Research for a final project

Raz Elbaz
Ariel University
raz4447@gmail.com

February 2023

Abstract

With the advancement of technology, new malware technologies and ways to carry out attacks are emerging. MP4 is the most common video file format due to the fact that it provides high-quality video in a relatively small size compared to other video formats, and provides lossless compression after editing and re-saving the files. Video files are not generally considered malicious file types, but it is possible for malware to be embedded in a video file or disguised as a video file. Therefore, MP4 is considered a good candidate for embedding video malware. This paper will deal with the presentation of the various ways in which the use of video, in particular in the MP4 format, has been used to introduce malware, and the presentation of a learning machine designed to detect malware in video. Our learning machine detects potential malware and exploits, with an approach of analyzing the possibilities for insertion and existing attacks in order to label the data and train the learning machine in an optimal way.

1 INTRODUCTION AND RELATED BACKGROUND

MP4 is a container file format that allows multiple data streams to be embedded in a single file and is extensible, meaning it does not define a strict structure and allows a custom structure and hierarchy for each media type. The data in an MP4 file is divided into two parts: (1) the media-related data: includes streams of video, audio, and subtitles; (2) metadata: specifies information about the original data which assists in identifying the nature and features of that data and making it easier to use, search, and manage the video such as flags, timestamps, bit rate and so forth.

The structures in MP4 are typically referred to as atoms or boxes and no data is outside a box. Atoms (or boxes) are data within the video file container that contain specific information about the parameters of the video file. These are descriptive atoms and are not the actual media data. Minimum atom size is 8 bytes as first 4 bytes specify size of the atom and next 4 bytes specify its type. There is a convention that the first box in an MP4 file is an ftyp box. ftyp box, identifying itself as an MP4 and providing additional compatibility information - information about the file type, description, and the common data structures used. After the ftyp box usually comes a moov box, containing metadata organized in a nested structure of other - a few trak boxes which provide reference information for interpreting the encoded data streams. The trak including in addition offset information into the rest of the file (like pointers to a video player) about where the encoded streams can be found. The actual encoded bitstreams are, in turn, contained within the following mdat box , containing the audio or video payload.

In order to embed a malicious payload in MP4, you can try to use steganography . Steganography is the process of secretly embedding information inside a data source without changing its perceptual quality so it is difficult for the human eye to detect the hiding of information. The most common use of steganography is to hide information that can be a file, malicious code, message, image or video and usually converts them to alternative equivalent multimedia files such as images, video or audio. There are many attacks that exploit the structure of mp4 to carry out attacks. One of the possibilities to carry out an attack [8] is to use a new technique to insert the information into The mp4 video files. This technique is based on changing the GOP (group of pictures) structure and exchanging the streams in the video file. Other options include data hiding methods based on audio and video synchronization in the MP4 container [6] or method for data hiding in video by utilizing the least significant

bit (LSB) method [13]. Damage in the video may be in 3 possible places: header, video data and end of file, therefore these will be part of our tests. We will want to detect the introduction of malware using different detection methods to train our learning machine and make it recognize when a video file contains a malicious file.

2 RELATED WORK

There are many different ways to insert malicious code into a video, as a result different methods have been developed to detect them. We will first present works for methods of introducing malicious code and then methods for detection. The simplest method is to hide the data in the Least Significant Bits (LSB). [13] [12] Both presented a technique by utilizing the least significant bit (LSB) method. The first also used Using Knight Tour Algorithm while the second was based on Hash function. Another method [10] of embedding into a file used a random number generator function in a method called random pixel embedding to hide secret text over a video file. In addition, there is a method that called TCStego [9]. One of the latest weapons is a new technology that combines the power of TrueCrypt® (one of the best known and easy to use encryption programs) with a steganography twist. This latest advancement hides a TrueCrypt container inside an existing MP4 or QuickTime multimedia file. In this part of the book, an open source code of a simple script in the Python programming language that detects such anomalies is published.

In one of the articles on detection [2] they stated that while it is difficult to discover TCSteg, using their detection method it is possible to estimate the existence of the hidden volume. They also concluded that although there are not many MP4 steganography analysis tools on the market, there are some already existing applications that can be used for this purpose. One of the studies [11] on the subject of a detection method discusses the detection of digital video manipulation while referring to blind methods for detecting video forgery. Different frameworks of video forgery detection methods are classified and generalized in this paper and the performance of some typical video forgery detection methods are compared. [7] Proposed a verification system for MP4 by jointly using data embedding in different media tracks (ie video and audio tracks) as well as different layers (ie compression/media and system layers). In another paper [4] proposed a passive forensic method for object-based video spoofing. [1] Reviewed journal articles from 2015 to 2021 for steganography and steganalysis using deep lean based algorithms. They analyzed the combination of image and video steganography using ML and DL algorithms to hide the data within media files such as audio, video, images, etc. [3] Proposed the use of color detection in the LSB bits (LSB Color Detection) to adjust ratios of concealing data according to respective pixel RGB values. In the article [5] present a survey on passive detection methods of video tampering. Passive video tampering detection methods are classified into the following three categories based on the type of spoofing they address: detection of double or multi-compressed videos, region tampering detection, and video inter-frame spoofing detection.

Our goal is to learn the possibilities of introducing malicious code and the possibilities of detecting so that we can classify with a learning machine every MP4 video file as to whether it contains malware or not.

3 METHODOLOGY

As mentioned before, our paper aims to develop a learning machine that can identify in video files (1) malicious code - unwanted files or programs that can cause damage to the computer or endanger data stored on the computer. Different classifications of malicious code include viruses, worms, and Trojan horses. (2) Exploitation - a video is a legitimate file, hiding malware in it is an exploitation of a user who does not suspect that it contains something malicious and dangerous. (3) Hidden messages-steganographic data.

The extraction of the data from the video files in the dataset included the extraction of the metadata of the video, the title of the video file and the end of the file, which malware commonly exploits. However, advanced malware and data hiding methods use the video content to evade detection. Therefore, the learning machine will generate features that are built from all the data that an mp4 file contains: metadata and media-related data. Some malicious files do not follow the structure of a video file or have an unreasonable video size. For example, we will analyze the bit rate of an MP4 file, which will give us an indication of what we should expect for the size of the file. For example, if we have a file where both the file's audio and video streams have a combined bit rate of 1 megabyte per second, a length of 10 seconds, and a file size of 10 megabytes, we can say that the file appears to be in order. If we had a file with the same attributes, but were 20MB in size, we could say the file has 10MB uncaptured. Any video out of range is considered malicious by the learning machine. For evaluation and creation of new features, we send the files in the dataset to cloudmersive, which knows if the file contains malicious code. The study takes a qualitative approach and therefore the size of the dataset was not huge. Investigating video and file malware is a wide-ranging topic that involves a lot of ongoing research because the ways to do it are constantly evolving. The goal was to create a high-quality learning

machine that knows how to analyze the information of the video file from many and varied directions, so the approach was to focus on those ways to introduce maliciousness into the video, learn the structure of the video, and learn how to export the information while creating significant and high-quality features in order to achieve maximum results.

4 EVALUATION

4.1 DATASET DESCRIPTION

The dataset in this paper will be 333 videos in MP4 format. Since there is a shortage of finding malicious videos to download, we had to create a few, and the ones we could find we included in the dataset. The data set includes: videos that were previously malicious, normal videos that were downloaded to train the learning machine, and new malicious videos that we created according to the following steps :Mining frames from the videos using a built-in python library in OpenCv, mining audio from videos, encoding the information in the lsb bit of one of the frames, and creating videos from frames while maintaining the video quality and original file size. An MP4 is a common "container format" for video files that allows you to store a lot of video and audio information in a smaller file size. The minimum file size in the dataset is 17069 bits and the maximum 30217369 bits.

MP4 videos can be played universally on any device, therefore we will present a warning regarding the dataset: it may contain viruses therefore the work environment in which we will test and investigate it will be a virtual machine in order not to damage the computer.

4.2 CREATING FEATURES

As stated in the methodology section, in order to train the learning machine to recognize the existence of malicious or embedded code we export the information from each video file as features. To simulate steganography we embed messages inside regular video files. To export the media related data we will use the FFmpeg library - a free and open source software project consisting of a package of libraries and software for handling video, audio and other multimedia files and streams. At its core is the ffmpeg command line tool itself, designed for processing video and audio files; we used the ffprobe method that collects information from multimedia streams and prints it in a human and machine readable manner. To export the metadata we used the library os in Python which provides functions to interact with the operating system. This module provides a portable way to use OS-dependent functionality; we used the stat() function which is used to acquire all the information about a file or folder. To find malicious code in the video we will use the Cloudmersive API. Cloudmersive Virus Scanning covers millions of virus and malware signatures, multi-threat and multi-factor scanning, processing is not done in memory and therefore ensures fast performance and strong security. In addition, we will use the struct module to extract basic information from a bit header.

5 RESULTS

In this section, we will review the results of the ML.

Confusion matrix			
precision	recall	f1-score	support
benign	0.97619	1.00000	0.98795
malware	1.00000	0.94444	0.97143
accuracy			0.98305
macro avg	0.98810	0.97222	0.97969
weighted avg	0.98345	0.98305	0.98291

6 SUMMARY

In this study, a learning machine was presented to detect malware in video (mp4 format). Based on a dataset containing malicious and non-malicious videos, some of which were manually generated. Shown were the ways to export the file properties and file behavior to produce features that would allow us to train our learning machine to best classify each file. During the preliminary investigation, additional ideas and possibilities for "suspicious behaviors" were raised that were investigated in order to achieve maximum results, among other things, deep learning was carried out about the format and finding possibilities for the introduction of malware and video.

References

[1] K. Fathima P. Sathish Kumar B. Karthik S. Siva Kumar Ankush Ghosh, B. Sowmya. Studies on steganography images and videos using deep learning techniques.

[2] Alfredo Fernandez Anthony Ramirez. Mp4 steganography: Analyzing and detecting tcsteg.

[3] Phet Imtongkhua1 Chatchai Poonriboon Chakchai So-In Paramate Horkaew2 Arada Suttichaiya1, Yuwarat Sombatkiripaiboon1. Video steganography with lsb color detection.

[4] Zhu Ningbo Chen Richao, Yang Gaobo. Detection of object-based manipulation by the statistical features of object contour. *School of Information Science and Engineering, Hunan University, Changsha 410082, China.*

[5] Zhu Ningbo Chen Richao, Yang Gaobo. Digital video tampering detection: An overview of passive techniques.

[6] Kiyoshi Tanaka Imdad MaungMaung, KokSheik Wong. Reversible data hiding methods based on audio and video synchronization in mp4 container. *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS).*

[7] KokSheik Wong Imdad MaungMaung, Yiqi Tew. Authentication of mp4 file by joint data embedding in audio and video tracks. *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA).*

[8] M. Jokay. The design of a steganographic system based on the internal mp4 file structures. *INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS Issue 4, Volume 5, 2011.*

[9] Chet Hosme Michael Raggo. Elsevier, 1st edition - november 9, 2012 edition, 2012.

[10] Edy Suharto Kharis Khasburrahman Muhammad Khaerul Anam, Eko Adi Sarwoko. Random pixel embedding for hiding secret text over video file. *2017 1st International Conference on Informatics and Computational Sciences (ICICoS).*

[11] GHAZALI SULONG2 OMAR ISMAEL AL-SANJARY1. Detection of video forgery: A review of literature. 2015 publisher=jatit LLS.

[12] Bhagyashri Rahangdale Prof. Dr. P. R. Deshmukh. Data hiding using video steganography.

[13] Ghada Thanoon Younus Zeyad Safaa Younus ORCID logo EMAIL logo. Video steganography using knight tour algorithm and lsb method for encrypted data. *Journal of Intelligent Systems.*