ProtectNIC: SmartNIC-based Ransomware Detection

Anson Xu*, Arnav Choudhury†, Eason Liu‡, Sean Choi§
Santa Clara University
*axu2@scu.edu, †achoudhury2@scu.edu, ‡eliu@scu.edu,§sean.choi@scu.edu

2024 Silicon Valley Cybersecurity Conference (SVCC) | 979-8-3503-8314-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/SVCC61185.2024.10637308

***Abstract*—Ransomware, a form of malware that restricts access to data until a ransom is paid, accounts for 20% of all cyber crimes. Although companies and organizations often require their personnel to take training for awareness of such bad actors, social engineering is constantly evolving and ransomware slips through the cracks every year. In this work, we propose a work-in-progress system called ProtectNIC that is design to detect ransomware using a Smart Network Interface Card (SmartNIC) that runs machine learning algorithms to detect ransomware before it ever enters the system. ProtectNIC enables increased security via performing detection before any interaction with the host begins and also provides lower latency in ransomware detection, while saving host CPU and memory resources. The preliminary results show promising results in ransomware detection with over 0.93 F1 score and 99% accuracy on the test set.**

***Index Terms*—SmartNIC, Ransomware Detection, In-network Machine Learning**

I. Introduction

Ransomware is a form of malware software that threatens to expose an individual's personal data or permanently restricts access to it unless a ransom is provided. While basic ransomware may merely immobilize the system without harming any files, more sophisticated malware employs a method known as cryptoviral extortion. This method encrypts the victim's files, rendering them unattainable, and requires a ransom in exchange for decryption.

There are 2 types of ransomware attacks that are very popular. The first type is the crypto-ransomware that encrypts files in a system and have the users pay some sort of ransom to unlock their files. The second type of ransomware is the locker ransomware that may lock the user out of the system. The system may have the mouse and keyboard enabled so that the user can pay to unlock the system. This malware does not intend to delete files, but lock the system until payment is made. Ransomware is most often spread as a Trojan, a virus that is disguised as another program. Phishing emails, scareware, and other forms of social engineering are common ways to install ransomware on a victim's computer. As of 2022, Ransomware accounts for 20% of all cyber crimes. [1]

Existing solutions to ransomware like training personnel are always susceptible to human error and are retrospective. Frequent backups can be used to prevent data loss, but it may be expensive to maintain these backups. With recent developments in machine learning, researchers are actively working on training models to detect ransomware. However, many of these solutions require that models run inside host machines that can be compromised. In addition, running such models can utilize precious resources of the host. To avoid such issue, this work proposes a work-in-progress system called ProtectNIC, that performs machine learning based ransomware detection in a programmable network device called SmartNIC. ProtectNIC is designed to be more secure, while reducing overheads and latency related to ransomware detection. The rest of this paper disucsses the background of this work, ProtectNIC overview and preliminary evaluation results.

II. Background

We now briefly discuss the latest advancements and pertinent information on SmartNICs, In-network Machine Learning and Ransomware detection, the three topics that are crucial to ProtectNIC.

A. SmartNICs

Smart Network Interface Cards (SmartNICs) are programmable network accelerators, often used to offload network-related tasks off of a host server. SmartNICs are increasingly being used in data centers to reduce the "overhead", such as CPU and memory usage, involved in tasks such as network virtualization, security and storage [2].

There are three main types of SmartNICs: (1) ASIC based SmartNICs, which are built with custom ASIC designs, (2) Multicore System-on-Chip (SoC) based SmartNICs, which provide much better programmability than ASICs at the cost of performance. (3) Field Programmable Gate Arrays (FPGAs) are integrated circuits with generic logic blocks that can be reprogrammed after manufacturing. ProtectNIC focuses mostly on running ransomware detection on SoC based SmartNICs, as it gives us the most programmability to utilize existing machine learning frameworks.

B. In-Network Machine Learning

In order to reduce overhead, such as hardware requirement and latency, involved in machine learning related tasks, many researchers are offloading machine learning tasks, both training and inference, on to programmable network devices, which includes network devices such as switches, routers and network interface cards. Offloading machine learning tasks can be as simple as offloading one of many mathematical operations, and can be as complicated as training and serving an entire model.

To illustrate some prior works, Planter [3] and MAP4 [4] are frameworks that allow developers to map machine learning models onto programmable network devices. P4Guard [5] is a firewall built using classification on a programmable switch. SwitchTree is an in-network analysis of network traffic via a

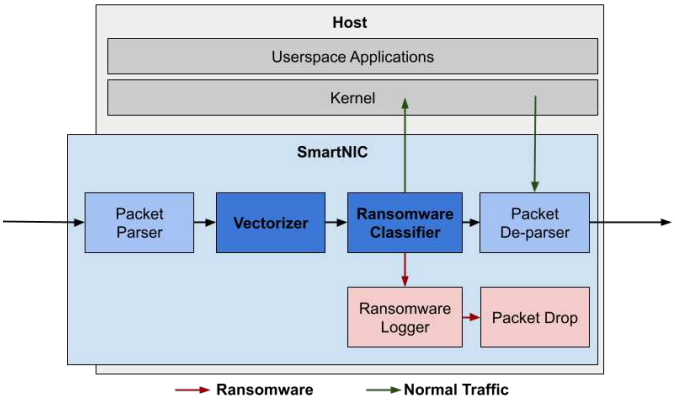

Figure 1: ProtectNIC system overview

machine learning model, called Random Forest classification, hosted in a programmable switch [6]. Furthermore, there are works that demonstrate that high accuracy packet classification can be done on SmartNICs [7]. ProtectNIC is similar in the way that it offloads a machine learning model that performs ransomware detection on the SmartNIC.

C. Ransomware Detection

Ransomware detection is a subset of malware detection, where a series of algorithm is used to detect programs (or packets) that contain ransomware. There are numerous literature on ransomware detection and methods to detect ransomware can be as simple as rule-based algorithm and can be complicated as a complex machine learning model [8]. In addition, some of these works provide their datasets and pre-trained machine learning models to reuse [9].

One of the downside when using machine learning models to detect malware is the consideration of zero-day variants. Because new forms of malware may use exploits not seen before in the training dataset, they can be difficult to detect. Existing research has also been done to consider these risks [10]. ProtectNIC utilizes some of the datasets and techniques illustrated in these works to improve our detection rate.

III. ProtectNIC Overview

In this section, we discuss a high-level overview of the components that make up ProtectNIC.

A. Proposed Design

Our proposed ransomware solution is to build a SmartNIC based ransomware detector that runs a machine learning model on a SoC based SmartNIC, especially Nvidia BlueField DPU [11]. Building on top of previous research on existing and zero-day ransomware detection using machine learning, we attempt to develop software that will allow a SmartNIC to detect anomalous activity in a byte stream. In a server, this would offload work from the CPU so it can use expensive computational power on tasks other than security.

Fig. 1 illustrates the entire process. The design involves loading four sets of programs on to a SmartNIC: parser, vectorizer, classifier and deparser. One thing to note is that the parser and deparser is not added by ProtectNIC, rather it is part of all network switching operation. First, the SmartNIC uses the parser to read every packet and then the vectorizer creates an embedding, i.e., a vector of floating point represenation of the packet, using the ProtectNIC vectorizer. Then, the embedding is passed into the ProtectNIC ransomware detection model to perform the ransomware detection. If the packet deemed malicious by the model, it is logged by the SmartNIC and is dropped, while other packets are be forwarded to the host system, as if it is a regular traffic. The two benefits of this approach is that: (1) Ransomware detection is isolated from the host machine, so attack surface of ransomware is minimized, and (2) ransomware detection can be run at a very low latency.

IV. Preliminary Evaluation

Given the design in Section III, we provide a set of preliminary results of this work.

A. Dataset

ProtectNIC ransomware classifier is trained from multiple public datasets. First, the packet traces of ransomware are obtained from the Ransomware PCAP repository, which provides samples of network activity that was recorded during the encryption of files by different families of ransomware [12]. There are a total of 39 families of ransomware in the repository, each with multiple PCAP files of sizes from a couple hundred megabytes to several gigabytes. The repository has recent data, and has files from 2015 up until 2021. Because they capture network activity during ransomware activity, these datasets will help us train a model that can identify malicious network activity packet by packet. Second, the packet traces of goodware, i.e., non-malicious packet traces, are obtained from the UNSW-NB15 dataset [13].

Using a subset of the traces available due to resource constraints, the traces in the dataset were grouped into the same flow, giving us a total of 31,030 goodware flows and 2,355 ransomware flows. 90% of this set was used for training and the other 10% was used for evaluation.

B. Vectorizer

In order to create the embedding to train a machine learning model, all the flow is broken down into 4 byte chunks to create a dictionary of unique 4 byte chunks. Given N unique chunks that are available across the entire dataset, each embedding, i.e., vector, of size N is created by taking the normalized occurrence of each packets. For example, if the packet only has 4 bytes, one of the entries in the N vector is populated. Given this process, every flow is embedded into a vector to be used for machine learning model training.

C. Ransomware Classifier

Given the embeddings, two types of classifiers: XGBoost [14] and random forest, are trained for evaluation. Table I and Table II provide the details of the hyperparameters used for training. Given the trained model, Fig. 2 provides the model

Hyperparameter	XGBoost Parameter	Value
Learning Rate	`eta`	0.3
l_1 Regularization	`alpha`	0
l_2 Regularization	`lambda`	1
Min. Split Loss	`gamma`	0
Row Subsample Ratio	`subsample`	1
Column Subsample Ratio	`col_subsample`	1
Max. Tree Depth	`max_depth`	6
Boosting Rounds	`num_round`	256

Table I: Hyperparemeter used for XGBoost training

Hyperparameter	Random Forest Parameter	Value
Number of Trees	`n_estimators`	100
Split Quality	`criterion`	gini
Min. observation	`max_samples_split`	2
Min. leaf size	`min_samples_leaf`	1
Num. Max. features	`max_features`	sqrt

Table II: Hyperparemeter used for Random Forest training

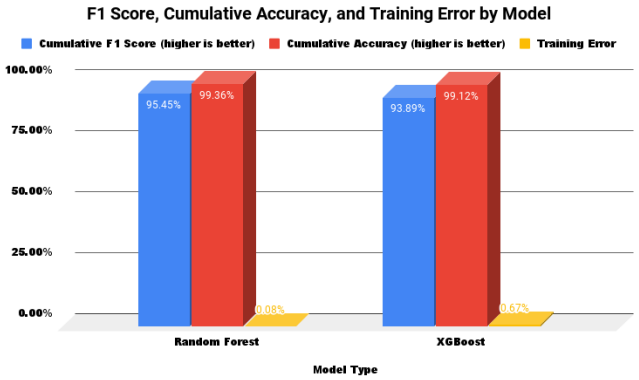

Figure 2: Model evaluation results for Random forest and XGBoost algorithm for ransomware detection.

performance in both accuracy and F1 score on the test set. The results show that while both random forest and xgboost performs at a very high F1 and accuracy (over 0.93 and 0.99 respectively) random forest algorithm performs slightly better. An assumption for this result is that the data imbalance is not resolved correctly for xgboost model training, which is one of the future optimizations to be made.

V. Future Works

Given the promising ransomware detection result for ProtectNIC vectorizer and model training algorithm, future work involves three of the following steps: (1) further optimization of the classifier and vectorizer via more sophisticated methods, such as neural embeddings, (2) latency and model size optimization when offloaded to a SmartNIC, (3) real-world testing against zero-day variants. Once these objectives are achieved, we also plan to test the entire end-to-end system deployed to a real-world testbed.

VI. Conclusion

In this paper, we present ProtectNIC, a work-in-progress idea to enable secure and low-latency ransomware detection using unique machine learning models offloaded to a SmartNIC. The initial preliminary results show a promising ransomware detection performance and the initial design also shows promising potential in reducing resource usage and latency overhead in ransomware detection.

References

[1] C. Griffiths, "The latest ransomware statistics (updated april 2024): Aag it support," Jan 2024. [Online]. Available: https://aag-it.com/the-latest-ransomware-statistics/

[2] D. Firestone, A. Putnam, S. Mundkur, D. Chiou, A. Dabagh, M. Andrewartha, H. Angepat, V. Bhanu, A. Caulfield, E. Chung, H. K. Chandrappa, S. Chaturmohta, M. Humphrey, J. Lavier, N. Lam, F. Liu, K. Ovtcharov, J. Padhye, G. Popuri, S. Raindel, T. Sapre, M. Shaw, G. Silva, M. Sivakumar, N. Srivastava, A. Verma, Q. Zuhair, D. Bansal, D. Burger, K. Vaid, D. A. Maltz, and A. Greenberg, "Azure accelerated networking: SmartNICs in the public cloud," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. Renton, WA: USENIX Association, Apr. 2018, pp. 51–66. [Online]. Available: https://www.usenix.org/conference/nsdi18/presentation/firestone

[3] C. Zheng, M. Zang, X. Hong, L. Perreault, R. Bensoussane, S. Vargaftik, Y. Ben-Itzhak, and N. Zilberman, "Planter: Rapid Prototyping of In-Network Machine Learning Inference," *ACM SIGCOMM Computer Communication Review*, 2024.

[4] B. M. Xavier, R. Silva Guimarães, G. Comarela, and M. Martinello, "Map4: A pragmatic framework for in-network machine learning traffic classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4176–4188, 2022.

[5] R. Datta, S. Choi, A. Chowdhary, and Y. Park, "P4guard: Designing p4 based firewall," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 1–6.

[6] J. Lee and K. Singh, "Switchtree: in-network computing and traffic analyses with random forests," *Neural Computing and Applications*, 2020, publisher Copyright: © 2020, Springer-Verlag London Ltd., part of Springer Nature.

[7] B. M. Xavier, R. S. Guimarães, G. Comarela, and M. Martinello, "Programmable switches for in-networking classification," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[8] A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020.

[9] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Expert Systems with Applications*, vol. 209, p. 118299, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422014312

[10] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive pareto ensemble classifier," *Scientific Reports*, vol. 12, no. 1, September 2022. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/28937/

[11] "NVDIA Bluefield Networking Platform," https://www.nvidia.com/en-us/networking/products/data-processing-unit/, 2024.

[12] E. Berrueta, D. Morató, E. Magaña, and M. Izal, "Open repository for the evaluation of ransomware detection tools," 2020. [Online]. Available: https://dx.doi.org/10.21227/qnyn-q136

[13] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785