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

The informational technology progress, that is in continous growth, brings a lot of benefits along with new responsabilities. There are plenty of applications that we use everyday across the entire World Wide Web. We don't even realize how much of our personal information is transfered to the virtual environment. That being said, it's important to be prepared for cybersecurity threats that can misuse our sensitive data. The problem is that the cyber attackers develop a lot of *hacking*[1] techniques, which are becoming increasingly difficult to detect. The popular software applications are the best target to inject malicious behavior. This happened with the Adobe PDF format. PDF documents are well known, trustworthy files and they are a global solution for sharing information. There are a lot of PDF readers and even browsers and email applications have support to open these files for viewing. It became so convenient to work with this format, that users interact with PDF documents without noticing any possible danger. However PDF is a often used attack vector. The large number of discovered PDF vulnerabilities and also the support of embedding Javascript code into documents are just some of the most exploited methods.

1.2 Motivation

Under the guise of seeming harmless, PDF documents are used on a daily basis across numerous public institutions, private companies and for personal purposes. Most of the people don't consider that these files could be dangerous and just copy the documents to their computers and access them. It is enough for one PDF to be malicious and the entire network affiliated to an institution becomes compromised. The cyber attackers succeed in achieving their goals,

[1] attempt to gain unauthorized access to data in a system

but the victim institution requires huge resources, both financially and time wise, to restore the integrity and security of their infrastructure. A measure to combat the described situation is having an active real time protection installed on the computer. The most common solution, antivirus applications, work by signature matching, which is effective for detecting previously identified malware[2]. This means that all the antivirus applications require permanent updates in order to keep their malware databases up-to-date.

1.3 Paper structure and original contributions

[2]any software intentionally designed to cause damage to a computer, server, client, or computer network

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

2.6 Machine Learning for Malware Detection

2.7 Benefits of Cloud Computing

Chapter 3

Related work

Beginning with the first reported occurences of 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 transfering 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 a sparse amount of shared academic works. As compensation, in the antivirus industry there is a fast growing interest for Cloud Computing, which has higher computational power and could therefore run more advanced and complex detection algorithms.

An important example is **VirusTotal** [7], a popular Cloud application developed by Hispasec Sistemas, that aggregates more than 50 antivirus engines and makes them publicly available for scanning uploaded files. From the user perspective, this is an excellent application, that can extract metadata of the submitted file and can identify any dubious signal. The provided result represents a comparation between analysis verdicts of cybersecurity market leaders. Of course this is an advantage in terms of the scanning result precision and respectively a gain regarding spent computational time. On average it takes aprox. 55 seconds to upload and analyze a 400KB file. VirusTotal is also helping to maintain the global cybersecurity at a high level, by sharing all the submitted suspicious files with the security researchers. Thereby the antivirus engines will be permanently improved.

Sandbox Analyzer is another antimalware cloud solution developed by Bitdefender [1]. It's approach is a bit different, because it ensures the security on a private network, where the Bitdefender product is installed, thus not being available for public access. It's operating principle is also specific, by preventing the execution of threats on an endpoint and automatically sending of harmful files to the Cloud. After extra analysis in the Cloud, the Sandbox Analyzer can take remediation action based on the verdict. In that way, malicious files get disinfected, quarantined or deleted. One of the benefits for this solution is in-depth analysis of malicious files in an isolated environment, rather than on user's machine. Thereby the risk for performance implication, as well as the risk of accidental run of malware on an endpoint machine are eliminated. At the core of the Sandbox Analyzer there are Machine Learning algorithms and dynamic behavior analysis techniques that are constantly improved to detect fresh threats.

3.2 Detection of malicious PDF

PJScan, presented in the paper of Laskov and Srndic [5], 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 [8]. 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 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. [3] describes an algorithm that uses a 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 achieves 94%, the algorithm being also capable to classify aprox. 80% of the malware into

[1] Adobe Flash file format used for multimedia

different categories.

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

[10]

4.3.2 PDF Tools and Metasploit Framework

[6] [11] [4]

4.3.3 Python Flask

4.3.4 Scikit-learn Machine Learning Library

[2] [9]

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] Bitdefender. Sandbox analyzer. https://download.bitdefender.com/resources/files/News/CaseStudies/2017-TechnicalBrief-SandBoxAnalyzer-crea2103-A4-en-EN-2-GenericUse.pdf. [Online; Accessed 1-April-2020].

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

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

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

[5] 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.

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

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

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

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

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

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