Automatic Malware Detection on an Alexa-Pi IoT Device

Mahshid Noorani*†, Spiros Mancoridis†, Steven Weber*
*Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA
†Department of Computer Science, Drexel University, Philadelphia, PA

***Abstract*—This work explores some of the security concerns pertaining to running software similar to Amazon Alexa home assistant on IoT-like platforms. We implement a behavioral-based malware detector and compare the effectiveness of different system attributes that are used in detecting malware, *i.e.,* system calls, network traffic, and the integration of system call and network traffic features. Given the small number of malware samples for IoT devices, we create a parameterizable malware sample that mimics Alexa behavior to varying degrees, while exfiltrating data from the device to a remote host. The performance of our anomaly detector is evaluated based on how well it determines the presence of our parameterized malware on an Alexa-enabled IoT device.**

I. Introduction

In this work, we built a one-class Support Vector Machine (SVM) for behavioral-based anomaly detection. The detector was separately trained on three sets of system attributes. We will describe how it is possible to detect the existence of malware on the Alexa device by first training an anomaly detector on operating system kernel layer system call data, then training an anomaly detector on network traffic data, and finally training an anomaly detector on a set of features integrated from system call and network features as shown on the left side of Figure 1.

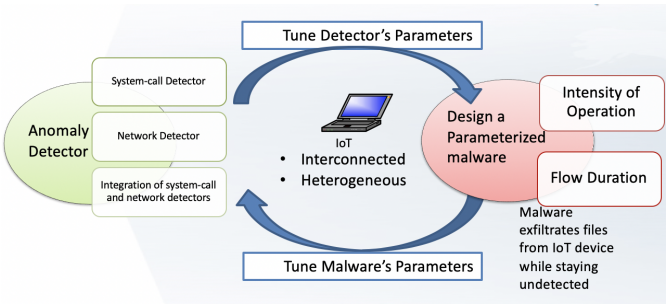


Fig. 1: Concurrent Design of Anomaly Detector & Parameterized Malware Sample

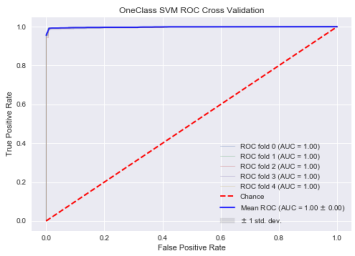
These anomaly detectors are trained using only benign (normal) usage scenarios. Each detector is subsequently subjected to both regular usage scenarios and malware infections in order to demonstrate its effectiveness to distinguish normal Alexa operation from anomalous, perhaps malicious, operation in real time. While we demonstrate the effectiveness of our technique on Amazon Alexa, we anticipate that small variations of our technique can be used to create anomaly detectors for other IoT devices, because these devices are typically highly specialized and, hence, exhibit a predictable normal behavior. This is in contrast to creating an anomaly detector for a general purpose computer, which has a more elaborate and varying behavioral profile.

To overcome the limitation of only having a small number of pre-existing malware samples for IoT devices, we decided to create a parameterizable malware sample that mimics Alexa behavior to various degrees, while exfiltrating data from the device to a remote host as shown in the right side of Figure 1. We tune the designed malware parameters to mimic Alexa behavior more closely based on the feedback received from our detector on which features are a better representation of Alexa behavior. The performance of each of our anomaly detectors is evaluated based on how well it determines the presence of our parameterized malware on an Alexa-enabled IoT device. In this work, we also have compiled a labeled dataset of Alexa-Pi system calls and network traffic features, which can be used in future studies of IoT devices.

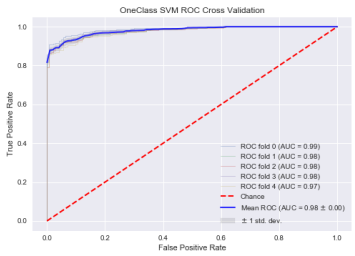
II. Accomplishments & Future Plans

This work focused on anomaly-based malware detection because of the limitations of traditional signature-based detectors (*e.g.,* ineffective on zero-day attacks) and the fact that current IoT devices do not yet have enough malware samples that can be used to build an accurate malware detection model. Therefore, our focus was on developing detection techniques that do not require a prior knowledge of specific threats, but rather are designed to detect notable deviations from normal Alexa's behavior.

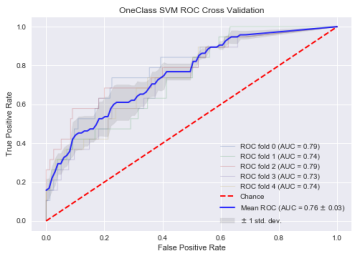
We designed a real IoT system and collected system call and network traffic data. We described the creation of behavioral anomaly detection systems designed to detect the execution of malware on an Amazon Alexa IoT device. Three detectors were designed based on different system features, *i.e.,* network traffic, system call, and an integration of system call and network traffic



(a) System calls in Idle Mode

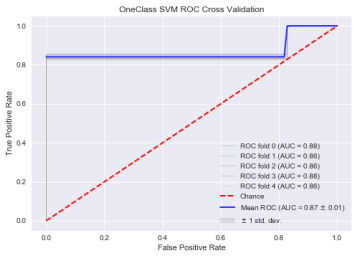


(b) System call in Ambient Mode

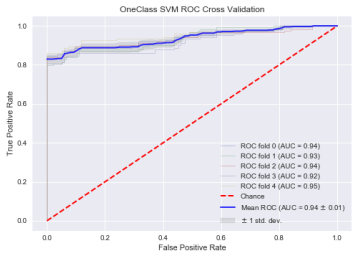


(c) System call in Query Mode

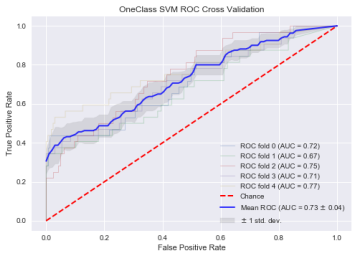
Fig. 2: System call ROC Curves in Each Mode



(a) Network Traffic in Idle Mode

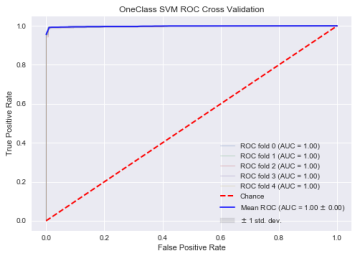


(b) Network Traffic in Ambient Mode

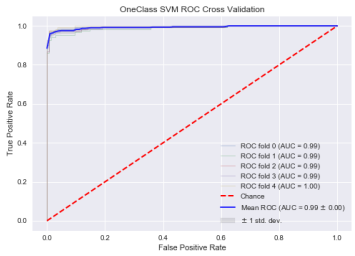


(c) Network Traffic in Query Mode

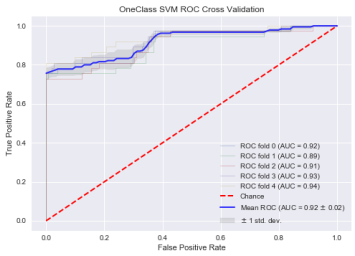
Fig. 3: Network Traffic ROC Curves in Each Mode



(a) Combined Features in Idle Mode



(b) Combined Features in Ambient Mode



(c) Combined Features in Query Mode

Fig. 4: Combined Features ROC Curves in Each Mode

features. We then created a parameterizable malware that mimics Alexa's behavior to varying degrees.

The key findings and contributions in this work are as follows: (I) We observed that a detector based on combined system call and network traffic features provides better detection compared to system call-based or network traffic-based detectors individually. (II) This work focused on the co-evolution of data exfiltration malware targeting IoT devices and anomaly detectors based on network and system call data. The focus of this work was not to optimize the detector or the malware, in general, but, rather, to evolve the malware based on the input from the detector, and vice versa. (III) We created a dataset consisting of network traffic and system call data to study IoT devices. We are planning to make this dataset available on GitHub.

In this work, we studied the cost and benefits of combining system call and network traffic features of an anomaly-based detector for IoT devices. However, we did not determine the optimal contribution of each specific detector to the combined detector.

ACKNOWLEDGMENT

The work was funded in part by Spiros Mancoridis' Auerbach Berger Chair in Cybersecurity.