**Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigen space Learning**

**Literature Review**

Malware detection methods can be static or dynamic. In dynamic malware detection approaches, the program is executed in a controlled environment (e.g., a virtual machine. The samples are available on https://github.com/azmoodeh/ IoT Malware Detection, where the benign samples are in binary and the malware samples are in OpCode or a sandbox) to collect its behavioural attributes such as required resources, execution path, and requested privilege, in order to classify a program as malware or benign . Static approaches (e.g., signature-based detection, byte-sequence n-gram analysis, opcode sequence identification and control flow graph traversal) statically inspect a program code to detect suspicious applications. David et al proposed Deep sign to automatically detect malware using a signature generation method. The latter creates a dataset based on behaviour logs of API calls, registry entries, web searches, port accesses, etc, in a sandbox and then convert’s logs to a binary vector. They used deep belief network for classification and reportedly achieved 98.6% accuracy. In another study, Pascanu et al. proposed a method to model malware execution using natural language modelling. They extracted relevant features using recurrent neural network to predict the next API calls. Then, both logistic regression and multi-layer perceptron’s were applied as the classification module on next API call prediction and using history of past events as features. It was reported that 98.3% true positive rate and 0.1% false positive rate were achieved. Demme et al. examined the feasibility of building a malware detector in IoT nodes’ hardware using performance counters as a learning feature and K-Nearest Neighbour, Decision Tree and Random Forest as classifiers. The reported accuracy rate for different malware family ranges from 25% to 100%. Alam et al. applied Random Forest on a dataset of Internet-connected smartphone devices to recognize malicious codes. They executed APKs in an Android emulator and recorded different features such as memory information, permission and network for classification, and evaluated their approach using different tree sizes. Their findings showed that the optimal classifier contains 40 trees, and 0.0171 of mean square root was achieved. In order to detect crypto-ransomware on Android devices as management nodes of an IoT networks, Azmoodeh et al. recorded the power usage of running processes and identified distinguishable local energy consumption patterns for benign applications and ransomware. They broke down the power usage pattern into sub-samples and classified them, as well as aggregating sub-samples’ labels to determine the final label. The proposed approach reportedly achieved 92.75% accuracy. The need to secure IoT backbone against malware attacks motivated Haddad Pajouh et al. to propose a two-layer dimension reduction and two-tier classification module to detect malicious activities. Specifically, the authors used Principle Component Analysis and Linear Discrimination Analysis to reduce the dataset and then used Naıve Bayes and K-Nearest Neighbour to classify samples. They achieved detection and false alarm rates of 84.86% and 4.86%, respectively. While OpCodes are considered an efficient feature for malware detection, there does not appear to have been any attempt to use OpCodes for IoT and IoBT malware detection. In addition, using deep learning for robust malware detection in IoT networks appears to be another understudied topic. Thus, in this paper, we seek to contribute to this gap by exploring the potential of using OpCodes as features for malware detection with deep Eigen space learning.