Malicious PDF Detector

AI For Good

MarkovML

—

Team Members

Chirag Shameek Sahu
Gitika Jain
Umang Srivastava

Team Mentors

Akshaye Srivastava and Zad Basheer

Abstract

Due to their size and flexible capabilities, PDF documents have grown to be a popular target for targeted and broadcast phishing attempts. In this research, we offer a technique based on feature extraction from document metadata and structure for effectively identifying fraudulent documents. We demonstrate that we can provide a reliable malware classifier by meticulously extracting a broad variety of feature sets.

Introduction

Malicious malware is frequently bundled or inserted in papers that look official, such as government paperwork and bank statements. As a result, harmful documents are sometimes known as trojan documents because they include a malicious payload in an apparently desired document that acts as a vehicle for malware spread. As the usage of vulnerabilities in client programmes, such as document readers, has grown, so too has the popularity of using documents as a vehicle for exploitation. Moreover, social engineering makes it simpler to persuade individuals to open a document that contains malware.

Modern documents may be exceedingly difficult to identify malicious code due to their complexity and structure, which provides numerous opportunities for code to be mistaken for data. The situation is made worse by the fact that client applications, such as document readers and data containers, frequently consume foreign code and data, making them prime candidates for abuse. Recent research has demonstrated that highly socially engineered phishing assaults are regularly carried out by teams of highly skilled, persistent, and focused attackers with the aim of espionage.

In this project, we develop a model to extract the features from the PDF and classify it as malicious or benign. This requires the development of a dataset of PDF files that includes both malicious and benign files, as well as the use of appropriate feature extraction techniques and classification algorithms to build a robust model that can accurately classify new PDF files. The model is also evaluated thoroughly to ensure that it has a high degree of accuracy and does not produce many false positives or false negatives.

Implementation Procedure

Dataset Collection

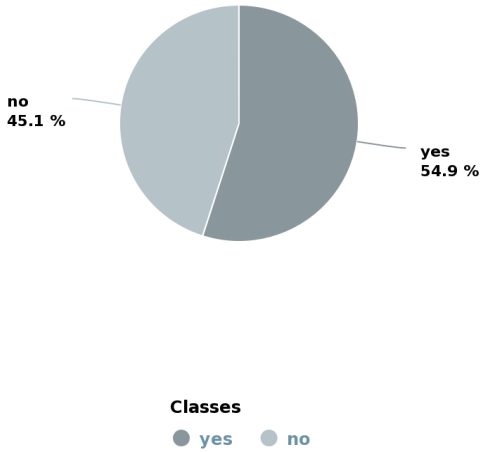
The Canadian Institute for Cybersecurity (CIC) provides a comprehensive dataset of malicious and benign PDFs that can be used for training and evaluating machine learning models for malware detection. The dataset contains over 10,000 PDF files, including both malicious and benign samples, which have been manually labeled and classified by cybersecurity experts. The dataset covers a wide range of malware families, including exploit kits, Trojans, and ransomware, as well as benign PDFs for comparison. This dataset is particularly useful for developing and testing machine learning models, as it provides a large and diverse set of samples that can be used to train models to accurately identify malicious PDFs while minimizing false positives. The CIC dataset is publicly available and can be used by researchers and security practitioners to improve their malware detection capabilities and enhance their overall cybersecurity posture.

In this research, we present a new evasive PDF dataset, Evasive-PDFMal2022 which consists of 10,026 records with 5558 malicious and 4468 benign records that tend to evade the common significant features found in each class. This makes them harder to detect by common learning algorithms. Once collected, we extracted 32 features from each, and after deduplicating the records, we wisely combined the two dataset records into one final file, which resulted in a more representative dataset of the PDF distribution. Moreover, we employed K-means, an unsupervised machine learning that clusters the resource data points into two groups by their similarity. The samples falling into the wrong cluster with the malicious label are taken as an evasive set of malicious records, with an intuition that the features of these samples were not so similar with the rest of the class so that they are not clustered with the majority of the same label samples. We applied the same logic for the benign records and finally combined the results with the new "Evasive-PDFMal2022".

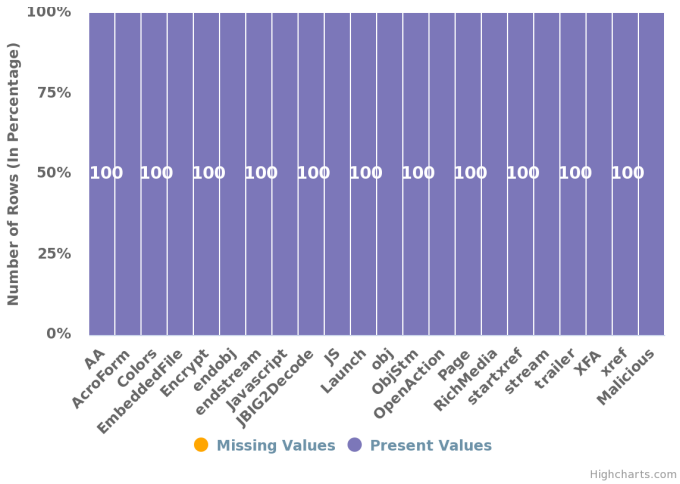
The dataset was uploaded on MarkovML application to analyse and validate the information that we have consolidated. The dataset we had obtained was not biased as it had almost equal number of malicious and benign PDFs. The data present in the dataset had no missing values in it.

Type	Rows
Mixed Or Categorical	10,026

Type of dataset and the number of data present in the dataset by MarkovML



Distribution of the classes of benign and malicious files present.



Details of missing values in the dataset (if present)

General features	Structural features
• PDF size • title characters • encryption • metadata size • page number • header • image number • text • object number • font objects • number of embedded files • average size of all the embedded media	• No. of keywords "streams" • No. of keywords "endstreams" • Average stream size • No. of Xref entries • No. of name obfuscations • Total number of filters used • No. of objects with nested filters • No. of stream objects (ObjStm) • No. of keywords "/JS", No. of keywords "/JavaScript" • No. of keywords "/URI", No. of keywords "/Action" • No. of keywords "/AA", No. of keywords "/OpenAction" • No. of keywords "/launch", No. of keywords "/submitForm" • No. of keywords "/Acroform", No. of keywords "/XFA" • No. of keywords "/JBig2Decode", No. of keywords "/Colors" • No. of keywords "/Richmedia", No. of keywords "/Trailer" • No. of keywords "/Xref", No. of keywords "/Startxref"

General Features

These features generally describe the PDF file, such as its size, whether it contains text or images, number of pages, and the title. There are 12 features from this category.

- Number of characters in the title: Legitimate PDF files usually have a proper and more meaningful titles.
- Metadata size: Metadata is the section where information about the PDF file is provided, which can be exploited for embedding hidden contents.
- Document Encryption: This feature shows whether the PDF document is password protected or not.
- Number of pages: Malicious PDF files tend to have fewer pages (most of them have one blank page) as they are not concerned about content presentation.
- Presence of text inside the PDF: As content presentation is not the objective of malware PDF files, they may include less text in their files.
- Size of the whole document: The malicious PDF size usually tends to vary from the benign due to its variation in page size and content.
- Number of embedded files inside the document: PDFs are capable of attaching/embedding different types of files within themselves that might be used for exploitation, including other PDF files, doc files, images, etc.

- Average size of all the embedded media: Embedded files in the PDF may be of various sizes depending on what they contain. The average size might lead to an insight into the content of the embedded files.
- Number of total objects inside the PDF: As PDFs are made of objects, the number of objects combined with the rest of the features can represent the PDF in general.
- Number of font objects: Font objects indicate the types of fonts used for the PDF text.
- Presence of a valid PDF Header: As PDF header obfuscation is common for evading anti-virus scans, malicious PDF files tend to modify the header format.
- Number of images in the document: PDF files may contain one or any number of images.

Structural Features

These features describe the PDF file in terms of the structure, which requires a deeper parsing and provide an insight into the overall skeleton of the PDF. We propose a set of 25 features related to the PDF structure.

- Number of indirect objects: This might be some indication of an obfuscation attempt.
- Number of obfuscations: PDFs support many types of obfuscations such as string obfuscations of hex, octal etc. which are generally applied for evasion attempts.
- Number of streams: This shows the number of sequences of binary data in the PDF.
- Number of of endstreams: Keywords that denote the end of the streams.
- Average Stream size: Size of the stream as the malicious code may be hidden inside streams.
- Number of stream objects (ObjStm): Streams that contain other objects.
- Number of Javascript keywords: This denotes the number of objects containing Javascript code, which is the most commonly exploited feature as evident.
- Number of JS keywords: Number of objects containing Javascript code.
- Number of Launch keywords: Launch is a keyword that can be used to execute a command or program.
- Number of URI keywords: Indicates a presence of URL to which the PDF file attempts to connect to.
- Number of Action keywords: Specifies a specific action upon an event.
- Number of AA keywords: Specifies a specific action upon an event.
- Number of OpenAction keywords: Specifies a specific action upon opening the PDF file. This feature combined with Javascript has been observed in the majority of typical malicious PDF files.
- Number of SubmitForm tags: Indicates the PDF button that collects form information and sends them to specified destinations.

- Number of Acroform tags: Acrobat forms are PDF files containing form fields that support scripting technologies that can be misused for attackers.
- Total number of filters used: There are various types of compression filters applied on some PDF objects, which also might be exploited by attackers.
- Presence of JBig2Decode filter: JBig2Decode is a common filter to encode malicious content.
- Number of objects with nested filters: Nested filters can be an indication of evasion, as they make the decoding process more difficult.
- XFA: XFAs are XML Form Architecture included in certain PDF 40 files that support scripting technologies that can be misused for attackers.
- Colors: Different colors used in the PDF.
- Trailer: Number of trailers inside the PDF.
- Xref: Number of Xref tables.
- Startxref: Number of keywords with ”startxref” which denotes where the Xref table is started.
- Xref enteries: The number of entries in the PDF Xref tables as malformed Xref tables are another common observation in malicious PDF files.
- RichMedia: Number of RichMedia keywords which denotes the number of embedded media and flash files.

Feature Engineering

Feature engineering is the process of selecting and extracting relevant features from raw data that can be used to train machine learning models. The quality and quantity of features used in a machine learning model can significantly impact its performance and accuracy. Feature engineering involves identifying the most important features that can differentiate between different classes of data, such as malicious and benign PDFs, and then extracting those features from the data. This process can involve a variety of techniques, including statistical analysis, data preprocessing, and domain knowledge. Feature engineering is an iterative process that often requires multiple rounds of experimentation and refinement to identify the most effective set of features. Good feature engineering can greatly enhance the performance of machine learning models and can lead to more accurate and effective malware detection systems.

After a careful evaluation, we noticed the the physical features were not so important for classifying the dataset as malicious. We dropped some features and kept those with a significant output regarding malicious and benign PDFs.

Classification Models

Calculating accuracy, precision, recall, and the confusion matrix of a classification model are crucial steps in detecting malicious PDF files. These metrics help to evaluate the performance of the machine learning models used in the classification process. Accuracy measures how often the model correctly identifies a file as malicious or benign, while precision measures the proportion of correctly classified malicious files. Recall, on the other hand, measures the proportion of actual malicious files that are correctly identified by the model.

The confusion matrix provides detailed information about the number of true positives, true negatives, false positives, and false negatives in the classification process, enabling a deeper understanding of the model's performance.

In the case of malicious PDFs, false positives can be particularly problematic. This is because they can lead to legitimate PDFs being flagged as malicious and potentially blocked or deleted, causing inconvenience for users. On the other hand, false negatives, where a malicious PDF is incorrectly classified as benign, can be equally problematic as they allow malicious files to go undetected and cause harm to users. Therefore, it is crucial to balance both false positives and false negatives and optimize the classifier's performance by calculating metrics such as accuracy, precision, recall, and the confusion matrix.

By calculating these metrics, the effectiveness of the classification process can be assessed, and improvements can be made to enhance the model's accuracy and reliability. Overall, these measures are essential for ensuring the efficient and effective detection of malicious PDF files, protecting users from potential security threats.

Naive Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

```
Gaussian Naive Bayes model Accuracy (in %): 98.78252168112076

Confusion Matrix:
 [[2668   22]
 [  51 3255]]

              precision    recall  f1-score      support
no             0.981243  0.991822  0.986504  2690.000000
yes            0.993287  0.984574  0.988911  3306.000000
accuracy       0.987825  0.987825  0.987825     0.987825
macro avg      0.987265  0.988198  0.987707  5996.000000
weighted avg   0.987883  0.987825  0.987831  5996.000000
```

Values obtained from the classifier.

Decision Trees Classifier

Decision tree builds classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

```
Decision Tree model Accuracy (in %):  99.69979986657772

Confusion Matrix:
 [[2730    8]
 [  10 3248]]

              precision    recall  f1-score      support
no             0.996350  0.997078  0.996714  2738.000000
yes            0.997543  0.996931  0.997237  3258.000000
accuracy       0.996998  0.996998  0.996998     0.996998
macro avg      0.996947  0.997004  0.996975  5996.000000
weighted avg   0.996998  0.996998  0.996998  5996.000000
```

Values obtained from the classifier.

Values obtained from MarkovML evaluations.

Random Forest Classifier

It is a supervised learning algorithm that uses ensemble learning method for classification. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

```
Random Forest model Accuracy (in %):  99.93328885923948

Confusion Matrix:
 [[2719    0]
 [   4 3273]]

              precision    recall  f1-score      support
no             0.998531  1.000000  0.999265  2719.000000
yes            1.000000  0.998779  0.999389  3277.000000
accuracy       0.999333  0.999333  0.999333     0.999333
macro avg      0.999266  0.999390  0.999327  5996.000000
weighted avg   0.999334  0.999333  0.999333  5996.000000
```

Values obtained from the classifier.

Values obtained from MarkovML evaluations.

Feature Extraction

Feature extraction is a critical component of building a machine learning model to classify PDFs as either malicious or benign. There are several features that can be extracted from PDFs that can provide important information for classification purposes. These features include metadata such as author, title, and creation date, as well as structural features such as the number of pages and the size of the file. Other relevant features include the presence of JavaScript, embedded files, and embedded URLs. Additionally, text-based features, such as the frequency of certain words or phrases, can provide important information for identifying malicious PDFs. Feature extraction from PDFs is a complex process that requires specialized tools and techniques, including parsing the PDFs and analyzing their content. By extracting the necessary features from PDFs, machine learning models can be trained to accurately identify potential indicators of malicious behavior and provide effective malware detection.

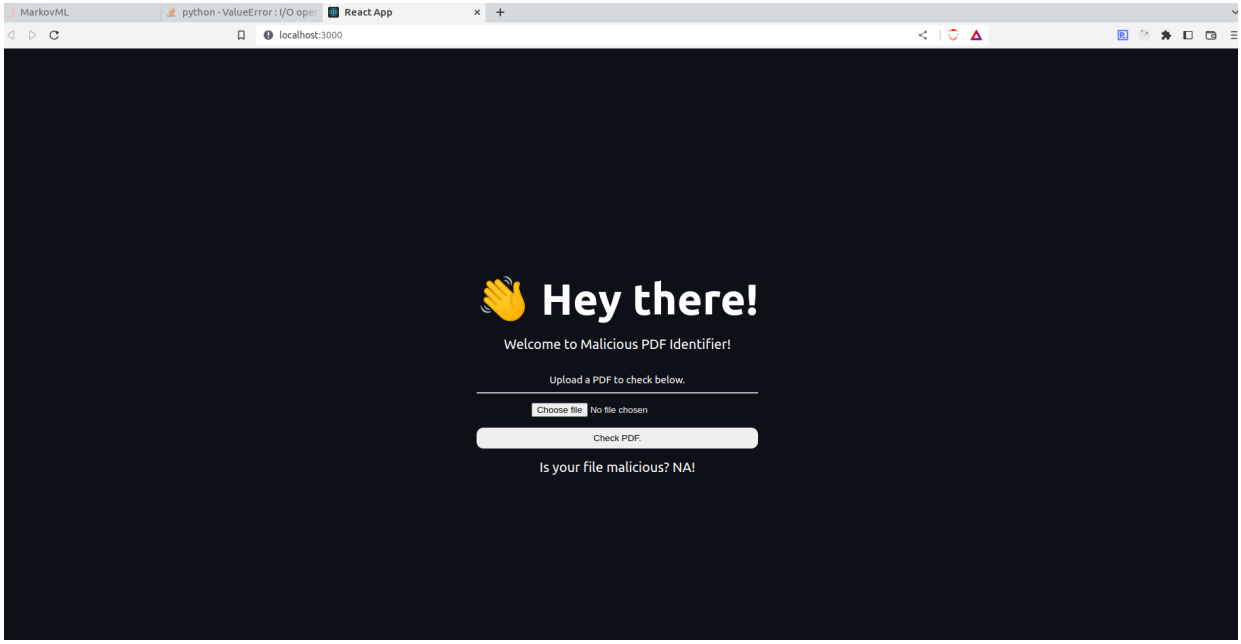
Deployment

A user-friendly web application has been developed that can classify PDFs as either malicious or benign. This web app allows users to upload PDF files for analysis, and then provides an immediate result indicating whether the file is malicious or not. The app uses a machine learning model that has been trained on a large dataset of labeled PDFs, and is capable of accurately identifying potential indicators of malicious behavior. The interface of the web app is designed to be user-friendly and intuitive, allowing even non-technical users to easily upload files for analysis. This web app can be a valuable tool for individuals and organizations looking to enhance their cybersecurity posture and protect against potential threats posed by malicious PDF files.

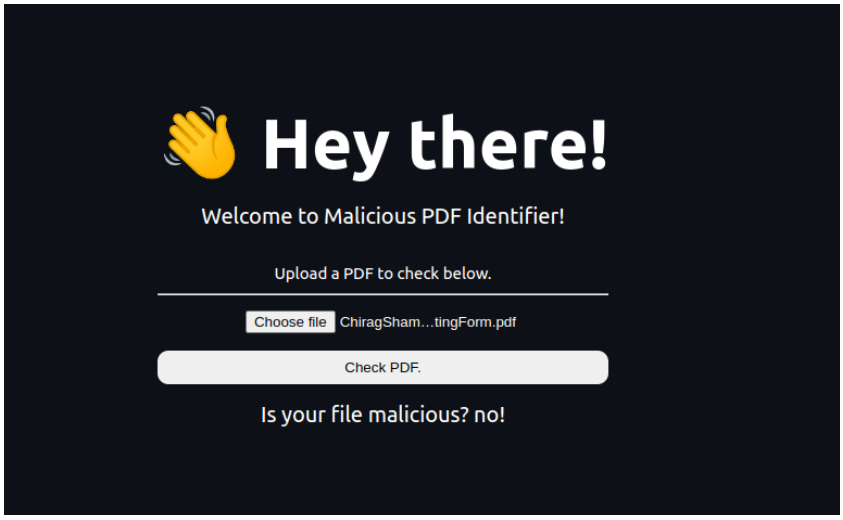
The web application's frontend is built using the React.js library, allowing users to upload a PDF file to be checked for malicious content. Once the user submits the file, it is sent to the backend, where all the necessary features of the file are extracted. These extracted features are then passed to the machine learning classification models that were previously discussed. In the event that one of the classification models incorrectly classifies the file, the application takes the maximum of the two classification outcomes to ensure greater accuracy. Finally, the results of the classification process are displayed on the screen, providing the user with information about whether the PDF file is malicious or benign. This process allows for quick and efficient analysis of PDF files, helping to protect users from potential security threats.

Pickle is a useful Python tool that allows you to save your ML models, to minimise lengthy re-training and allow you to share, commit, and re-load pre-trained machine learning models. In order to avoid the need for re-training the machine learning model every time a PDF is tested, a pickle file has been set up. This file allows for the quick and easy loading of

the pre-trained model, reducing the response time of the application significantly. By using a pickle file, the trained model can be stored and loaded quickly, without the need for time-consuming re-training, which would be necessary otherwise. As a result, the application is able to process PDF files much more quickly and efficiently, providing users with faster results and a smoother user experience overall.



Malicious PDF Detector Application



Testing a benign file.

Future Goals

To make this very easily accessible and useful, we plan on to develop an extension that will classify the PDF before downloading it. Also, the extension can have the option to upload a file to classify it.

In the future, the PDF classifier could be expanded to classify other types of files, such as Word documents, Excel spreadsheets, and image files. This would involve developing a more general-purpose classifier that can analyze the content and metadata of different file formats and identify potential indicators of malicious behavior. To achieve this, the classifier may need to be trained on a larger and more diverse dataset, and new features may need to be extracted and analyzed. Additionally, the classifier could be integrated into a broader security framework that includes other threat detection techniques, such as network traffic analysis and endpoint protection. Ultimately, the goal of expanding the classifier to other file formats is to provide a more comprehensive and effective defense against malware and other types of cyber attacks.

Product Issues Feedback

The complete issues and feedback for the product are documented in the following link:
MarkovML Product Issues