IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023

DOI: 10.17148/IJARCCE.2023.121108

Detection of Malware in PDF and Office Documents using Ensemble learning

Mrs.Priyanka Patil[1], Mrs.Madhuri Gedam[2]

Department of Computer Engineering, Universal College of Engineering[1]

Department of Computer Engineering, Shree L.R.Tiwari College of Engineering, Mira Road, Mumbai, India[2]

Abstract: Malware threats targeting PDF and Word documents have become increasingly prevalent, posing significant risks to information security. The review covers signature-based detection, behavior-based analysis, machine-learning approaches, and hybrid models. By examining the strengths and limitations of each technique, this abstract highlights the current state of research and identifies potential avenues for future improvements in malware detection for PDF and Word documents. The current survey serves as a valuable resource for researchers, practitioners, and decision-makers seeking insights into combating malware threats in these widely used file formats.

Keywords: PDF files, Office Documents, malware detection, static analysis, dynamic analysis

I. INTRODUCTION

States, businesses, and private users all need to strengthen their security as a result of growing Internet usage to safeguard sensitive data. Additionally, over the past few years, cyber security has received increased focus. 91% of all firms experienced a cyber attack in 2013 as a result of an increase in these attacks since 2009. Malware is currently one of the main cyber risks due to the Internet's rapid expansion. Malware is any software that carries out malicious operations, such as data theft, espionage, etc. Malware is a term coined by Kaspersky Labs to describe "a kind of computer application intended to damage a computer in many ways by infecting that of a legitimate user. Millions of hosts are attacked as a result of anti-virus scanners' inability to protect against the growing variety of malware. In 2015, four million unique malware items were found, and 6 5, 63, 145 different hosts were attacked, according to Kaspersky Labs (2016). Additionally, since there are so many attacking tools available on the Internet these days, malware can now be made with less technical skill. No matter their level of competence, anyone can become an attacker because anti-detection tools are widely available and it is possible to buy malware on the black market. Recent polls indicate that a growing amount of attacks are being carried out automatically or by script kids.

Therefore, one of the most crucial cyber security jobs for both individual users and corporations is protecting computer systems against malware. Recent assessments indicate that a growing amount of attacks are automated or carried out by script kids.

Malware protection of computer systems is therefore one of the most important cyber security tasks for both individual users and corporations, as even a single attack can result in compromised data and considerable losses.

It is vital to use precise and speedy detection techniques due to significant losses and frequent attacks. Today's static and dynamic methods are ineffective for detection, especially when dealing with zero-day attacks. As a result, techniques based on machine learning can be used. Malicious documents are used by attackers to transmit malware because of the rise in ignorant consumers and the incapacity of contemporary antivirus technology to identify them. The attacker uses social engineering to persuade the recipients to open the document. First off, most consumers don't understand how dangerous software works and frequently click on links and documents they get. The majority of recipients have learnt through osmosis that executable files which are downloaded from suspicious sources may contain malware, except they do not view non-executable documents with the same amount of skepticism. Second, the majority of detection techniques for malicious software either rely on signature-based or heuristic-based techniques. The fact that signature-based detection has a high failure rate and offers no defense against recently disclosed malware and zero-day vulnerabilities is known to the security community. The authors of (Bazrafshan, 2013) describe heuristic-based detection, which uses features for identification including OpCode or Control Flow Graphs (CFGs), API calls, and machine learning models. However, a hacker can get around these security measures to avoid being identified because the bulk of machine learning models are trained on a specific collection of harmful software. Along with web pages and email, document formats like PDF and Office Open XML facilitate communication over the internet. These file types were developed to offer a universal platform independent of hardware configurations that could send files to any device.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

These file formats are widely used, which has drawn the attention of hackers who want to spread harmful malware. The addition of information like other papers, photographs, videos, or computer languages like JavaScript or Visual Basic is made possible by the variety of file types available. Attackers frequently employ these traits to distribute malware.

Despite the security community's greatest efforts, social engineering and email phishing are still widespread. It preys on the user's ignorance and gullibility, as noted in (Wash, 2018).Despite advances made in spam detection and malware filters, harmful email attachments are still prevalent, as noted in (A. Cohen, 2016). According to the authors of Sophos (2017), the Locky ransomware family is showing symptoms of a reappearance, and they discuss how it spreads via spam e-mail, malicious PDF files, and Microsoft Word documents with embedded macros. The analysis of malware detection techniques in connection to risky publications is presented in this article. We address an attack surface and the possibilities an attacker has while working with a PDF or an Office document. It offers a thorough review of the problems that malware detection is now facing.

II. RELATED WORKS

Malware is any harmful program used to obstruct computer operations, collect private data, or gain access to other people's computers. Software that harms users unintentionally because of a flaw is not considered malware because it lacks harmful intent and operates against the user's preferences. Both genuine (malicious) malware and accidentally destructive software are frequently referred to as "badware." These are designed to enter computer systems and network resources, disrupt computer operations, and steal personal information without the system owner's consent. They so pose a risk to the internet's availability, the dependability of its hosts, and the privacy of its users. Malware has impacted many aspects of daily life, including e-governance (S. K. Talukder, 2014), social networks (Carbunar, June 2017), digital automation (S. K. Talukder, Digital land management system: A new initiative for bangladesh, April 2014), and mobile networks(M.Gedam, 2021).

In the previous year, 47% of the firms reported having encountered malware security incidents or network breaches, according to a poll (Corporation, 2019)carried out by Symantec in February 2019. The amount, diversity, and velocity of malware are all continuously increasing due to the evolving threat landscape, new harmful techniques, and threat fluidity. These are changing, getting more advanced, and adopting fresh strategies to attack computers and mobile devices. Each day, McAfee (Labs, 2019) catalogues over one lac new malware samples, which works out to sixty-nine new threats every minute or one threat per second. The number of straightforward yet sophisticated tools is expanding, along with the sophistication, tenacity, and ignorance of the new breed of cyber threats and attacks. Advanced malware is targeted, covert, stealthy, personalized, and zero-day as opposed to classic malware, which was broad, well-known, open, and only existed once. Advanced malware also differs from traditional malware in that it is zero-day and not as well known or as accessible. While some researchers(B. Anderson, 2012) attempted to bridge the static/dynamic gap, others studied the comparison of static, dynamic, and hybrid analysis for malware detection (A. Damodaran, 2017).

Numerous studies employ static analysis for malware detection, employing exact decompilation(E. Schulte, 2018) similarity testing framework (Zhou, 2015), based on register contents (M. Ghiasi, 2015), using two-dimensional binary program features (Berlin, 2015), employing subroutine based detection (J. Sexton, 2016), employing statistics of assembly instructions (P. Khodamoradi, 2015), employing file relation graphs (L. Chen, 2015), de-anonymizing programmers via code stylometry (A. Caliskan-Islam, 2015), and employing wavelet-based analysis (B. Gu, 2015), opcode disassembler analysis and comparison (M. Nar, 2019).

Dynamic analysis is used in studies that synthesize the semantics of obfuscated code, multi-hypothesis testing (Perdisci, 2016), quantitative data flow graph metrics (T. Wüchner, 2015), use a simplified data-dependent API call graph (A. A. E. Elhadi, 2013), downloader graph analytics (B. J. Kwon, 2015), access behaviour (A. Mohaisen, 2015)(W. Mao, 2015), APIs in initial behavior (Omote, 2015), and log-based crowd sourcing analysis (M.Gedam, 2019).

With the help of machine learning, numerous studies have examined its dynamic and static features, concept drift, predicting signatures, hybrid frameworks, malware metadata, and reverse engineering of big datasets of binaries for the detection and analysis of malware. (M. Polino, 2015).

III. PDF MALWARE

The Adobe Software-maintained PDF file format, which was developed in 1993 and has been around for thirteen years, is one of the most widely used methods for spreading malware, claims. Making PDF files adaptable so that sending text, photos, and videos became independent of hardware as well as software was one of the main goals of establishing this format. To provide the aforementioned capability, Adobe's architects added programmable features. 2008 saw the open

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023

DOI: 10.17148/IJARCCE.2023.121108

standardization of the document format under the name ISO320001:2008. Software developers could now freely include the ISO (2008) standard into their creations after this change. As was already demonstrated, malicious document attacks are flexible and make full advantage of the freedom offered by the document structure. Usually, a structural attack (embedded objects) and an interactive feature are used to target the reader software. (Java Scriptor Action Script).

PDF Document Structure

A typical PDF file has the following structure:

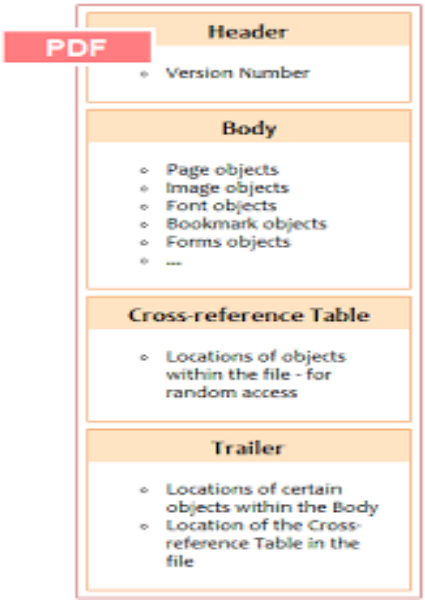

Fig 1: Structure of a PDF Document

Detection and Analysis of PDF Malware

Behavior-based and signature-based malware detection techniques fall into these two groups. Prior to using either static or dynamic malware analysis, it is essential to understand their foundations.

Static Analysis:
Static analysis is done "statically," or without running the file, as the name suggests. Dynamic analysis, on the other hand, examines the file as it is being used, such as in a virtual machine.
Static analysis might be compared to "reading" the malware's source code in an effort to identify the characteristics of the file's behavior.

They focus on document structure and metadata. Without ever running the code that a file contains, the objective is to ascertain whether it is hazardous. The security community has developed specific techniques for the examination of PDFs in order to do this.

Several strategies can be used in static analysis:
1. File format inspection: A file's metadata may be useful. For instance, Windows PE (portable executable) files can provide a lot of details about the build process, functions that are imported and exported, etc.
2. String Extraction: This is the procedure for deducing information about how malware operates from program output (such as status or error messages).
3. Finger printing: Computing cryptographic hashes and identifying environmental artifacts, such as hard-coded usernames, file names, and registry strings, are required.
4. AV scanning: Most anti-virus scanners ought to be able to recognize the file in question if it contains well-known malware. It can seem inconsequential, but A V vendors or sandboxes sometimes use this detection technique to "confirm" their findings.
5. Disassembly: To determine the software's logic and objectives, machine code is translated into assembly language. This method of static analysis is the most widely used and trustworthy one.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

Certain tools are frequently used in static analysis. They can offer details on malware security methods in addition to simple analyses. Discovering all potential behavioral scenarios is static analysis' key benefit. By studying the code itself, a researcher may examine all possible virus execution scenarios, not just the one that exists right now. Additionally, because the file is not run during this type of inspection, any negative effects on the system are prevented. Static analysis, on the other hand, requires a lot more time. Due to these factors, it is rarely utilized in dynamic real-world situations like anti-virus systems, but is frequently employed for research purposes, such as when creating signatures for zero-day malware.

Dynamic Analysis:
They concentrate on the document's structure and metadata. Without ever running the code embedded in it, the objective is to ascertain whether a file is malicious. To this end, the security community has created certain tools for the analysis of PDFs.

Searching for phrases that can imply harmful presence is one of the most popular and basic analysis strategies. Instead of decoding or decompressing the PDF document, the analyst typically only needs to look at the strings already present in it. The analysts are searching for phrases that are indicative of maliciousness, such as JavaScript, OpenAction, GoTo, URI, and Rich Media.

If no keywords are found, the analyst can state with certainty that the file being examined is not harmful. In contrast to static analysis, execution-time behavior of the file is observed, and its properties and intentions are inferred from that information. Usually, a sandbox-like virtual environment is used to run the file. The finding of all behavioral characteristics, such as opened files, produced mutexes, etc., is made possible by this type of research. Additionally, it develops much more quickly than static analysis. On the other hand, the static analysis just shows the behavioral situation relevant to the existing system attributes.

A general approach to detect and analyze PDF malware can be described as below:

1. Use an Antivirus Scanner: Start by scanning the PDF file with a reliable antivirus scanner. Many antivirus solutions have the capability to detect and identify known malware signatures in PDF files. If the scanner flags the file as malicious, follow the recommended actions provided by the antivirus software.

2. Analyze Metadata: Analyze the PDF file's metadata. The author, creation date, program, and possibly any embedded URLs or JavaScript code can all be learned via metadata. Keep an eye out for any unusual or suspicious facts that might point to nefarious motives.

3. PDF Structure Analysis: PDF files have a specific structure defined by the PDF specification. Analyzing the structure can help identify any anomalies or malicious elements within the file. Tools like PDFid or peepdf can assist in this process. Look for unexpected or obfuscated JavaScript, embedded objects, or hidden content layers.

4. PDF Parsing and Rendering: Use a reliable PDF parsing and rendering library or tool to process the PDF file. This step helps to identify any malicious behavior or unexpected actions triggered when the file is opened or rendered. Pay close attention to JavaScript actions, embedded URLs, or any attempts to exploit vulnerabilities in PDF readers.

5. Behavioral Analysis: Observe the behavior of the PDF file during execution. Note any unusual activities such as attempts to download additional files, create new processes, modify system settings, or interact with the user in unexpected ways. Behavioral analysis helps understand the intentions and capabilities of the malware.

6. Code Analysis: If the PDF file contains JavaScript or other executable code, perform a detailed analysis of the code. Look for obfuscation techniques, function calls to suspicious APIs, attempts to exploit vulnerabilities, or any other indicators of malicious intent. Tools like PDFStreamDumper or pdf-parser can assist in extracting and analyzing JavaScript code from PDF files.

7. Network Traffic Analysis: Capture and analyze network traffic generated by the PDF file. Use tools like Wireshark or tcpdump to monitor the network activity during the execution of the PDF. Look for connections to known malicious domains, communication with command-and-control servers, or attempts to download additional payloads.

8. Reporting and Mitigation: Document your findings, including any indicators of compromise (IOCs) and behavioral patterns exhibited by the PDF malware. Share the information with relevant security teams or communities to help prevent future infections. Based on your analysis, implement appropriate mitigation measures, such as updating antivirus signatures, blocking malicious domains, or patching vulnerable software.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

It is important to note that analyzing and handling malware requires specialized knowledge and expertise. If you are not experienced in malware analysis, consider seeking assistance from cyber security professionals or organizations dedicated to malware research and analysis.

IV. MS OFFICE MALWARE

Microsoft Office programs like Word, Excel, and PowerPoint have enjoyed a cult following ever since they were released. These programs are widely used by both business and home computer users, making them good targets for attacks similar to those targeting PDFs. The software suite has a lengthy history of being misused by dishonest authors dating back to its introduction in 1990.

Malware can attack victims in current Office document apps using a variety of techniques. Here are three typical attack strategies:

1. Macro-Based Attacks: Macro-based attacks involve malicious macros embedded within Office documents, such as Word documents (with .doc or .docx extensions) or Excel spreadsheets. Macros are scripts or code snippets that automate tasks within the Office application. Attackers often trick users into enabling macros, which then execute malicious code that can download and install malware on the victim's system. These macros may be disguised as legitimate functions or content, making them appear harmless to unsuspecting users.

2. Embedded Objects and Links: Office documents can contain embedded objects, such as images or media files, which may be used to deliver malware. Attackers can embed malicious executables or exploit vulnerabilities in these objects to execute code and compromise the victim's system. Similarly, Office documents can include hyperlinks that lead to malicious websites or download malicious files when clicked. This method is commonly used in phishing campaigns, where users are tricked into clicking on seemingly legitimate links.

3. Exploiting Vulnerabilities: Attackers can exploit vulnerabilities in Office applications themselves to deliver malware. These vulnerabilities may exist in the underlying software code or its components, such as the parsing and rendering engines. By crafting malicious Office documents specifically designed to exploit these vulnerabilities, attackers can execute code on the victim's system and gain control or install malware. It's crucial to keep Office applications and associated plugins up to date with the latest security patches to mitigate the risk of such attacks.

To defend against these methods, users and organizations should adopt the following practices:

- Keep software and applications updated with the latest security patches.
- Enable security features provided by Office applications, such as macro security settings or protected view mode.
- Exercise caution when opening email attachments or downloading Office documents from untrusted or unknown sources.
- Disable macros by default and only enable them for trusted documents from reliable sources.
- Be vigilant for social engineering techniques, such as emails that urge immediate action or requests for sensitive information.
- Employ security solutions, such as antivirus software, that can detect and block known malware and malicious files.
- Educate users about the risks and best practices for handling Office documents, including the importance of not enabling macros or clicking on suspicious links.

By adopting these proactive measures, users can significantly reduce the risk of falling victim to malware attacks through Office documents.

Structure of Office Documents:

Microsoft Office applications have transitioned from the Compound File Binary Format to the Office Open XML (OOXML) format as the default file format for their documents. Here's an overview of both formats:

1. Compound File Binary Format (doc/ppt/xls): The Compound File Binary Format, also known as Binary Interchange File Format (BIFF), was the original file format used by Microsoft Office applications.
This format was used for .doc (Word documents), .ppt (PowerPoint presentations), and .xls (Excel spreadsheets) files created in older versions of Microsoft Office (prior to Office 2007).
The file structure of Compound File Binary Format involved a binary file format with a hierarchical structure. It stored data in a binary format using a series of records and streams.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023

DOI: 10.17148/IJARCCE.2023.121108

2. Office Open XML (OOXML):

The Office Open XML (OOXML) format is an XML-based file format introduced by Microsoft with the release of Office 2007.

OOXML is an open standard that is based on XML, making it more accessible and interoperable with other software applications.

The file extensions for Office Open XML format are .docx (Word documents), .pptx (PowerPoint presentations), and .xlsx (Excel spreadsheets), among others.

OOXML format files are essentially ZIP archives that contain a collection of XML files and other resources, such as images, embedded objects, and stylesheets.

The XML files within the OOXML package define the structure, content, and formatting of the document, making it more modular and extensible.

The transition to the OOXML format brought several advantages, including improved file size, enhanced data recovery, better compatibility across platforms, and the ability to work with different programming languages for manipulating Office documents.

While the OOXML format is now the default format for Microsoft Office documents, the older Compound File Binary Format is still supported by the Microsoft Office suite to ensure compatibility with legacy documents and applications.

Detection and analysis of Office malware involve several steps to identify and understand malicious behavior within Office documents. Here is a general process for detecting and analyzing Office malware:

1. Antivirus Scanning: Use antivirus software or security solutions that include signature-based detection to scan Office documents for known malware signatures. This step helps quickly identify and block known malicious files.
2. Behavior Analysis: Analyze the behavior of the Office document when opened or executed. Observe any suspicious activities such as attempts to download additional files, network connections, or unauthorized access to system resources. Tools like sandbox environments can be used to isolate and monitor the document's behavior without impacting the host system.
3. Macro Analysis: If the Office document contains macros, carefully analyze the macro code for any suspicious or malicious commands. Look for obfuscated code, calls to external resources, attempts to download or execute files, or any other indicators of malicious intent. Consider using macro analysis tools or scripting language parsers to assist in code analysis.
4. Embedded Objects Analysis: Examine any embedded objects within the Office document, such as images, media files, or linked content. These objects may contain malicious code or exploit vulnerabilities. Validate the integrity and safety of embedded objects and review their associated scripts or macros.
5. Vulnerability Assessment: Check for any known vulnerabilities within the Office application or its components that could be exploited by malware. Stay informed about security patches and updates provided by the software vendor to mitigate known vulnerabilities.
6. File Format Analysis: Understand the structure and specifications of the Office file format being analyzed. This knowledge helps identify anomalies or suspicious elements within the file, including unexpected data structures or modifications that could indicate malicious intent.
7. Reverse Engineering: For more advanced analysis, consider reverse engineering the Office malware. This process involves disassembling the malware to understand its inner workings, uncover hidden functionality, and identify potential indicators of compromise (IOCs).
8. Threat Intelligence and Collaboration: Leverage threat intelligence sources and collaborate with other security professionals or organizations to stay informed about emerging Office malware threats. Sharing information and insights can help in early detection and analysis of new and evolving malware variants.
9. Incident Response: Develop an incident response plan to handle Office malware incidents effectively. This plan should include steps for containment, eradication, and recovery in case of a successful malware infection.

It's important to note that the detection and analysis of Office malware require a combination of automated tools, manual analysis, and expertise in malware analysis. Regular updates of antivirus software, adherence to best security practices, and user awareness about potential risks associated with Office documents are also crucial in mitigating the risk of malware infections.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023

DOI: 10.17148/IJARCCE.2023.121108

V. MALWARE DETECTION PROCESS USING MACHINE LEARNING

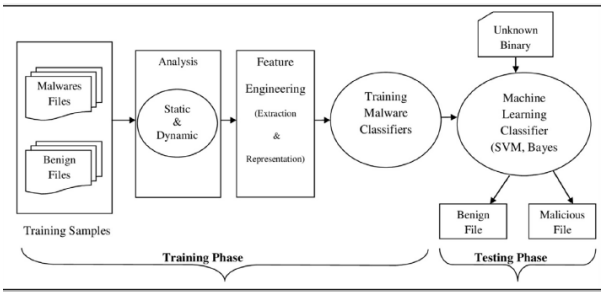

Fig 2: Framework of Malware Detection System using Machine Learning

The overall process of malware detection using machine learning involves several key steps. Here is a high-level overview of the process:

1. Data Collection: Gather a large dataset of known malware samples and benign files. The dataset should be diverse, representative of different malware families, and include various file types (e.g., executables, scripts).
2. Feature Extraction: Extract relevant features from the collected files. Features can include static attributes such as file size, file type, and metadata, as well as dynamic attributes like API calls, system calls, or network traffic generated during execution. The goal is to capture characteristics that distinguish malware from benign files.
3. Pre processing: Clean and pre process the extracted features to ensure data quality and compatibility. This step may involve normalizing, scaling, or transforming the features, handling missing values, and encoding categorical variables.
4. Feature Selection: Select the most relevant features from the pre processed dataset. Feature selection helps reduce the dimensionality of the data, improves model performance, and eliminates noise or irrelevant information. Techniques like correlation analysis, information gain, or regularization methods can be used for feature selection.
5. Training and Validation: Divide the pre-processed dataset into a training set and a validation set. The validation set is used to assess and improve the performance of the machine learning model after it has been trained using the training set. Decision trees, random forests, support vector machines (SVM), and deep learning models like convolutional neural networks (CNN) and recurrent neural networks (RNN) are examples of common methods used for malware identification.
6. Model Training: On the training dataset, train the chosen machine learning model. Based on the features given, the model learns the patterns and traits of malware. The model's parameters are changed during training to reduce the discrepancy between predicted and real labels.
7. Model Evaluation: Utilizing the validation dataset, assess how well the trained model performed. Metrics like accuracy, precision, recall, and F1 score can be used to evaluate how well the model can distinguish between malicious and benign files. Based on the particular use case and required level of security, it's critical to take into account the trade-off between false positives and false negatives.
8. Model Optimization: To improve the model's performance, tweak hyperparameters like learning rate, regularization strength, or tree depth. This process involves iterative experimentation to find the best configuration for the model.
9. Testing and Deployment: Once the model is optimized, it can be tested on an independent test dataset to assess its performance in real-world scenarios. The model can be implemented in the production environment to detect malware in real-time if it satisfies the specified performance criteria.
10. Monitoring and Updating: Continuously monitor the performance of the deployed model and collect feedback data to assess its effectiveness in detecting new or evolving malware threats. Regularly update the model with new data and retrain it to adapt to the changing threat landscape.

It's important to note that machine learning-based malware detection is not a standalone solution and should be complemented with other security measures, such as signature-based detection, behavior monitoring, and human analysis, to provide a comprehensive defense against malware.

VI. DISCUSSION & ANALYSIS

The Portable Document Format is a recognized tool for carrying out evil activities. A broader attack surface resulted from the format's introduction as an open standard in 2008.Targets are typically software and PDF readers.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

Malevolent actors can get beyond security measures and profit if they are aware of the execution pattern or strategy, as was presented using the example of PDFid for document static analysis. Similar problems, higher computing costs, and occasionally misleading findings beset dynamic detectors. Three types of academic research were covered: java script-based detectors, signature analysis and pattern matching, and structural and metadata features. Attacks involving metamorphosis and polymorphs may be susceptible to signature-based detections. As mentioned in (Maiorca D. C., 2013) , mimicry, reverse mimicry, and other polymorphic attacks play a substantial and common role in the majority of structural models. (Maiorca D. &., 2017) go into great depth about how these attacks affect PDF detectors. The difficulties with Javascript-based detectors are comparable.Adversarial samples can also trick machine learning-based detectors into missing data. These documents are made by altering legitimate documents, then using erroneous generalizations made by machine learning algorithms to cause a misclassification. These assaults not only target conventional categorization methods but also neural networks and deep learning. (Papernot, 2016).
Because any one of these components could be subpar and result in evasion or high false-negative rates, it is vital to choose the dependencies, the analysis method, and decision algorithm at the time of design.

Issues & Challenges

Previous research demonstrated that machine learning techniques were successfully used in the malware detection and antivirus sectors. There are, however, a few other difficulties as well that need more research.

Evaluation and implementation of the detection findings: Although malware can be found in a collection of unknown file sample files using techniques for data mining, such as classification/clustering methods, establishing the existence of potentially hazardous files is one of the difficult implementation problems. This is due to the fact that manual evaluation of these potentially hazardous files typically takes a lot of time and skill from subject matter experts.

Incremental Learning: A classifier can be trained to recognize recently circulated malware by using historical file sample collections that include both safe and hazardous samples. However, daily fresh malware samples are created, and malware strategies are always changing. Data-mining-based malware detection systems must take into consideration the most current file sample collection in order to account for the temporal trends of virus writing. The training sets must be regularly updated to accommodate fresh samples while maintaining the essential components of prior data gathering for the classifier(s) to continue to function effectively. Therefore, one of the problems with virus detection systems based on data mining is incremental learning.

Active learning: It is necessary to choose representative samples from huge unknown file sets in order to increase the detection accuracy. For instance, the recently published Trojan-Downloader and the associated trojans are collected from the clients and designated as unknown before being discovered. The linked trojans can be appropriately identified if we can find the Trojan-Downloader and tag it using the collected features and classification/clustering algorithms. Numerous studies have been conducted in recent years on active learning as a paradigm that may effectively handle the problem of data scarcity, optimize the learning gains from input from domain experts, and lower the cost of acquiring labeled instances for supervised learning. Active learning is rarely used in malware detection studies to select representative samples from a huge sample collection of files.

Malware prevalence prediction: In addition to spotting malware in a collection of unknown files, it's crucial to foresee how malware prevalence will change over time. There is, however, limited study on predicting the incidence of malware.

Adversarial Learning: When detecting malware, data mining techniques give malware attachers the opportunity to "mistrain" the classifiers (for example, by altering the distribution of the data or the weighting of the features). Thus, the challenge of how to create approaches that are reliable and secure in hostile environments arises.

VII CONCLUSIONS

In conclusion, malware detection in PDF and Word documents poses unique challenges due to the potential for embedded malicious elements and the ability to execute code within these file formats. Here are a few key points:

1. Signature-based detection: Antivirus software and security solutions often employ signature-based detection to identify known malware signatures in PDF and Word documents. Regular updates to antivirus databases are crucial to stay current with new malware variants.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

2. File structure analysis: Analyzing the structure of PDF and Word documents can reveal hidden or obfuscated elements that may contain malware. Understanding the file format specifications and using specialized tools can help identify suspicious or malicious components.

3. Embedded macros and scripts: Both PDF and Word documents support macros and scripting languages (such as VBA). Malicious macros and scripts can be used to execute code and launch attacks. Detection techniques may involve analyzing the code, examining metadata, or using behavior-based analysis to identify potentially malicious behavior.

4. Sandbox analysis: Running PDF and Word documents in isolated environments, such as sandboxes, can help observe their behavior without risking the host system's security. Sandbox analysis aids in identifying potential malware activities, such as network connections, file modifications, or attempts to exploit vulnerabilities.

5. Machine learning-based detection: Machine learning techniques can be used to detect malware in Word and PDF documents. By using large datasets of well-known benign and dangerous files as training data, machine learning algorithms can learn to recognize patterns and attributes that distinguish malware from real files.

6. User awareness and caution: Educating users about the risks associated with opening unsolicited or suspicious documents, especially those received via email or other untrusted sources, is crucial. Encouraging users to exercise caution and use trusted sources for file downloads can help prevent malware infections.

It's important to note that malware detection is an ongoing challenge, as malware authors continuously evolve their techniques to evade detection. Employing a multi-layered approach that combines signature-based detection, behavior analysis, machine learning, and user awareness is essential for effective malware detection in PDF and Word documents. Regular updates to detection mechanisms and security practices are necessary to stay ahead of emerging threats.

REFERENCES

[1]. A. A. E. Elhadi, M. A. (2013). Improving the detection of malware behavior using simplified data dependent api call graph. *International Journal of Security and Its Applications*, 29-42.
[2]. A. Caliskan-Islam, R. H. (2015). De-anonymizing programmers via code stylometry. *24th {USENIX} Security Symposium ({USENIX} Security 15*, (pp. 255-270).
[3]. A. Cohen, N. N. (2016). SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods. *Expert Syst. Appl.* , pp. 324-343.
[4]. A. Damodaran, F. D. (2017). "A comparison of static, dynamic, and hybrid analysis for malware detection". *Journal of Computer Virology and Hacking Techniques* , 1-12.
[5]. A. Mohaisen, O. A. (2015). Amal: High-fidelity, behavior-based automated malware analysis and classification. 251-266.
[6]. A. Raff, D. P. (2019). *System and methods for malware detection using log based crowdsourcing analysis.* US Patent.
[7]. B. Anderson, C. S. (2012). Improving malware classification: bridging the static/dynamic gap. *Proceedings of the 5th ACM workshop on Security and artificial intelligence*, (pp. 3-14).
[8]. B. Gu, Y. F. (2015). A new static detection method of malicious document based on wavelet package analysis. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, (pp. 333-336).
[9]. B. J. Kwon, J. M. (2015). The dropper effect: Insights into malware distribution with downloader graph analytics. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, (pp. 1118-1192).
[10]. Bazrafshan, Z. H. (2013). A survey on heuristic malware detection techniques. . *In Ikt 2013-5th conference on information and knowledge technology* (pp. 113–120). Shiraz, Iran: https:// ieeexplore.ieee.org/abstract/.
[11]. Berlin, J. S. (2015). Deep neural network-based malware detection using two dimensional binary program features. *10th International Conference on Malicious and Unwanted Software (MALWARE)* (pp. 11-20). IEEE.
[12]. Carbunar, S. T. (June 2017). "When friend becomes abuser: Evidence of friend abuse in Facebook,". *9th ACM Conference on Web Science.* New York: WebSci.
[13]. Corporation, S. (2019). *Internet security threat report.* Symantec.
[14]. E. Schulte, J. R. (2018). Evolving exact decompilation. Workshop on Binary Analysis Research (BAR).
[15]. J. Sexton, C. S. (2016). Subroutine-based detection of apt malware. *Journal of Computer Virology and Hacking Techniques* , 225-233.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023
DOI: 10.17148/IJARCCE.2023.121108

[16]. L. Chen, T. L. (2015). Intelligent malware detection based on file relation graphs. *Intelligent malware detection based on file relation graphs* (pp. 85-92). IEEE.
[17]. Labs, M. (2019). *Mcafee labs threats reports.* McAfee.
[18]. M. Ghiasi, A. S. (2015). *Dynamic vsa: a framework for malware detection based on register contents.* Engineering Applications of Artificial Intelligence.
[19]. M. Nar, A. G. (2019). Analysis and comparison of disassemblers for opcode based malware analysis. *4th International Conference on Computer Science and Engineering (UBMK)* (pp. 17-22). IEEE.
[20]. M. Polino, A. S. (2015). Jackdaw: Towards automatic reverse engineering of large datasets of binaries. *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 121-143). Springer.
[21]. Maiorca, D. &. (2017). *Digital investigation of pdf files: Unveiling traces of embedded malware.* IEEE Security & Privacy.
[22]. Maiorca, D. C. (2013). Looking at the bag is not enough to find the bomb: An evasion of struc- tural methods for malicious pdf files detection. .
[23]. Omote, N. K. (2015). Malware function classification using apis in initial behavior. *10th Asia Joint Conference on Information Security* (pp. 138-144). IEEE.
[24]. P. Khodamoradi, M. F. (2015). Heuristic metamorphic malwaredetection based on statistics of assembly instructions using classification algorithms. *18th CSI International Symposium on Computer Architecture and Digital Systems (CADS)* (pp. 1-6). IEEE.
[25]. Papernot, N. M. (2016). The limitations of deep learning in adversarial settings. In Security and privacy. (pp. 372-387). Saarbrucken, Germany: ieee european symposium.
[26]. Perdisci, P. V. (2016). Maxs: Scaling malware execution with sequential multihypothesis testing. *Proceedings of the 11th ACM on Asia Conference on ``````````````````````Computer and Communications Security*, (pp. 771-782).
[27]. S. K. Talukder, M. I. (2014). "Model for e-government in bangladesh:A unique id based approach," . *International Conference on Informatics, Electronics Vision (ICIEV)*, (pp. 1-6).
[28]. S. K. Talukder, M. I. (April 2014). Digital land management system: A new initiative for bangladesh. *International Conference on Electrical Engineering and Information Communication Technology*, (pp. 1-6).
[29]. S. Talukder, I. I. (2017). Attacks and defenses inmobile ip: Modeling with stochastic game petri net. *International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)* (pp. 18-23). IEEE.
[30]. T. Blazytko, M. C. (2017). "Syntia: Synthesizing the semantics of obfuscated code". *26th {USENIX} Security Symposium ({USENIX} Security 17*, (pp. 643-659).
[31]. T. Wüchner, M. O. (2015). Robust and effective malware detection through quantitative data flow graph metrics. *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 98-118). Springer.
[32]. Virustotal. (2018). *Submission statistics.*
[33]. W. Mao, Z. C. (2015). Probabilistic inference on integrity for access behavior based malware detection. *International Symposium on Recent Advances in Intrusion Detection* (pp. 155-176). Springer.
[34]. Wash, R. &. (2018). *In Proceedings of the 2018 chi conference on human factors in computing systems*, (p. 492). New York, NY, United States.
[35]. Zhou, J. U. (2015). Variant: a malware similarity testing framework. *10th International Conference on Malicious and Unwanted Software (MALWARE)*, (pp. 31-39). IEEE.
[36]. Madhuri N. Gedam and B. B. Meshram(2021) ,Database Private Security Jurisprudence: A case study using Oracle, IJDMS.
[37]. MN Gedam, BB Meshram(2019), Proposed Secure Content Modeling of Web Software Model,– IJETT.
[38]. M Gedam, B Meshram (2019)Vulnerabilities and attacks in SRS for object-oriented software development, WCES, USA.