo* para reconhecer esses ataques. Detalhes da avaliação dessa arquitetura serão discutidos a seguir.

4. Avaliação

Para avaliar a viabilidade técnica da proposta, foram realizados experimentos no simulador de robótica em 3D conhecido como Gazebo[2], no qual foi inicializado um mundo virtual e instanciados o robô e sua aplicação de controle. Nesta seção, serão apresentados detalhes desta avaliação e resultados preliminares.

4.1. Construção dos Conjuntos de Dados

A técnica avaliada utiliza um algoritmo de SVM. Esse algoritmo é normalmente utilizado em problemas de classificação, sendo que, neste trabalho, foi empregado como um classificador binário para identificar se determinada instância de um conjunto de dados representa uma anomalia ou não. Os conjuntos de dados foram gerados a partir da captura do tráfego de rede, por meio de aplicações amplamente conhecidas: Tcpdump e Tcpstat. As estatísticas reportadas pelo Tcpstat se configuram nas *features* necessárias ao algoritmo SVM, sendo que, para este trabalho, foram selecionadas o conjunto de sete, conforme a Tabela 1. Os conjuntos de dados foram rotulados com duas classes, uma representando a presença de anomalia, identificada com o número "1", e outra representando

[1]Garage, W. *et al.* (2010). Turtlebot3 - what is a turtlebot? (https://www.turtlebot.com)
[2]Howard, A. *et al.* (2002). About gazebo. (https://gazebosim.org/about)

a ausência de anomalia, identificada com o número “0”. A fim de gerar o modelo SVM e avaliá-lo foi empregada a biblioteca LIBSVM [Chang and Lin 2011] e aplicado o método descrito em [Hsu and Chang 2003]. Conforme este método, foi utilizado o *kernel* RBF juntamente com validação cruzada e *grid-search* para encontrar os melhores parâmetros C e γ, que são 32768.0 e 8,0, respectivamente.

Tabela 1. Lista de *features* selecionadas.

Índice	Feature
1	Número de bits por segundo
2	Número de pacotes por segundo
3	Número de pacotes TCP por segundo
4	Tamanho médio de pacote em bytes
5	Desvio padrão do tamanho de cada pacote em bytes
6	Utilização da rede durante a última janela observada
7	Tamanho máximo de pacote em bytes

4.2. Resultados Preliminares

Foi realizado um experimento com conjuntos de dados de 24 horas e janela de 30 segundos, tanto para treino quanto para teste. A duração dos ataques variou entre 1 e 2 minutos e o percentual de anomalia injetada foi de 20,17% para treino e 10,83% para teste. Na Figura 2 é possível visualizar um exemplo do comportamento do tráfego de rede referente a estes conjuntos de dados, sendo que em vermelho é representado o momento em que ocorrem as anomalias. Aplicando o modelo proposto obteve-se acurácia de 92,54%. Além da acurácia foram avaliadas a matriz de confusão, curva ROC (*Receiver Operating Characteristic Curve*) e AUC (*Area Under The Curve*), métricas amplamente utilizadas na avaliação de modelos de aprendizado de máquina.

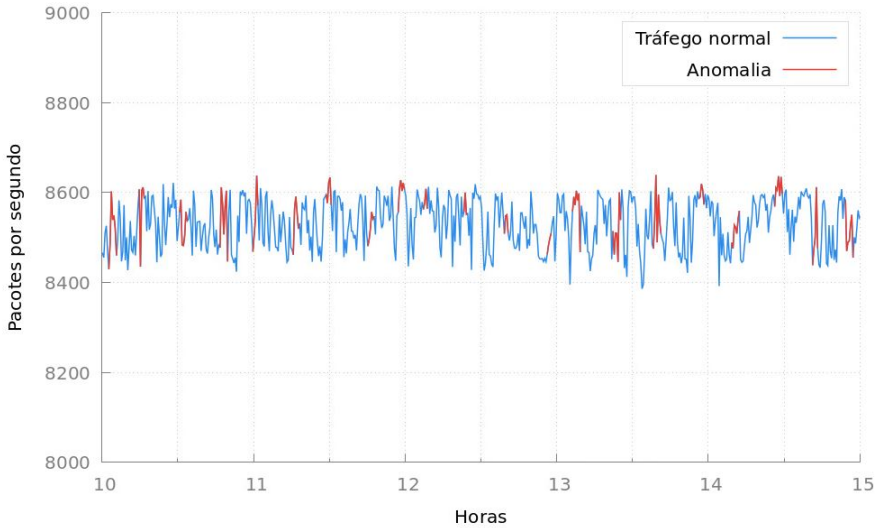

Figura 2. Média de pacotes por segundo.

A matriz de confusão é uma tabela que fornece alguns dados sobre o modelo, a partir dos quais é possível obter diversas métricas úteis à avaliação, dentre as quais

estão a Taxa de Verdadeiros Positivos (Sensibilidade) e a Taxa de Verdadeiros Negativos (Especificidade), dadas, respectivamente, pelas fórmulas 1 e 2:

$$TVP = \frac{VP}{VP + FN} \tag{1}$$

$$TVN = \frac{VN}{VN + FP} \tag{2}$$

A Tabela 2 apresenta a matriz de confusão. A partir dela obtém-se os valores 0,65 e 0,96 para TVP e TVN, respectivamente, o que indica 96% de acerto na detecção da classe 0 e 65% de acerto na detecção da classe 1.

Tabela 2. Matriz de confusão.

		Classe prevista	
		0	1
Classe esperada	0	VN: 2463	FP: 105
	1	FN: 110	VP: 202

A curva ROC demonstra o desempenho de um modelo de aprendizado de máquina, que seja um classificador binário, por meio da relação da Taxa de Verdadeiros Positivos e da Taxa de Falsos Positivos, variando um *threshold*. Já a AUC é uma métrica que indica a área sob a curva ROC, sendo utilizada para resumir a curva ROC em um único valor, simplificando sua interpretação. Analisando a Figura 3 percebe-se que quanto maior a Taxa de Verdadeiros Positivos e menor a Taxa de Falsos Positivos maior é a AUC, ou seja, melhor é a capacidade do modelo de separar classes. Nesse caso, a AUC é de 0,8786, indicando um bom desempenho do modelo.

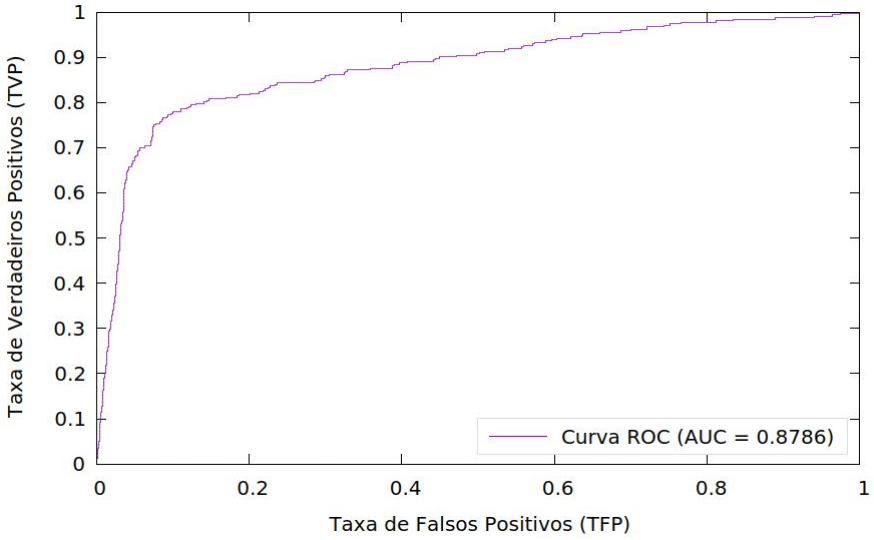

Figura 3. Curva ROC e AUC.

5. Considerações Finais e Trabalhos Futuros

A partir dos resultados parciais, nota-se que o SVM apresentou bom desempenho, atingindo acurácia de 92% e sensibilidade de 96%. Apesar disso, há espaço para

melhorias, principalmente em termos de especificidade, onde obteve-se 65%, indicando desempenho inferior na detecção de anomalia em comparação à detecção de tráfego normal. Também merece atenção a avaliação de outras técnicas empregadas em detecção de intrusão, tais como *One-Class* SVM e redes neurais, a fim de comparação de resultados.

A avaliação explora um cenário de aplicação real e sensível, no entanto, entende-se que o modelo proposto pode ser aplicado em outros cenários. Nesse sentido, possibilidades de trabalhos futuros incluem avaliar o modelo em aplicações mais complexas, tais como condução autônoma e execução de tarefas domésticas. Também é interessante avaliar o modelo quanto à injeção de outros tipos de dados maliciosos, como laser, imagem e coordenadas GPS; além de avaliar e classificar anomalias de diferentes características e causas raiz (má configuração, falha de hardware, ataques de DoS, etc.).

Referências

Amrouche, F., Lagraa, S., Frank, R., and State, R. (2020). Intrusion detection on robot cameras using spatio-temporal autoencoders: A self-driving car application. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE.

Breiling, B., Dieber, B., and Schartner, P. (2017). Secure communication for the robot operating system. In *2017 annual IEEE international systems conference (SysCon)*, pages 1–6. IEEE.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Dieber, B., Breiling, B., Taurer, S., Rass, S., and Schartner, P. (2017). Security for the robot operating system. *Robotics and Autonomous Systems*, 98:192–203.

Hsu, C.-W. and Chang, C.-C. (2003). A practical guide to support vector classification.

Mayoral-Vilches, V., Pinzger, M., Rass, S., Dieber, B., and Gil-Uriarte, E. (2020). Can ros be used securely in industry? red teaming ros-industrial. *arXiv preprint arXiv:2009.08211*.

Mohammadi Rouzbahani, H., Karimipour, H., Rahimnejad, A., Dehghantanha, A., and Srivastava, G. (2020). *Anomaly Detection in Cyber-Physical Systems Using Machine Learning*, pages 219–235. Springer International Publishing, Cham.

Narayanan, V. and Bobba, R. B. (2018). Learning based anomaly detection for industrial arm applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 13–23.

Rivera, S. and State, R. (2021). Securing robots: An integrated approach for security challenges and monitoring for the robotic operating system (ros). In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 754–759.

Teixeira, R. R., Maurell, I. P., and Drews, P. L. (2020). Security on ros: analyzing and exploiting vulnerabilities of ros-based systems. In *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pages 1–6. IEEE.

Vasquez, G., Miani, R. S., and Zarpelao, B. B. (2017). Flow-based intrusion detection for scada networks using supervised learning. *XVII Simpósio Brasileiro em Seguranca da Informacao e de Sistemas*, pages 168–181.