POSTER: Breaking Graph-based IoT Malware Detection Systems Using Adversarial Examples

Ahmed Abusnaina, Aminollah Khormali, Hisham Alasmary, Jeman Park,
Afsah Anwar, Ulku Meteriz, and Aziz Mohaisen
University of Central Florida
Orlando, USA

ABSTRACT

The main goal of this study is to investigate the robustness of graph-based Deep Learning (DL) models used for Internet of Things (IoT) malware classification against Adversarial Learning (AL). We designed two approaches to craft adversarial IoT software, including Off-the-Shelf Adversarial Attack (OSAA) methods, using six different AL attack approaches, and Graph Embedding and Augmentation (GEA). The GEA approach aims to preserve the functionality and practicality of the generated adversarial sample through a careful embedding of a benign sample to a malicious one. Our evaluations demonstrate that OSAAs are able to achieve a misclassification rate (MR) of 100%. Moreover, we observed that the GEA approach is able to misclassify all IoT malware samples as benign.

CCS CONCEPTS

• **Security and privacy → Malware and its mitigation**.

KEYWORDS

Adversarial Learning, Deep Learning, Graph Analysis, Internet of Things, Malware Detection

1 INTRODUCTION

Internet of Things (IoT) devices, including sensors, voice assistants, automation tools, are widely used, increasing the attack surface of the Internet due to their evolving and often insecure software. However, the research work on IoT software analysis has been very limited not only in the size of the analyzed samples, but also the utilized approaches. A promising direction leverages a graph-theoretic approach to analyze IoT malware. Representative static characteristics of IoT applications can be extracted from the Control Flow Graph (CFG), which can be utilized to build an automatic IoT malware detection system [2].

Machine Learning (ML) algorithms, specifically DL networks, are actively used in a wide range of applications [4]. However, it has been shown that ML/DL networks are vulnerable to AL, where an adversary can force the model to his desired output, *e.g.* misclassification. Although it is an active research area, there is very little research work done on understanding the impact of AL on DL-based IoT malware detection system and practical implications [3], particularly those that utilize CFG features for detection [1].

Goal of this study. Motivated by the aforementioned issues, our main goal is generating *adversarial IoT software samples that (1) fool the classifier and (2) function as intended.*

Approach. To tackle the above objectives, we designed two approaches to craft adversarial examples, including OSAA and GEA approaches. The OSAA approach incorporates six well-known adversarial learning methods to force the model to misclassification. Whereas, the GEA approach aims to preserve the functionality and practicality of the generated adversarial samples through a careful connection of benign graph to a malicious one.

Contributions. Our contributions are as follows: 1) We examined the robustness of CFG-based deep learning IoT malware detection system using two different approaches, including off-the-shelf adversarial learning algorithms and graph embedding and augmentation, while maintaining the practicality and functionality of the crafted AEs. 2) We found that the first approach can generate AEs with MR of 100%. However, they do not guarantee the practicality and functionality of the crafted AEs, unlike the GEA approach.

2 GENERATING ADVERSARIAL EXAMPLES

In order to generate realistic AEs that preserve the functionality and practicality of the original samples we design two approaches: Off-the-Shelf Adversarial Attacks (OSAA) and Graph Embedding and Augmentation (GEA).

OSAA. This approach incorporates well-established adversarial machine learning attack methods into IoT malware detection. These methods apply small perturbation into the feature space to generate AEs that lead to misclassification.

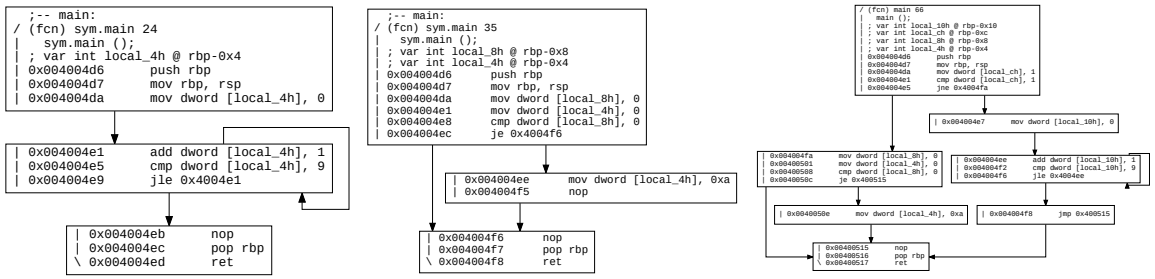
GEA. Assume an original sample x_{org} and a selected target sample x_{sel}, our main goal is to combine the two samples while preserving the functionality and practicality of x_{org} and achieving misclassification. Prior to generating the CFG for these algorithms, we compile the code using GNU Compiler Collection (GCC) command. Afterwards, Radare2 is used to extract the CFG from the binaries. 1(a) and 1(b) show the generated graphs for x_{org} and x_{sel}, respectively.

3 EVALUATION AND DISCUSSION

Dataset. We obtained the CFG dataset of the IoT malware from Alasmary *et al.* [2] to assess our proposed approach. The dataset consists of 2,281 malicious and 276 benign IoT samples. We extracted 23 different features in seven different groups, including betweenness centrality, closeness centrality, degree centrality, shortest path, density, # of edges, and # of nodes.

WiSec '19, May 15–17, 2019, Miami, FL, USA

ACM ISBN 978-1-4503-6726-4/19/05.
https://doi.org/10.1145/3317549.3326296



(a) Original sample's CFG (b) Taraget sample's CFG (c) Crafted adversarial CFG using GEA

Figure 1: A practical implementation of the GEA approach. Fig. 1(a) shows the generated CFG for the original sample and used for extracting graph-based features (graph size, centralities, etc.) for graph/program classification and malware detection. 1(b) shows the graph for the selected target sample generated as in Fig. 1(a). Finally, The generated adversarial graph using GEA approach. Note that this graph is obtained logically by embedding the graph in Fig. 1(b) into the graph in Fig. 1(a).

Table 1: Evaluation of the OSAA methods. MR: misclassification rate, Avg.FG: average number of changed features, and CT: computation time.

Attack Method	MR (%)	Avg.FG	CT (ms)
C&W	100	12.60	25.30
DeepFool	86.39	14.90	2.56
ElasticNet	100	5.42	114.18
JSMA	99.80	4.00	0.78
MIM	100	20.60	0.90
PGD	100	22.56	2.40

Table 2: GEA: Malware to benign (Mal2Ben) and benign to malware (Ben2Mal) misclassification rate. MR: misclassification rate, CT: computational time.

	Size	# Nodes	MR (%)	CT (ms)
Mal2Ben	Minimum	2	7.67	33.69
	Median	24	95.48	37.79
	Maximum	455	100	1,123.12
Ben2Mal	Minimum	1	30.65	40.65
	Median	64	57.60	69.23
	Maximum	367	88.04	473.91

Results & Discussion. This section is divided into three subsections. The following is detailed discussion of the obtained results.

- Deep Learning-based IoT Malware Detection System: We designed a CNN-based classifier, which distinguishes IoT malware samples from benign ones, trained over 23 CFG-based features categorized in seven groups, including betweenness centrality, closeness centrality, degree centrality, shortest path, density, # of edges, and # of nodes, extracted from CFGs of 2,281 malware and 276 benign samples. We achieved an accuracy rate of 97.13% with a FNR of 11.26% and FPR of 1.55%, the high value of FNR is due to the imbalanced number of malware and benign samples.
- OSAA: We implemented six generic AL attack methods to generate AE by perturbing the feature space. Overall, those approaches have shown, in general, a good performance (see Table 1). Detailed discussion regarding OSAA can be found in [1].
- GEA: This approach is designed to generate AE that fools the classifier, while preserving the functionality and practicality of the original sample. Here, we discuss the inherent overhead of the GEA approach. We investigate the impact of the size of the graph, determined by the number of the nodes in a graph, on the MR. Note that all generated samples maintain the practicality and the functionality of the original sample.

  Graph Size Impact. We selected three target graphs from each of the benign and malicious IoT software, consisting of a minimum, median and maximum graph size, to understand the impact of size on MR with GEA. The results are shown in Table 2. We found that the MR increases when the number of nodes increases. In addition, the time needed to craft the AE is proportional to the size of the selected sample. We achieved a malware to benign MR of as high as 100%, and a benign to malware MR of 88.04%.

4 CONCLUSION

In this work, we generated the CFGs of the IoT samples, we then extracted 23 representative features from the CFGs to train our DL model. The focus of this study is to investigate the robustness of the trained DL model. Thus, we designed two approaches, including OSAA methods and GEA. OSAA methods incorporates six different attacks to generate the AE. In our evaluation, we obtain a MR of up to 100% using these attacks. GEA approach focuses on preserving the functionality and practicality of the generated samples, which is not guaranteed in OSAA methods. Our evaluation showed that GEA is able to misclassify all malware samples as benign.

Acknowledgement. This work is supported by the NSF grant CNS-1809000, NRF grant 2016K1A1A2912757, Cyber Florida seed grant, and a gift from the NVIDIA GPU Program.

REFERENCES

[1] Ahmed Abusnaina, Aminollah Khormali, Hisham Alasmary, Jeman Park, Afsah Anwar, and Aziz Mohaisen. 2019. Adversarial Learning Attacks on Graph-based IoT Malware Detection Systems. In *39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019.*

[2] Hisham Alasmary, Afsah Anwar, Jeman Park, Jinchun Choi, DaeHun Nyang, and Aziz Mohaisen. 2018. Graph-Based Comparison of IoT and Android Malware. In *Proceedings of the 7th International Conference on Computational Data and Social Networks, CSoNet.* 259–272.

[3] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick D. McDaniel. 2017. Adversarial Examples for Malware Detection. In *22nd European Symposium on Research Computer Security.* 62–79.

[4] Aziz Mohaisen, Omar Alrawi, and Manar Mohaisen. 2015. AMAL: High-fidelity, behavior-based automated malware analysis and classification. *Computers & Security* 52 (2015), 251–266.