Adversarial Examples on Malware Detection

or just why you shouldn't use Neural Networks for Security Purposes

-

Data Mining - Sapienza

Pietro Borrello - 1647357

February 11, 2019

Contents

1 Introduction

As widely known in the literature, machine learning algorithms, like deep neural networks, are in general susceptible to adversarial examples. It is possible to generate inputs, applying worst-case perturbations to existing ones, such that the crafted input is misclassified with high confidence [1]. Therefore an adversarial example given an input for a neural network (or any other ML model), is a perturbed version of the original one, such that preserves its functionalities, but is misclassified by the network. The problem arises in a large number of domain where neural networks are applied: for example, adversarial examples can be generated from images or speech, producing perfectly looking inputs undistinguishable on how they are perceived by humans, but that will be misclassified by the network.

In this paper we are going to investigate on how to produce adversarial examples in the case of Malware Detection. While when dealing with image or speech the goal is to apply small perturbations to the whole input not to change how it is perceived, when dealing with executable files, careful attention must be taken, since just a small modification to the binary values (i.e. changing an offset or an opcode in the executable), leads to complete changes in functionalities. Therefore classical adversarial examples generation, is not suitable for our goal, and must be tuned. Therefore our approach consisted in identifying which bytes in the malicious binary could be modified without changing its functionality, and how to map back these changes to the original binary, to produce a new malicious file, evading detection while retaining its malicious functionalities, misclassified as benign by the neural network.

2 Related Work

2.1 Adversarial Examples

A great part of machine learning models have been shown to be vulnerable to manipulations of their inputs to produce adversarial examples. Adding a carefully chosen manipulation to craft a humanly indistinguishable new input against a target model, leads it to consistently misclassify the crafted input. This behaviour has been explained to be caused by the inherent linearity of the components of the model, even when the components result in a non-linear model, as in neural network [1]. The goal of the misclassification can be targeted or un-targeted, depending on the fact whether the adversary

chooses a particular class to produce the input to be misclassified into, or not. In case of binary classification both targeted and un-targeted attacks produce the same result.

State of the art approach, have shown how adversarial examples can be transferable between models. It is shown how adversarial example, once crafted for a particular model trained on a particular dataset, remains misclassified with high probability, when analysed from both the same network trained on a different dataset, and from a different network trained on the same dataset. It has been shown also possible to generate an adversarial example in a black box fashion, without the knowledge neither of the underlying model, generating a synthetic dataset.

2.2 Malware Detection

The never ending cat and mouse game between malware writers and antivirus vendors, has seen an infinite list of techniques from both the parties. Any time malware detection system introduce a new method, a new malware evasion technique comes out from the hood to try to defeat the new method. As machine learning improves, neural networks seem the most promising step into precise malware detection, resilient to metamorphic code, or obfuscation techniques, producing encouraging results.

Malware Detection through machine learning model, can be based either on feature extracted from the inspected binary, or on the raw bytes itself. Since malware producer, knowing which features are collected for the classification, could efficiently hide the malicious behaviour (has done for existing anti-antivirus techniques), the most promising ML technique to analyse malware seems to be the one based on raw bytes.

References

[1] Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, arXiv:1412.6572

[2] Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples, Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, 2016

[3] `https://www.kaggle.com/lorerex/bitcoin-turtles-strategy-with-back-testing`