IMPROVING THE ARCHITECTURE FOR THE MALCONV MODEL TO DETECT MALWARE

Juan Mesa
Supervisor: Dr. Joseph Kehoe
M.Sc. in Data Science

1. Introduction

One of the most important tasks in Cybersecurity nowadays is to detect malicious software (Malware) for preventing data theft, infecting individual computers and computers from companies causing damages in these devices. Malware can be in different forms as the following:

- Virus: this kind of malware only infects clean files and they usually appear as an executable file (.exe).
- Trojans: This type of malware disguised as a known software and when it is executed allows other malware to let in the devices.
- Spyware: it is a type of malware that spies on the user. It takes notes what you do online, your passwords, credit card numbers or other sensitive data you can type on the web.
- Worms: this type of malware infects a network of devices, local or across the internet.
- Ransomware: This type of malware locks down the computer and shows a windows message indicating that it is not paid what the message says it will erase everything inside the device.
- Adware: this kind of malware is like an ad that is installed on the computer. It is not so malicious software, but it can allow other malware to get in the device.
- Botnets: They are networks of infected devices that normally are used by an attacker.

2. Research Question

Our focus in this project is to improve the architecture of the Malconv model to detect malware. Previous papers have worked with this architecture comparing to other approaches to detect malware and they have found out that Malconv architecture has improved the accuracy to predict and detect malware from other advanced Malware Detection techniques, such as N-grams.

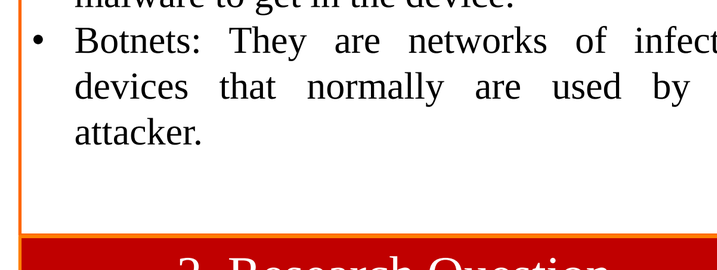

Figure 1: *High-Level Diagram of the Malconv Architecture.*
Source: DOI: 10.13140/RG.2.2.31250.71368

3. Review of previous works

- ***3.1 Malware Detection by eating a whole Exe:*** (Raff et al. 2017) took a static analysis approach, looked at the raw bytes of the file itself and built a neural network to determine maliciousness.
- ***3.2 Exploring Adversarial Examples in Malware Detection:*** (Suciu et al. 2019) highlighted architectural weakness in Malconv Model that can facilitate Adversarial Examples to change the input files of the Architecture intrinsically and modify the model.
- ***3.3 Deceiving End-to-End Deep Learning Malware Detectors using Adversarial Examples:*** (Kreuk et al. 2019) have found Deep Networks to be vulnerable to Adversarial Examples. They injected a small sequence of bytes to the binary file to modify malicious binaries.
- ***3.4 Explaining Vulnerabilities of Deep Learning to Adversarial Malware Binaries:*** (Demetrio et al. 2019) proposed an algorithm that generates adversarial malware binaries by changing few tens of bytes in the file Header.
- ***3.5 Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN:*** (Hu et al. 2017) proposed a generative adversarial network (GAN) based algorithm named MalGAN to generate adversarial malware examples.
- ***3.6 Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables:*** (Kolosnjaji et al. 2018) investigated the vulnerability of malware detection methods that use deep networks to learn from raw bytes and suggested a gradient-based attack to evade adversarial examples.
- ***3.7 Improved MalGAN: Avoiding Malware Detector by Leaning Cleanware Features:*** (Kawai et al.) proposed differentiated learning methods with different feature quantities and use one malware for MalGAN.

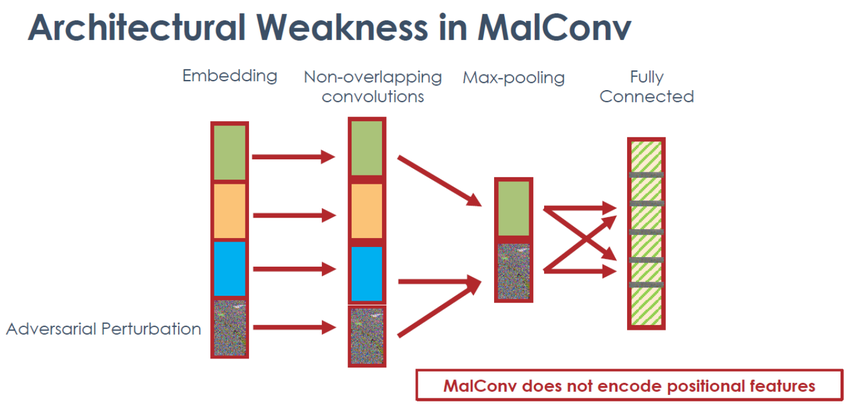

Figure 2: *Weakness in Malconv Model Architecture.*
Source: DOI: arXiv:1810.08280v3

4. Methodology

- Apply MalGAN along with Malconv Model to detect Malware.
- Change the Architecture of Malconv by incorporating encoding in positional features.
- Propose different learning methods with different feature quantities for each Malware.

5. Technologies

TensorFlow
python
Keras
scikit learn

6. Next Steps

- For future projects, create different learning methods and different feature quantities from the previous ones for each Malware.
- Propose a different algorithm to generate adversarial malware binaries.

References

[1] Demetrio, L., Biggio, B., Lagorio, G., Roli, F. and Armando, A. (2019) 'Explaining Vulnerabilities of Deep Learning to Adversarial Malware Binaries', in *3rd Italian Conference on Cyber Security, ITASEC 2019*, CEUR Workshop Proceedings,
[2] Hu, W. and Tan, Y. (2017) 'Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN', *arXiv.org*.
[3] Kawai, M., Ota, K. and Dong, M. 'Improved MalGAN: Avoiding Malware Detector by Leaning Cleanware Features', in IEEE.
[4] Kolosnjaji, B., Demontisy, A., Biggioyz, B., Maiorcay, D., Giorgio Giacintoyz, Eckert, C. and Roliyz, F. (2018) 'Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables', *arXiv.org*.
[5] Kreuk, F., Barak, A., Aviv, S., Baruch, M., Pinkas, B. and Keshet, J. (2019) 'Deceiving End-to-End Deep Learning Malware Detectors using Adversarial Examples', *arXiv.org*.
[6] Raff, E., Barker, J., Sylvester, J., Brandon, R., Cantanzaro, B. and Nicholas, C. (2017) 'Malware Detection by Eating a Whole EXE', in *AAAI*, 25 Oct 2017, ArXiv e-prints.
[7] Suciu, O., Coull, S. E. and Jones, J. (2019) 'Exploring Adversarial Examples in Malware Detection', in *IEEE Security and Privacy Workshops (SPW)*, IEEE .

Contact:
Juan Mesa
Phone: 0871754884
Email: c00245009@itcarlow.ie