BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE IN GERMAN

Cloud based malicious PDF Detection using Machine Learning

– Diploma thesis –

Author
Viorel GURDIS

2020

Abstract

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Context

1.2 Motivation

1.3 Paper structure and original contributions

Chapter 2

Scientific Problem

2.1 Problem definition

2.2 Background processes in Microsoft Windows

2.3 Filesystem monitoring

2.4 Analyzing PDF File Structure

2.5 Malware in PDF

Since PDF documents became a well known solution for information sharing, they also became a good target for cybercriminals.

2.6 Machine Learning for Malware Detection

2.7 Benefits of Cloud Computing

Chapter 3

Related work

Beginning with the first reported occurences of the malware, the security researchers have made a huge effort to prevent the harmful behavior. Over the years malware has evolved in different forms and of course the antivirus industry has developed more complex detection solutions. Since PDF documents became a good target for cybercriminals, more and more specialists pay attention to the analysis of dangerous PDFs. Integration of so many analysis tools in a single antivirus software has a negative impact on computers performance. For this reason, the cybersecurity field attaches importance to Cloud Computing. In the following sections we will analyze other academic and industrial approaches for transfer of antimalware engines to the cloud and some documented detection techniques of malicious PDFs.

3.1 Cloud based malware detection

In the field of Cloud solutions for malware detection, there is sparsed amount of shared academic works. As compensation, in the antivirus industry there is a fast growing interest for high computation to run more advanced and complex detection algorithms, which is provided by Cloud Computation. An important example is **VirusTotal**, a popular Cloud application developed by Hispasec Sistemas [6]

3.2 Detection of malicious PDF

PJScan, presented in the paper of Laskov and Srndic [4], is a Machine Learning approach that trains One-Class Support Vector Machine ($OCSVM$) to classify PDF files. This approach is focused on static analysis of embedded Javascript code, as it is known for the ability of integrating malicious behavior in PDF documents. The authors used *n-gram* analysis to extract lexical features, such as Javascript operators and other tokens. The obtained sequence of features served later as input for the machine learning algorithm. The trained model can

correctly classify malicious samples containing Javascript code. However PJScan has a lower accuracy, because it is not able to detect obfuscated parts and there are also some samples containing other types of malicious payload, such as SWF [1].

Another example of static analysis of PDF Structure is **PDF Tools** by Didier Stevens [7]. It represents a suite of tools for scanning, parsing and dumping PDF files. This approach focuses on identifying the fundamental elements of the format, such as Streams, Cross Reference Tables etc. The advantage of the PDF Tools is their ability of name obfuscation handling and their simplicity. Because of their high speed performance, PDF Tools are largely used in cybersecurity research.

A completely new approach mentioned in the research paper of Fettaya et al. [2] describes an algorithm that uses Convolutional Neural Network (*CNN*) to detect malicious PDF files. The trained model, based on a single convolutional layer with a global max pool and a linear layer doesn't require any data preoprocessing. Instead of this, the model is trained using as input the binary representations of the files. It is worth noting, that the described algorithm could be efficiently used for distinguishing various families of malware. The rate of correct detections achieve 94%, the algorithm being also capable to classify aprox. 80% of the malware into different categories.

[1] Adobe Flash file format used for multimedia

Chapter 4

Proposed approach

4.1 Dataset

4.2 Proof of Concept

4.2.1 Feature Selection

4.2.2 Classification Techniques

4.2.3 Performance Evaluation

4.2.4 Experiment

4.3 Used technologies

4.3.1 Microsoft .NET Framework

[9]

4.3.2 PDF Tools and Metasploit Framework

[5] [10] [3]

4.3.3 Python Flask

4.3.4 Scikit-learn Machine Learning Library

[1] [8]

4.3.5 ReactJS Framework

Chapter 5

Application

5.1 Design

5.2 Windows Service

5.3 Cloud API

5.4 Dashboard Interface

Chapter 6

Conclusion and future work

Bibliography

[1] Clarance Chio and David Freeman. *Machine Learning and Security*. O'Reilly Media, Inc., 2018.

[2] R. Fettaya and Y. Mansour. Detecting malicious pdf using cnn. *International Conference on Learning Representations*, 2020.

[3] Adobe Systems Incorporated. *PDF Reference, sixth edition: Adobe Portable Document Format Version 1.7*. Adobe, 2006.

[4] P. Laskov and N. Srndic. Static detection of malicious javascript-bearing pdf documents. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 373–382. ACM, 2011.

[5] Xakep Magazine. Looking for exploits in pdf-documents on our own. https://xakep.ru/2014/09/26/search-document-exploit/, 2014.

[6] Hispasec Sistemas. Virustotal. https://www.virustotal.com/. [Online; Accessed 1-April-2020].

[7] Didier Stevens. Pdf tools. https://blog.didierstevens.com/programs/pdf-tools/, 2008.

[8] Emmanuel Tsukerman. *Machine Learning for Cybersecurity Cookbook*. Packt Publishing, 2019.

[9] Pavel Yosifovich, Alex Ionescu, Mark E. Russinovich, and David A. Solomon. *Windows Internals*. Microsoft Press, 2017.

[10] Lenny Zeltser. https://zeltser.com/information-security/. [Online; Accessed 1-April-2020].