TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN - ĐHQG-HCM
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NETSEC DAY
NGÀY HỘI MẠNG MÁY TÍNH VÀ AN TOÀN THÔNG TIN

ChainSniper: A Machine Learning Approach for Auditing Cross-Chain Smart Contracts

Võ Anh Kiệt, Trần Tuấn Dũng, Phan Thế Duy, Nguyễn Tấn Cầm, Phạm Văn Hậu

Introduction

Blockchain technology has advanced significantly in its ability to reliably store, exchange, and verify data. However, the independent development of blockchain systems with distinct protocols has led to interoperability challenges. Cross-chain technology emerges as a solution, enabling seamless asset and data transfer across diverse blockchain networks, addressing limitations in scalability, speed, and functionality. The implementation of smart contracts in cross-chain environments presents significant security challenges, with vulnerabilities potentially leading to substantial financial losses.

This paper introduces ChainSniper, a framework integrating machine learning to automatically assess vulnerabilities in cross-chain smart contracts. The research presents "CrossChainSentinel", a comprehensive dataset of 300 manually labeled code snippets, used to train machine learning models in distinguishing between secure and vulnerable smart contracts. ChainSniper offers an automated, scalable solution for identifying potential vulnerabilities in cross-chain smart contracts, promising to significantly enhance security auditing processes for decentralized applications spanning multiple blockchain networks.

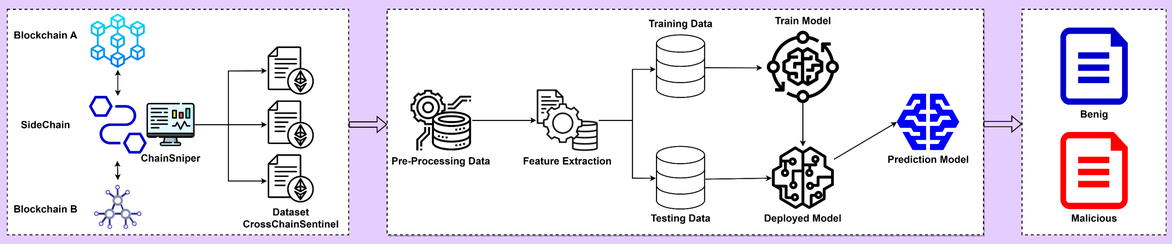

Figure 1. The conceptual architecture of the ChainSniper

Experiments and Results

The cross-chain communication environment consists of 3 virtual machines to simulate 2 different blockchain networks and a Sidechain virtual machine with 4 CPU cores, 8 GB RAM, and 60GB Hard Drive for each machine.
The machine learning training environment uses a Google Colab machine with 4 CPU cores, T4 GPU, 12.7GB RAM, and 166GB Hard Drive.

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.8519	0.8209	0.8730	0.8462
Random Forest	0.7312	0.6324	1.0000	0.7748
XGBoost	0.6211	0.5485	0.9924	0.7065
SVM	0.8458	0.7551	0.9911	0.8571
Logistic Regression	0.9000	0.9071	0.8864	0.8967
CNN	0.9000	0.8235	1.0000	0.9032
LSTM	0.5500	0.5500	1.0000	0.7100
FNN	0.8125	0.8636	0.7037	0.7755
RoBERTa	**0.9667**	**1.0000**	**0.8750**	**0.9333**

Figure 4. A Comparative Analysis: Performance of Machine Learning Models in Detecting Vulnerabilities within CrossChain Smart Contracts

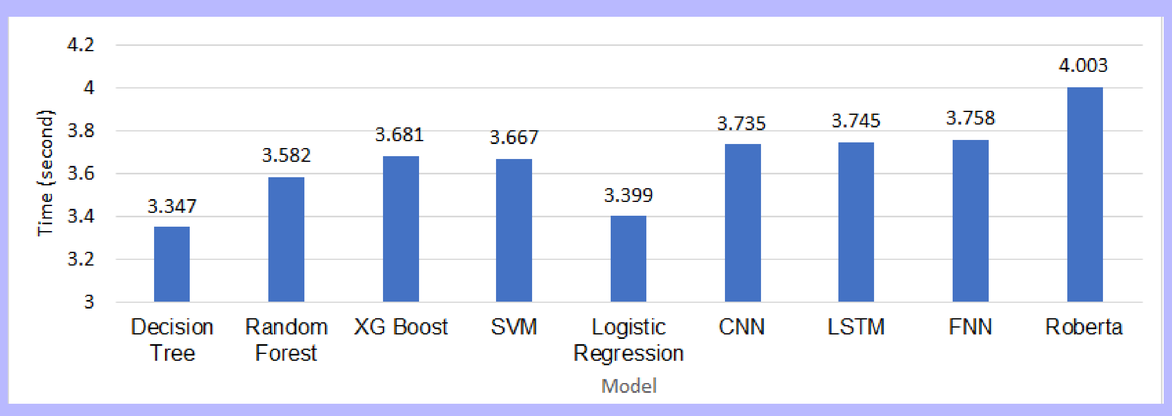

Figure 5. Time performance of the ChainSniper in detecting cross-chain smart contract vulnerabilities

Proposed Methodology

The interchain communication between two blockchains through a sidechain
The ChainSniper system enables interoperability between heterogeneous blockchains through a sidechain bridge. The sidechain processes and transfers data between interconnected networks via a two-way peg mechanism that locks assets on one chain and unlocks equivalent representations on the other using multi-signature contracts. When cross-chain transactions occur, the sidechain nodes log transaction data and aggregated into a dataset that provides insights into contract behaviors and execution patterns.

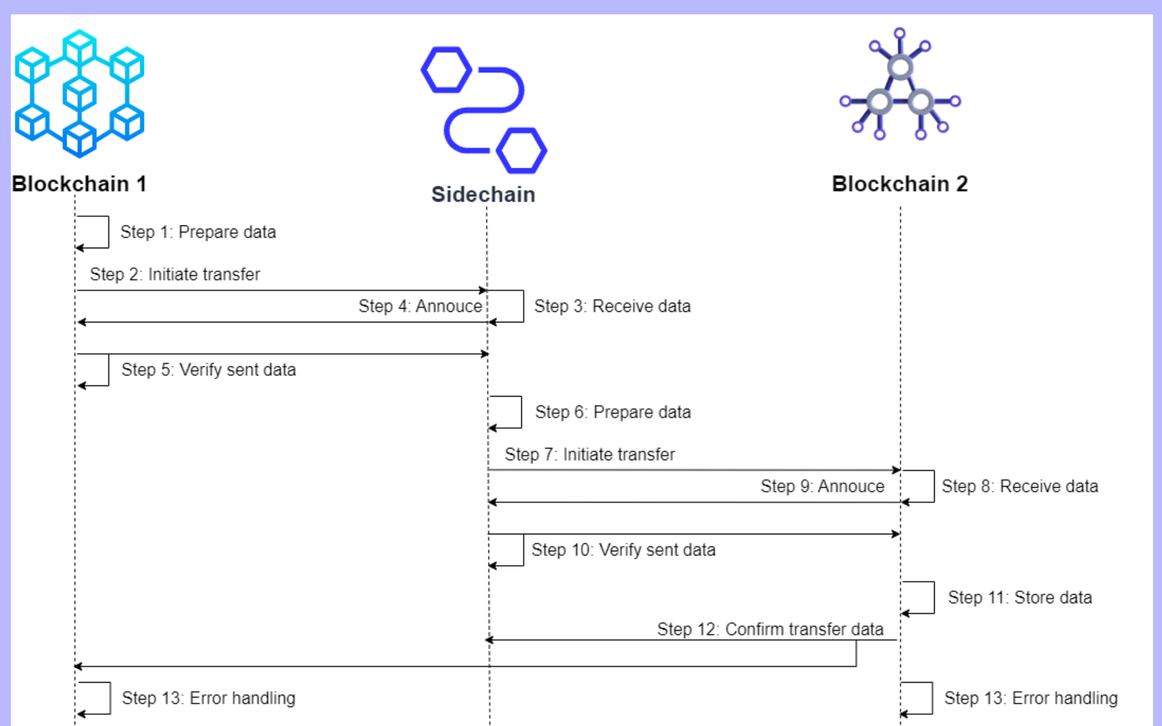

Figure 2. The sequence diagram of the interchain transfer data

The CrossChainSentinel Dataset
To analyze vulnerabilities in sidechain bridge contracts, a new dataset called CrossChainSentinel was created. This dataset contains 300 smart contract files, with 158 benign and 142 malicious samples. The malicious contracts include 42 with reentrancy vulnerabilities, 48 with integer overflow/underflow bugs, and 52 with unprotected ether withdrawal issues. The data was modeled after 15 different real-world providers such as Commos, Avalanche, and Chainlink. The dataset is labeled with two types: binary labels (benign or malicious) and multi-class labels (specific vulnerability classifications). The data is then preprocessed and transformed into feature vectors to be input into machine learning models. This diverse dataset enables robust auditing, testing, and detection of dangerous flaws in cross-chain bridging mechanisms before main-net deployment.

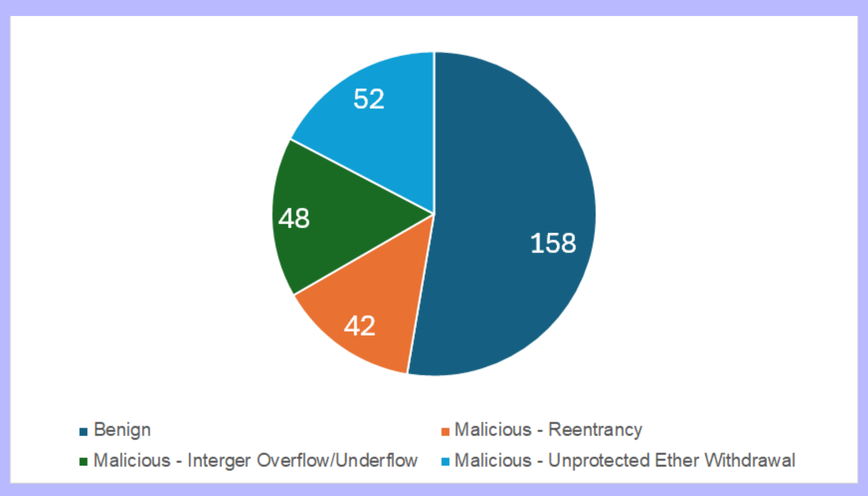

Figure 3. Distribution of benign and malicious samples corresponding to vulnerabilities

Application of Machine Learning
ChainSniper integrates a diverse range of machine learning and deep learning models to detect vulnerabilities in cross-chain smart contracts. Classical machine learning models such as Decision Trees, Random Forests, SVMs, XGBoost, and Logistic Regression. On the deep learning side, models like CNNs, LSTMs, and RoBERTa are applied to learn directly from source code, capturing complex syntactic and semantic patterns. These models can automatically learn latent relationships in code structure, thereby identifying signs of security flaws. This application creates a powerful automated scanning tool capable of detecting various types of vulnerabilities in cross-chain environments.