Detection of Malicious Powershell Scripts Using Machine Learning

Degree programme : BSc in Computer Science
Thesis advisor : Dr. Souhir Ben Souissi
Expert : Alain Joray

In recent years, PowerShell has systematically emerged as the tool of choice for many cyber-attackers. In this context, how can machine learning be used to provide support for the detection and analysis of malicious PowerShell scripts? This Bachelor‘s thesis investigates the question by exploring several natural language processing (NLP) and deep learning models aiming to classify PowerShell scripts as malicious or benign.

Motivation

In today‘s landscape, PowerShell has become a recurrent tool weaponized by threat actors, from first-stage droppers to destructive ransomware. Over 2022, Mandiant reported that PowerShell was used in 65% of the cases involving a scripting interpreter. Because of such a surge in the usage of PowerShell in every step of the cyber kill chain, giving defenders the tools to triage scripts at scale becomes critical.

Approach

Each sample is pre-processed and split into a set of tokens. For the NLP-based solutions using word embeddings, each token is attributed a feature vector, which is ultimately used to compute the sample features and build the classifiers. For the deep learning solutions using transformers, BERT and CodeBERT are trained onto the dataset using pre-trained tokenizers and models. Existing frameworks were used to extract features and generate the models, notably gensim, scikit-learn and Hugging Face. An overview of the data processing pipeline is shown in fig. 1.

Dataset

A corpus of over 15‘000 PowerShell samples was acquired from various sources and malware repositories over the Internet to train the models presented in this research project. For each sample, the source indicated if it was benign or malicious.

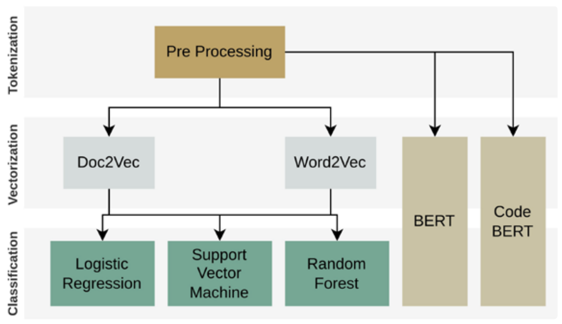

Fig.1 : An overview of the models analyzed and compared for this project

Results

Classification: Several word embedding algorithms and classifiers were explored to build a stable and efficient model. When classifying the sample as malicious or benign, the best results were achieved using Word2Vec for feature extraction coupled with Random Forest for the classification (Accuracy: 0.95) (fig. 2). BERT and CodeBERT (despite the lack of PowerShell support) both yielded exciting results (Accuracy 0.89 for both), albeit not as good as Random Forest,. The initial tokenization of the sample remains problematic as malicious scripts tend to be obfuscated. Exploring a tokenizer based on the script‘s AST could lead to more stable and precise results for future work on this topic.

Clustering: K-means, HDBSCAN, OPTICS, and BIRCH were used to identify clusters within the dataset, enabling users to get a potential attribution to a malware family for submitted samples.

Isabelle Mischler
Data Engineering
isabelle.mischler@protonmail.com

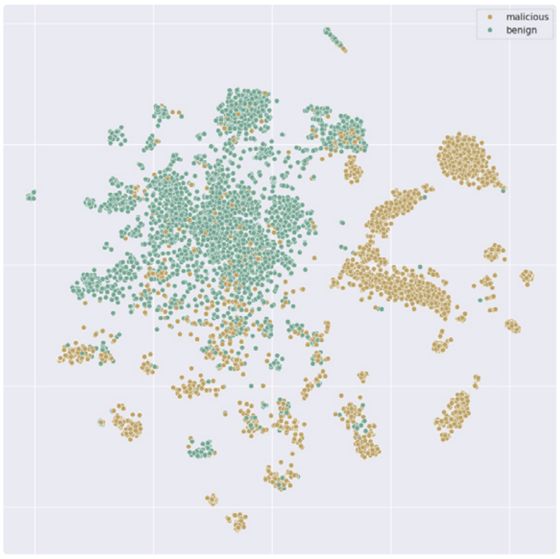

Fig.2 : A visualization of the classified dataset reduced to two features using PCA