2022 IEEE 40th International Conference on Computer Design (ICCD) | 978-1-6654-6186-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCD56317.2022.00046

Iron-Dome: Securing IoT Networked Systems at Runtime by Network and Device Characteristics to Confine Malware Epidemics

Sanket Shukla, Abhijitt Dhavlle, Sai Manoj P D
Dept. of Electrical & Computer Engineering
George Mason University, Fairfax
VA, USA
{sshukla4, adhavlle, spudukot}@gmu.edu

Houman Homayoun, Setareh Rafatirad
Dept. of Computer Science
University of California, Davis
CA, USA
{hhomayoun, srafatirad}@ucdavis.edu

***Abstract*—The rapid growth of IoT networks presents an enlarged "attack space" for the adversary and poses significant security risks on a large scale. A single device in a network that is compromised under the influence of a malware attack, has the potential to spread malware across the network. This leads to a plethora of attacks, including DoS and ceasing the network functionality. Given the scale of IoT networks and the connectivity among the devices, mere detection and quarantining of malware in IoT networks does not limit the propagation of malware in IoT networks. This work proposes an integrated defense, termed as "IRON-DOME", comprising of (1) an on-device application analyzer: Image-based Malware detector that utilizes grayscale images of executables, (2) Device dynamic behavior analysis: Reliable extraction and dynamic analysis of malware Hardware Performance Counter (HPC) values; and (3) Device communication trait analyzer: Uses network packet data analysis to confine and propagate malware in the IoT network. The proposed solution yields: (1) a runtime malware detection accuracy of 93% within 19 ns, (2) is resource and power efficient; it consumes 30% fewer resources and 40% less power than state-of-the art defense techniques.**

I. Introduction

The proliferation of Internet-of-Things (IoT) devices into the consumer and industrial markets with add-on connectivity to the Internet or WiFi towards smart and intelligent features leads to growth in the size of networks through which they communicate [1]. As per the reports [2], nearly 70 billion IoT devices will be connected to the IoT network. IoT devices are a priority choice for industry due to several features such as heterogeneous protocol communication, low-power, miniature footprint, and mobility. Despite the number of benefits offered by IoT devices, there are an equal number of potential vulnerabilities and some serious security risks [3], [4].

Software-based malware detection utilizes signature-based detection [5], [6] that matches the behavior signature of the application to its database. This approach fails to recognize zero-day attacks, and signatures that do not match its database. To overcome shortcomings such as latency and computational complexity of traditional malware detection techniques, including signature and semantics-based software-driven techniques [7], [8], Hardware-Assisted Malware detection (HMD) approaches are proposed [9]. HMD refers to utilizing the low-level microarchitectural hardware events for detecting and classifying malware from benign applications. The HMD delivers reduced malware detection latency by orders of magnitude with a smaller hardware cost [9]. In addition, researchers explored computer vision-based application analysis for malware detection. Natraj et al. [10] classifies malware images by using k-nearest neighbors. This approach is not capable of handling code obfuscation techniques like code relocation and mutation, and may not be feasible in resource constrained environments because to extract the image texture as features for classification, the system needs preprocessing. An artificial neural network (ANN) for malware classification is used in [11] which is computationally costly as there are multiple fully connected layers in (ANN). In [12], [13] authors convert the binary executable of malware samples to a gray-scale image and utilize a single-channel lightweight convolutional neural network (CNN) to efficiently detect IoT malware. Another potential security risk in IoT networks is the feasibility of malware spreading across the network.

To address the above issues, we propose an integrated defense "IRON-DOME" that addresses both the challenges: a) developing and deploying lightweight malware detection on IoT devices without incurring large overheads, and b) confine the propagation of malware in the IoT network even with imperfect infection information while preserving network integrity and overall performance. IRON-DOME secures the IoT devices and network against malware by deploying a three-fold strategy: (1) On-device application analyzer - utilizes grayscale images of executable files for malware detection, (2) Device dynamic behavior analysis - Extracts the dynamic HPC features during run-time for malware and benign applications; and (3) Device communication trait analyzer - A thorough network packet analyzer for malware confinement and propagation in IoT networks. Extracting and utilizing multiple features to confine malware and practicing multi-machine learning models makes the proposed technique processor independent. Moreover, since IoT devices have limited resources, we deploy lightweight multi-models for effective malware detection. The experimental results illustrate that the proposed framework is robust, processor agnostic, and safeguards the IoT networked devices from malware attacks by preventing the network functionality from getting compromised.

2576-6996/22/$31.00 ©2022 IEEE
DOI 10.1109/ICCD56317.2022.00046

II. Proposed Technique

The detailed overview of IRON-DOME is structured as follows: (1) a brief overview of the technique; (2) details of the vision-based malware detection technique; (3) HPC-based malware detection; and (4) device communication trait analyzer.

A. Overview of the Proposed Framework - "IRON-DOME"

The overview architecture of the proposed framework "IRON-DOME" is presented in Figure 1. As soon as the IRON-DOME encounters an incoming application, the application undergoes the following malware detection techniques in parallel: (1) malware detection based on images; (2) malware detection based on dynamic HPC features; and (3) malware detection based on network packet data. Generative Adversarial Networks (GANs) are used for data augmentation to generate additional synthetic data.

B. Synthetic Data Generation using GANs

In this section, we describe the generation of synthetic data using conditional GANs (CGANs) and Wasserstein Conditional GANs (WCGANs). The GAN block actively generates synthetic data if the number of samples is too small for efficient training. It should be noted that the GANs are used for generating synthetic data only if the number of training samples is insufficient for a specific class. In the situation where the dataset is balanced, the synthetic data generation can be skipped by deactivating the GAN block, ultimately speeding up the training process. Thus, GANs augment the training data to upscale the IRON-DOME's malware classification performance.

C. Vision-based On-device Application Analyzer

The vision-based on-device application analyzer is a real-time malware detection system based on the grayscale image of the application binaries. We use lightweight convolutional neural nets (CNNs) and lightweight vision transformers (ViT) for real-time malware detection. Before deploying the models, they need to be trained with sufficient samples of malware and benign image data. The grayscale images of the application binaries are converted using image processing and the synthetic data generated using GANs is used for training CNNs and vision transformers (ViT).

In the proposed framework, the CNN and ViT models perform inferencing in parallel. The reason for deploying two state-of-the-art vision models in parallel is that ViT provides an add-on advantage as compared to CNN in terms of identifying malware patterns that are hidden or stealthy. CNN uses pixel arrays, whereas ViT splits the images into visual tokens. We adopt and use the MobileNetV2 architecture [14] in our work. The visual transformer divides an image into fixed-size patches, correctly embeds each of them, and includes positional embedding as an input to the transformer encoder. Therefore, in a grayscale image, if the malware texture (or patch) is hidden inside the benign file, there can be a possibility that CNN will fail to classify the file as benign, but ViT will recognize the malicious pattern and correctly classify the file as a class of malware.

D. HPCs-based Dynamic Behavior Analysis

1) Feature Extraction, Multiple Attributes and ML based classification: To address reliability concerns that have not been addressed in previous works, we propose fine-tuning the cutting-edge model-specific registers (MSRs) available in modern computing system architectures, which are the source of HPC information. First, to solve the non-determinism challenge in HPCs, we redesigned HPC capturing protocols with proper context switching and handling performance monitoring interrupt (PMI) units in the system while collecting HPCs. Furthermore, to ensure proper context switching and reading of HPCs, PMIs can aid. It has been seen that configuring PMI per process often leads to better capturing of the HPCs [5]. Through this two-pronged utilization of context-switching and PMI, we collect reliable HPCs. To address the challenges such as over counting [5], we perform calibration through testing.

Another challenge that hinders portability of detection techniques to other vendors' architectures is the heterogeneity in the terminology used (naming convention) and the lack of HPCs on some architectures. This issue is addressed in this project in a two-step manner. Firstly, the event names are compared and clustered depending on the component (such as LLC, branch predictor, and data cache) and the similarity in the terms and functionality in the next stage for appropriate matching. If the names do not match and functionality cannot be derived directly, then through computations (performing arithmetic on a few HPC values), we derive the values, i.e., analogous to how temperature is derived from power through "soft"-sensors.

2) Obtaining Multiple Attributes:: The proposed hardware-assisted malware detection (HMD) comprises of feature selection and runtime malware detection stages. Feature selection is performed offline, and malware detection is performed online. Feature selection is performed to alleviate the challenge of a limited number of available on-chip HPCs. To perform this, during offline profiling, the irrelevant data (unnecessary HPC features) is identified and removed using a feature reduction algorithm (correlation analysis and feature scoring), and as such, only a subset of HPCs that represent the most critical features required for malware detection is selected during runtime. This method helps reduce the set of prominent features, including HPCs representing pipeline front-end, pipeline back-end, cache subsystem, and main memory behaviors, which are influential in the performance of standard applications. The reduced set of HPCs are monitored during application analysis at runtime to classify the application as benign or malware (and its class) utilizing the machine learning classifiers. Depending on the number of on-chip HPCs, the selected features are monitored online. The ML classifier is deployed on IoT devices to classify and detect malware. The prediction from multi-modal is considered and, based on the max-voting, the final label is predicted.

E. Device Communication Trait Analyzer

To enable efficient threat detection and be processor-agnostic, in such scenarios, we envision utilizing networking

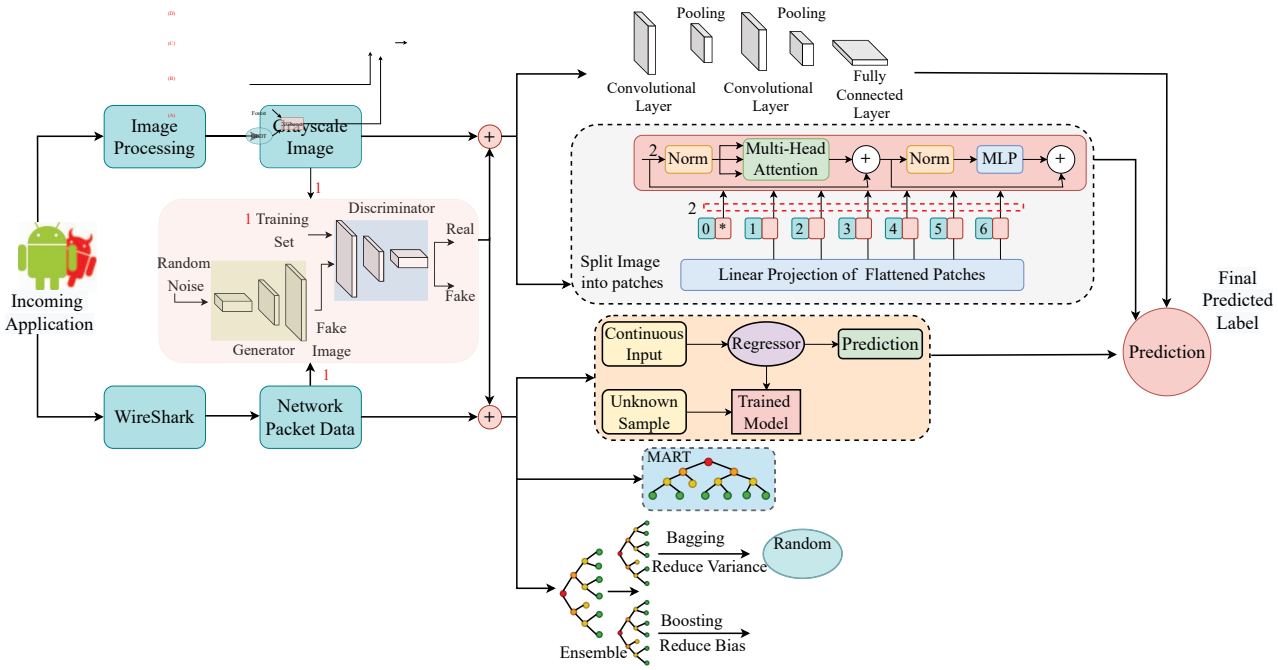

Fig. 1. Overview of Proposed Technique

attributes for threat detection. The network activity of the device in the form of packets (data) sent or received, the rate of connections, data transmission rate, syntax and semantics of data packets, devices, or HTTP to which the device sends data, and authorization to send is obtained. One needs to note that the above-mentioned features are merely a representation of each class, while more features and classes will be considered and analyzed. The Device communication trait analyzer uses machine learning techniques to classify different types of captured packets without decrypting their contents to detect any malicious intrusions in the IoT network. For this, the benign and malware applications are executed and the packets are captured by Wireshark for malicious traffic detection and analysis. The most prominent features from the captured packets are used for classification. The ML models used to classify the Wireshark packet data are regression models, Multiple Additive Regression Trees (MART), and XGBoost.

III. Experimental Results

A. Experimental Setup

We experimented with our proposed technique on a Raspberry Pi 4, model B, featuring a Quad-Core Cortex-A72 (ARM v8) 64-bit SoC with 4 GB of RAM, and the Raspbian operating system. The Raspberry Pi (RPi) also features internet connectivity, making it suitable for IoT applications. Wireshark was used to collect internet packet data. Malware binaries were downloaded from VirusShare [16], VirusTotal [17].

B. Training Data Collection and Generation

We collected HPCs inside a sandboxed environment; the sandboxing was ensured by using Docker containers. An instance of the container was created for every malware binary and destroyed after the execution was completed and HPCs collected. In order to extract the HPC information, we used *Perf* tool available under Linux. *Perf* provides rich, generalized abstractions over hardware-specific capabilities. It exploits *perf-event-open* function call in the background, which can measure multiple events simultaneously.

C. Observation Results

Table I reports the performance of the proposed framework in comparison with the state-of-the-art malware detection techniques. The classification results and metrics demonstrate that the proposed framework achieves the highest values (91% to 95%) of the evaluation metrics (accuracy, precision, recall, F1 score) for each class of malware as compared to HMD-based malware detector, HPC-based malware detection, and image-based malware detection. Moreover, performing individual classifier-based malware detection does not make the framework processor or architecture agnostic. As a result, a proposed framework has been developed to aid in performing processor and architecture agnostic malware detection by utilizing a multi-modal approach that employs multiple characteristics or features.

By incorporating multi-modals and training these models with grayscale-image-based features, HPC values, and network-based features, the malware epidemic in IoT networked devices is contained. The maximum voting criteria are used, which further reduces the probability of misclassifying the incoming application. The proposed IRON-DOME attains enhanced performance for efficient malware detection. The proposed IRON-DOME framework outperforms other classifiers in terms of testing latency. The testing latency achieved by IRON-DOME is less than 19 ns (Table II).

TABLE I
COMPARISON OF ACCURACY AND PRECISION METRIC

ML Classifier	Accuracy						Precision					
	Backdoor	Rootkit	Trojan	Virus	Worm	Bengin	Backdoor	Rootkit	Trojan	Virus	Worm	Bengin
CNN	0.90	0.89	0.86	0.88	0.90	0.89	0.92	0.90	0.88	0.88	0.92	0.91
ViT	0.86	0.85	0.84	0.82	0.87	0.85	0.87	0.88	0.86	0.85	0.87	0.84
Regressor	0.82	0.86	0.79	0.85	0.89	0.86	0.85	0.90	0.84	0.86	0.90	0.89
MART	0.87	0.78	0.77	0.82	0.83	0.86	0.88	0.80	0.82	0.85	0.88	0.90
XGBoost	0.89	0.83	0.85	0.87	0.84	0.89	0.90	0.86	0.88	0.87	0.84	0.92
HMD-Hardner [15]	0.90	0.93	0.81	0.85	0.82	0.84	0.90	0.91	0.84	0.86	0.85	0.82
HPC-based [5], [9]	0.87	0.85	0.86	0.82	0.84	0.85	0.85	0.86	0.88	0.84	0.85	0.87
Image-based [10]	0.84	0.86	0.89	0.87	0.83	0.86	0.82	0.86	0.87	0.86	0.82	0.85
IRON-DOME	**0.94**	**0.92**	**0.94**	**0.94**	**0.96**	**0.95**	**0.95**	**0.94**	**0.95**	**0.95**	**0.95**	**0.96**

TABLE II
COMPARISON OF TESTING LATENCY IN NANO SECONDS

Testing Latency in nano-seconds (ns)					
Classifier	Backdoor	Rootkit	Trojan	Virus	Worm
CNN	69.5	69.3	69.1	69.5	69.2
ViT	59.8	59.5	59.6	58.2	59.7
Regressor	68.8	68.76	68.7	68.6	67.8
MART	59.8	59.76	59.9	59.71	58.9
XGBoost	64.8	64.6	65.9	65.2	64.7
HMD-Hardner [15]	63.8	64.6	63.9	64.1	64.7
HPC-based [5], [9]	69.5	69.6	69.74	69.8	70.2
Image-based [10]	59.5	59.67	60.7	60.2	60.5
IRON-DOME	**18.4**	**18.9**	**18.3**	**18.6**	**18.4**

D. Comparison of Energy and Power Consumption

Hardware implementation of the classifiers embedded into the proposed technique is performed on the ASIC Broadcom BCM2711, a quad-core Cortex-A72 (ARM v8) 64-bit, 28 nm SoC running at 1.5 GHz. The power and energy values are reported at 100 MHz. We used the Design Compiler by Synopsys to obtain the area for the models. Power consumption is obtained using Synopsys Primetime PX. The post-layout power, and energy are summarized in Table III. Among all the techniques, HMD consumes the highest power, energy, and area on-chip (Table III). The post-layout energy numbers were almost 2 $\times$ higher than the post-synthesis results. This increase in energy is mainly because of metal routing resulting in layout parasitics. As the tool uses different routing optimizations, the power and energy values keep changing with the classifiers' composition and architecture.

TABLE III
POWER AND ENERGY CONSUMPTION

Techniques	Power (mW)	Energy (mJ)
HMD	48.61	4.12
HPC-based	45.64	3.51
Image-Based	46.63	3.46
IRON-DOME	**28.46**	**2.71**

IV. CONCLUSION

In this work we proposed and evaluated a multi-modal malware detection approach IRON-DOME. We analyze various aspects by extracting multiple features of malware and benign executables. We thoroughly evaluated and reported the performance metrics of IRON-DOME framework with the state-of-the-art malware detection techniques. The proposed IRON-DOME framework achieves a runtime malware detection accuracy of 93% with a latency of less than 19 ns. Moreover, the proposed framework is resource efficient and consumes 30% less resources and 40% less power than state-of-the-art techniques. Therefore, with the proposed multi-modal malware detection approach, the framework is not only robust but also architecture and processor independent.

REFERENCES

[1] T. Xu and et al., "Security of IoT systems: Design challenges and opportunities," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2014.

[2] S. Koley and et al., "Addressing hardware security challenges in internet of things: Recent trends and possible solutions," in *IEEE Int. Conf. on Ubiquitous Intelligence and Computing*, 2015.

[3] T. Abera and et al., "Things, trouble, trust: On building trust in iot systems," in *Design Automation Conference (DAC)*, 2016.

[4] J. Wurm and et al., "Security analysis on consumer and industrial iot devices," in *(ASP-DAC)*, 2016.

[5] S. M. P. Dinakarrao and et al., "Adversarial attack on microarchitectural events based malware detectors," in *Design Automation Conf.*, 2019.

[6] S. Shukla, G. Kolhe, P. Sai Manoj, and S. Rafatirad, "Work-in-progress: Microarchitectural events and image processing-based hybrid approach for robust malware detection," in *Int. Conf. on Compliers, Architectures and Synthesis for Embedded Systems (CASES)*, 2019.

[7] N. Patel and et al., "Analyzing hardware based malware detectors," in *Design Automation Conf.*, 2017.

[8] S. Shukla and et al., "Rafel - robust and data-aware federated learning-inspired malware detection in internet-of-things (iot) networks," in *Great Lakes Symposium on VLSI*, 2022.

[9] J. Demme and et al., "On the feasibility of online malware detection with performance counters," *SIGARCH Comput. Archit. News*, 2013.

[10] L. Nataraj and et al., "Malware images: Visualization and automatic classification," in *Int. Symp. on Visualization for Cyber Security*, 2011.

[11] S. Shukla and et al., "On-device malware detection using performance-aware and robust collaborative learning," in *Design Automation Conference (DAC)*, 2021.

[12] J. Su and et al., "Lightweight classification of iot malware based on image recognition," in *Computer Software and Applications Conf.*, 2018.

[13] S. Shukla and et al., "Rnn-based classifier to detect stealthy malware using localized features and complex symbolic sequence," in *Int. Conf. On Machine Learning And Applications (ICMLA)*, 2019.

[14] M. Sandler and et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," *Conference on Computer Vision and Pattern Recognition*, 2018.

[15] A. Dhavlle and et al., "Hmd-hardener: Adversarially robust and efficient hardware-assisted runtime malware detection," in *Design, Automation Test in Europe Conf. Exhibition (DATE)*, 2021.

[16] (2020) Virusshare team. Last accessed: 05-Dec-2020. [Online]. Available: www.virusshare.com

[17] (2020) Virustotal intelligence service. Last accessed: 05-Dec-2020. [Online]. Available: www.virustotal.com/intelligence