Malware Research in PDF Files

Shir Bentabou Alexey Titov
302895529 334063021
Advisor: Ph.D. Amit Dvir and Ph.D. Ran Dubin
Ariel University,Department of Computer Sciences,40700 Ariel, Israel

Abstract

Cyber is a prefix used in a growing number of terms that describe new actions that are being made possible by the usage of computers and networks. The main terms in those are cyber crime, cyber attack and cyber warfare, all of those can be carried out by malwares.

Malware, or malicious software, is any software intentionally designed to invade, cause damage, or disable computers, mobile devices, server, client,or computer network. Malware does the damage after it is implanted or introduced in some way into a target's computer. Nowadays there are many distribution strategies for malwares and many programs are used as platforms. Some of these programs are in the user's everyday use, and seem pretty innocent. In our project we will focus on PDF files as a platform for malware distribution.

PDF, Portable Document Format, is used for over 20 years worldwide, and has become one of the leading standards for the dissemination of textual documents. A typical user uses this format due to its flexibility and functionality, but it also attracts hackers who exploit various types of vulnerabilities available in this format, causing PDF to be one of the leading vectors of malicious code distribution.

Users normally open PDF files because they have confidence in this format, and thus allow malwares to run due to vulnerabilities found in the readers. Therefore, many threat analysis platforms are trying to identify the main functions that characterize the behavior of malicious PDF files by analyzing their contents, in order to learn how to automatically recognize old and new attacks.

The target of our work is to test and analyze how the use of machine learning methods can lead to effective recognition of malwares in PDF documents.

1 Introduction

Cyber is a prefix used in a growing number of terms that describe new actions that are being made possible by the usage of computers and networks. As the evolution of modern computers and networks progressed, a cat-and-mouse game evolved simultaneously, and continues to this day. This cat-and-mouse game is cyber warfare, and is caused by the challenges in information security, as a result of the various cyber threats that exist.

The first known cyber-attack is the Morris worm, when a student in Cornell university wanted to know how many devices existed in the internet. He wrote a program that passed between computers and this program asked each device it reached to send a signal to a control server that counted the signals sent to him. Nowadays the growth in number of cyber-attacks is unimaginable; according to published data by Check Point Software Technologies, there were 23,208,628 attacks in February 23rd, 2019 alone.

The growth in cyber-attacks is not only in the numeric value, but also in the different kinds of attacks that exist. There are many different types of threats, and usually they consist of one or more kinds of attacks in the following list:

Advanced Persistent Threats	DDoS	Intellectual Property Theft	Rogue Software
Phishing	Wiper Attacks	Spyware/Malware	Unpatched Software
Trojans	Money Theft	MITM	
Botnets	Data Manipulation	Drive-By Downloads	
Ransomware	Data Destruction	Malvertising	

Phishing is one of the popular distribution methods for malware. It is the fraudulent attempt to obtain sensitive information from a target by disguising as a trustworthy entity in electronic communication. Phishing is typically carried out by e-mail spoofing, and often directs users to enter personal information at a fake website that looks identical to the legitimate website.

Threats as these come from various sources. The profile of the attackers does not match one certain type and depends on the source's interests and available technology. The most common sources of cyber threats are: nation states or national governments, terrorists, industrial spies, organized crime groups, hacktivists and hackers, business competitors, and disgruntled insiders.

Malware, or malicious software, is any software designed to serve any kind of cyber-attack. Software is considered malware based on the intent of the creator rather than its actual features. Different kinds of malware serve for different uses, and it does the damage after it is implanted or introduced in some way into a target's computer.

The first recorded malware was named Elk Cloner, created by 15-year-old high school student Rich Skrenta as a prank, and affected Apple II systems in

1982. This virus was disseminated by infected floppy disks and spread to all the disks that were attached to the system by attaching itself to the OS.

What started as a teenage prank in 1982, has evolved into a wide range of variated software that is used today as malwares. The main types of malwares that exist today are:

- Trojan Horse – This type of malware infects a computer and usually runs in the background, sometimes for long periods of time, and gains unauthorized access to the affected computer, gathers information about the user / machine it is installed on. The information gathered by the Trojan is then sent to the attacker, normally a server side that stores the data for the attacker.
- Virus - A virus is software usually hidden within another program that can produce copies of itself and insert them into other programs or files, and usually performs a harmful action.
- Worm - Similar to viruses, worms self-replicate in order to spread to other computers over a network, usually causing harm by destroying data and files.
- Spyware - Malware that secretly observes the computer user's activities without permission and reports it to the software's author.
- Exploits - Malware that takes advantage of bugs and vulnerabilities in a system in order to allow the exploit's creator to take control.
- Ransomware - Malware that locks you out of your device and/or encrypts your files, then forces you to pay a ransom to get them back.

All the above methods intend to distribute malware to systems, normally without the users being aware that the process has happened at all. In order to do that, the platforms used in these methods are well known files and programs in the user's daily use. A very popular file type for distributing malware is PDF, Portable Document Format. In our project we will focus on PDF files as a platform for malware distribution.

PDF, Portable Document Format, is a file format used for over 20 years worldwide, and has become one of the leading standards for the dissemination of textual documents. Based on the PostScript language, each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, vector graphics, raster images and other information needed to display it. PDF files are composed by a set of sections:

Header
Body
‘xref’ Table
Trailer

1. **PDF Header** - This is the first line of a PDF file and it specifies the version number of the used PDF specification which the document uses (e.g. ”

2. **PDF Body** - The body of the PDF document is composed of objects that typically include text streams, fonts, images, multimedia elements, etc. The Body section is used to hold all the document’s data being shown to the user. Notice that streams are interesting to us in the security aspect because they can store a large amount of data and thus store executable code that runs after some event.

3. **Cross-Reference Table** – Also called ‘xref’ Table, this table contains the references to all the objects in the document. The purpose of a cross reference table is that it allows random access to objects in the document, so we don’t need to read the whole PDF document to locate an object. Each object is represented by one entry in the table, which is always 20 bytes long. If the document changes, the table is updated automatically.

4. **Trailer** – This section specifies how the application reading the PDF document should find the cross-reference table and other special objects in the document. The trailer section also contains the EOF indicator.

Here [**?**] is an example of a PDF file. This simple example gives us a notion of how the PDF files look like before they are parsed, and can be seen for every PDF file, opening it with notepad. In more complex files, it is possible to see different kinds of objects in the body section. Every object resides between obj – endobj tags and contains different kinds of data. Streams for example, can contain large amounts of data (even executable code) and is normally compressed do that it is not readable without using some tool to decompress the data.

PDF is used widely around the world since it’s creation, and still is because of two main advantages that it provides compared to other file formats: (1) **PDF files are compatible across multiple platforms** - A PDF format represents a document independently of the hardware, operating system and application software used to create the original PDF file. It was designed to create transferable documents that can be shared across multiple computer platforms. (2) **The software for viewing PDF file is free** - Most PDF Readers, including Adobe Reader, are free to the public. This ensures that anyone you send the file to will be able to see the full version of your document.

A typical user uses this format due to its flexibility and functionality, but it also attracts hackers from the same reasons: it enables cross platform attacks and is widely used (including specific targets for attacks) because many readers are free. Moreover, there are various types of vulnerabilities available in this format, causing PDF to be one of the leading vectors of malicious code distribution. The vulnerabilities available in PDF derive from PDF's support of various types of data in addition to text such as JavaScript, Flash, media files, interactive forms or links to external files and URLs.

In addition to that, PDF files are believed to be less suspicious than executable files. It is a common security practice for an IT administrator to define a policy to block executable files from staff e-mail attachments or web downloads, but it is rare to block PDF documents in such a manner. Users normally open PDF files because they have confidence in this format, and thus allow malwares to run due to vulnerabilities found in the readers. Therefore, many threat analysis platforms are trying to identify the main functions that characterize the identity and behavior of malicious PDF files by analyzing their contents, in order to learn how to automatically recognize old and new attacks.

2 Related Work

In this project we will focus on phishing carried out through PDF files, that means making the user download some file or entering an unsecure website not knowingly. This can be done easily using PDF because PDF files allow hyperlinks to be embedded in them for easy use, and also enables the use of JavaScript and PDF object streams in the body part of the file, and this way code can be executed without the user knowing, or with his knowledge but without the awareness that this could be unsafe for him. URL's are a main method for this kind of phishing, as the user can click or hover over a hyperlink in a PDF file and be directed to a malicious website or to download a malicious file. Moreover, because of the JavaScript and streams that can be part of the PDF file, this can happen without the user knowing, in the background.

In general, there are many tools available in the software market for classifying files and URL's as malicious or benign. Anti-virus products normally work in a static way, producing signatures for the identification of malicious files/URLs. In addition to that, dictionaries are in use in a remote server, that update all the time and signatures use these dictionaries in order to identify malicious files/URLs. Most of the AV's also use blacklists, these lists contain URLs, IPs and more, and every time the AV's identify a communication with some address in the blacklists, they will block it. Apart from the static tools there are dynamic analysis tools using sandbox environments that examine the behavior of a file/URL in a separated environment.

In previous works, we have seen projects that were meant to identify malicious PDF files, and in surveys we have read, there have been various attempts to identify malicious files by using a range of features of the PDF file (notice that features are tags used in the objects of the file). These features were combined to create a feature vector for the files and provided as input to machine learning algorithms (from variated kinds) that classified the files benign or malicious. Most of these surveys based their work on Didier Stevens and Otsubo's research and tools. The objective of these surveys was to create algorithms that identify malicious files with low FN (False Positive) and FN (False Negative) rates and classify files efficiently with the shortest time and resources needed.

In Torres and De Los Santos research [**?**], they have combined four different PDF analysis tools, different sets of features (based on 21 features Didier Stevens chose in his research), and three algorithms (Support Vector Machine, Random Forest, Multilayer Perceptron) to find the most efficient way to classify if a PDF is malicious or benign, using machine learning. The achieved results are shown in the table below. In their research, MLP algorithm showed the best results.

Algorithm	Accuracy	Recall	F1-Score	ROC-AUC
SVM	0.50	0	0	0.70
RF	0.92	0.94	0.92	0.98
MLP	0.96	0.967	0.96	0.98

In another aspect of our work, we have read surveys about URL analysis, and how to classify if a URL is malicious or benign. From the surveys we have read we have seen that not only static analysis tools and blacklists were used, but also a lexicographic approach was used in most of them, to improve the success rates of classifying URL's.

In D. R. Patil and J. B. Patil's research [**?**], they have provided an effective hybrid methodology to classify URL's. They have used supervised decision tree learning classification models and performed the experiments on a balanced dataset. The experimental results show that, by including new features, the decision tree learning classifiers worked well on the dataset, achieving 98

The subject of our project has not yet been researched, therefore specific researches about it haven't been found. This means we will enter a new field of research, hoping to reach some progress in finding an efficient and effective way to identify these kinds of attacks.

3 Contribution

Phishing in PDF can occur in different methods, as written above it can happen by downloading a malicious file or entering a malicious website through links that can appear in the file. The links can be in the text of the PDF file, or in the streams and objects we have discussed before. The attacks can be visible to the user and require some action from the user or be completely invisible and happen in the background without the user knowing.

In this project we will focus on three ways to identify these attacks in PDF files:

1. Links in the text of the file, and text detection in general – In the PDF file text we can extract URLs that are written inside them and check if

they are malicious, and furthermore we can learn about the file from the text inside, the same way as spam filters for e-mails work.

2. Links in the objects of the PDF file – These links can be found inside the objects of the file in the body section and can be completely hidden from the user. Our aim is to extract text from the files and create a text engine that will be able to extract the URLs from them efficiently.

3. Preview of the files – Anti-viruses work by creating hashes (such as MD5) for malicious files found. For every malicious file they detect, they create a hash for it, and store it in their databases, so that if they encounter these files again, they will be able to block or warn about them. The problem of this approach is that if a single attribute of the file is changed, the hash also changes, and this way a file can be only briefly changed and pass the AVs detection. There exists an option to open a PDF file for initial preview, without opening the file itself. It shows the first page of the PDF file. Please note that 'previewing' the file is not proven to safe.

   As AVs check the files by hash that can be easily changed, we want to create a detection based on the content of the file. That means, we want to be able to extract previews for the PDF files (in the form of pictures), extract text from these pictures and by the content of the files, be able to detect them. Our aim here is to create an efficient image similarity engine, to detect files by their image, and also extract text from their image and detect malware in the text itself.

4 Work Plan

Since we will work on this project with an external guide, we have two strict schedules we will need to stick to: