International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, July 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

Implementation of a Web-Based Malware Analysis System With Random Forest Integration

1st Muhammad Fauzan, Cholid Mawardi[2], Eka Desy Asgawanti[3]
Politeknik Negeri Media Kreatif, Jakarta, Indonesia
E-mail: muhfauzan.contact@gmail.com[1] , cholid@polimedia.ac.id[2] , ekadesy@polimedia.ac.id[3]

Abstract—*With the rapid advancement of digital technology, the threat of malware has become increasingly prevalent and sophisticated, posing significant risks to both individuals and organizations. Despite the growing need for robust protection, many existing malware analysis tools are overly complex, often requiring advanced technical knowledge, which makes them less accessible to general users. To address this gap, this study proposes the development of a web-based malware analysis system that is both powerful and user-friendly. The system is built using the Streamlit framework, which allows for the creation of interactive and responsive web applications with minimal overhead. At its core, the system integrates a Machine Learning model based on the Random Forest algorithm, chosen for its high accuracy and robustness in classification tasks, particularly in distinguishing between benign and malicious files. For in-depth file analysis, the system connects to the MetaDefender API, which scans submitted files using multiple antivirus engines and provides comprehensive threat intelligence data. To further enhance accessibility, especially for users without a technical background, the GPT API is integrated to automatically generate simplified interpretations of complex scan results, explaining the findings in natural language. The system displays results using graphical visualizations, making it easier for users to comprehend potential threats without needing to interpret raw data or technical jargon. This visual and interactive approach supports real-time decision-making and improves user experience. The methodology employed in this research is quantitative, focusing on the evaluation of the system's performance and the effectiveness of the Random Forest model in accurately classifying malware. Key performance metrics such as accuracy, precision, recall, and F1-score are used in the analysis. Overall, this system offers several competitive advantages: enhanced accessibility, improved ease of use, and simplified result interpretation compared to traditional malware analysis tools. The research contributes to the broader field of cybersecurity by providing a more practical and user-friendly solution for malware detection, thereby helping to raise awareness and improve protective measures against digital threats.*

Keywords: *Malware, Machine Learning, Random Forest, Cybersecurity, Streamlit, MetaDefender API, GPT API*

I. INTRODUCTION

In an increasingly digital age, cybersecurity threats have become a significant concern for individuals, organisations, and governments [1]. Malware is one of the most common and dangerous threats in the cyber world, with widespread impacts on data security and system infrastructure [2]. According to the 2023 report by the National Cyber and Cryptography Agency (BSSN), the number of malware attack incidents in Indonesia has increased significantly, reaching 4.2 million in a single year. The growing complexity and difficulty in detecting malware attacks necessitate systems capable of providing fast, accurate, and easily accessible analysis for various user groups.

The main challenge in malware analysis is that many of the detection systems currently available are designed for cybersecurity professionals, making them difficult for general users to use. These systems require in-depth technical understanding and complex analysis procedures [3]. Therefore, a solution is needed that not only detects malware with high accuracy but also provides analysis results in a format that is easier to understand. In this study, a web-based malware analysis system was developed that integrates the Random Forest Machine Learning algorithm to improve detection accuracy and the GPT API to simplify the interpretation of analysis results [4].

The main objective of this study is to develop a web-based malware analysis system that utilises Machine Learning technology to improve the effectiveness of malware detection. By using the Random Forest algorithm, this system can identify malware patterns more accurately based on the collected dataset. Additionally, this research aims to develop a web-based user interface using the Streamlit framework, enabling broader accessibility for non-technical users. With the integration of the MetaDefender API, this system is also expected to perform more in-depth and comprehensive malware analysis [5].

This research offers several significant benefits, both for the academic world and the cybersecurity industry. For academics, this research contributes to the development of new methods in Machine Learning-based malware analysis, which can be used as a reference for further study. For industry and the general public, this system can be an easily accessible tool to detect and analyse malware without requiring in-depth technical expertise. Thus, users can more quickly identify and address malware threats before they cause greater damage [6].

Overall, this research is expected to increase public awareness and understanding of the importance of cybersecurity. Implementing this system is expected to be a practical solution that helps users analyse malware more easily and effectively. With a high level of accuracy and a user-friendly interface, this system can be a first step in improving protection against evolving cyber threats [7].

II. RESEARCH METHODS

Research on Machine Learning-based malware detection has grown rapidly in recent years. Developing Dynamic Analysis-based malware analysis methods that enable real-time monitoring of malicious activities. This study shows that behaviour-based analysis methods are more effective than static analysis in detecting more complex malware. However, this approach still has limitations in interpreting results for non-technical users [8].

Another study by Akbar & Sutabri (2024) implemented AI technology in the detection and prevention of malware attacks on corporate computer networks. This study used Machine Learning algorithms to classify files based on their threat potential [9]. Although this research successfully improved attack detection with a high degree of accuracy, the developed system is still limited to corporate environments and is not designed for general users [10].

Sitorus, Sukarno, & Mandala (2021) used Support Vector Machine (SVM) and Random Forest methods for Android malware detection. The research results showed that combining Machine Learning methods can improve malware detection accuracy with lower error rates compared to traditional methods. However, this study focused solely on the Android platform and did not cover broader web-based approaches [11].

Compared to previous research, the system developed in this study offers a more comprehensive solution by integrating the MetaDefender API for deeper analysis and the GPT API to simplify analysis results. As a result, non-technical users can understand malware detection results without requiring a deep background in cybersecurity [12].

Additionally, this system was developed using the Streamlit framework, which enables more intuitive user interaction. This addresses the issue in previous research that still required a more user-friendly interface. With this approach, the system can be accessed by various groups, including academics, IT professionals, and even lay users who wish to enhance their cybersecurity [13].

The presented system is based on the previously used Random Forest algorithm, which achieved only 98.79% accuracy, but with optimised settings in the analysis process, it achieves a malware detection accuracy rate of no less than the minimum threshold of 95%, meaning it can reach 100%. This represents a significant leap in performance compared to previous research, where such high accuracy in web-based and cross-platform systems has not been explicitly reported [14].

With the integration of this latest method, this research contributes to improving the effectiveness of malware detection and provides a more practical solution in cybersecurity. Implementing a web-based system using Machine Learning technology with Random Forest is expected to support more accurate and efficient detection of malware threats, in line with the research objectives that have been designed [15].

This study uses a quantitative approach with exploratory methods to analyse the effectiveness of algorithms. Random Forest, as a machine learning integration, is then used in malware detection. The following is the workflow of the proposed web-based malware analysis system:

1. File Upload: Users upload files that need analysis to the system.
2. In-depth Analysis: The MetaDefender API will analyse the file label and extract the file's features, including parts of the hash, file signature, and file behaviour when running.
3. Processing by Machine Learning Model: The extracted features will be processed by a machine learning model with a Random Forest algorithm trained on a large malware dataset. This model will predict whether the file is malware or not.
4. Result Interpretation: The GPT API can translate the results into simple human language. The results of the Machine Learning prediction statistically predict the file label as Ransomware, malware, or a file that is entirely safe to use.
5. Result Presentation: The interpretation results will be displayed on the Streamlit web interface.

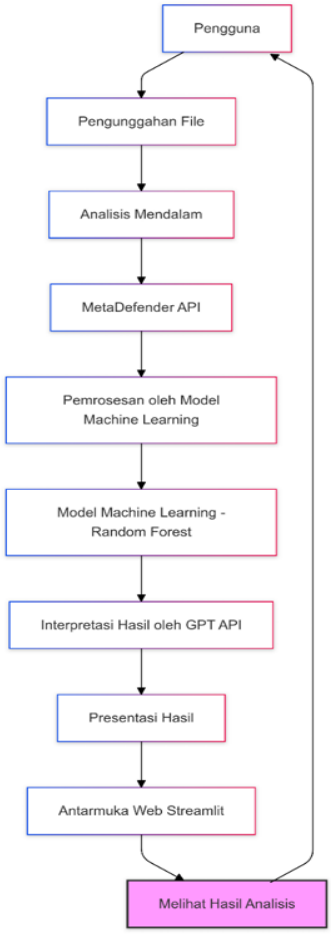


Figure 1. Research Workflow

Based on Figure 1, the research process can be used to formulate hypotheses to support this research as follows:

1. H1: A web-based malware analysis system that integrates Random Forest algorithms, MetaDefender API, and GPT API can accurately detect malware and provide comprehensive analysis results to users.
2. H2: The results of the web application content use an interactive and responsive Streamlit.

3. H3: Adding the MetaDefender API increases user confidence in malware analysis results.

4. H4: The GPT API simplifies the language of the analysis results, converting them into language that is easily understood by non-technical people.

The objects of this research are the LockBit 3.0 and Brain Cipher ransomware types. LockBit 3.0 is the latest ransomware in the LockBit family. It shares the same characteristics as LockBit in general, such as the ability to encrypt data quickly and threaten victims with ransom payments via crypto wallets [16][17]. However, this ransomware has further specifications in data encryption and threatening victims. Unlike LockBit, Brain Cipher is a more advanced ransomware than its predecessors, with newer encryption techniques and deeper obfuscation methods [18]. This research was conducted in the virtual laboratory of the malware analysis system owned by the IT and Digital Marketing consulting company PT Bintang Internasional Nusadigital. The location was chosen to explain the data tracing techniques with a deeper analysis of both ransomware. The malware analysis system is web-based and uses the Random Forest algorithm, MetaDefender API, and GPT API.

In this study, the data collection method was an experimental technique with a static malware analysis approach. The experiment was conducted by collecting samples of LockBit 3.0 and Brain Cipher ransomware through simulation and analysis in an isolated environment in the virtual laboratory. The experimental process involved downloading, extracting, and identifying malware files, as well as observing patterns that occurred when the malware operated on a pre-prepared system. This technique produced qualitative data regarding the activities and techniques used by both types of ransomware. Data was collected through literature studies related to malware detection and analysis, and direct testing of various file types to detect malware patterns and characteristics [19].

The collected data will be analysed using static and string analysis techniques for the malware codes found. Static analysis is used to measure the frequency of specific activities, such as file encryption frequency or network communication patterns. Malware code analysis is conducted to identify the functions and workflow of the malware. Qualitative coding will be used to document and categorise the various methods or strategies used by both types of ransomware to evade detection by security systems. The data will be analysed using static analysis techniques to evaluate the accuracy of the Machine Learning model's predictions. System performance will be evaluated based on user response and analysis speed [20].

III. RESULT AND ANALYSIS

The developed system is capable of detecting malware with higher accuracy than conventional methods, providing an interface that is easy to use and accessible to users with varying levels of technical understanding, accelerating the malware analysis process with results that can be directly interpreted by users, and raising awareness of the importance of cybersecurity in everyday digital life.

Figure 2. Malware Analysis System

Figure 2 shows the UI display of a malware analysis system called FauzanMalware. The system.

Figure 3. How to Use the System

Figure 3 shows how to use the malware analysis system. There are four steps, starting from uploading the file to waiting for the analysis results.

Figure 4. Upload Analysis File

Figure 4 indicates that the file to be analysed has been uploaded. The file is a trojan-type malware that needs to be analysed.

Figure 5. Security Verification

Figure 5 shows the security verification to identify whether the user is a human or a robot. Users are asked to fill in the verification from the simple math provided.

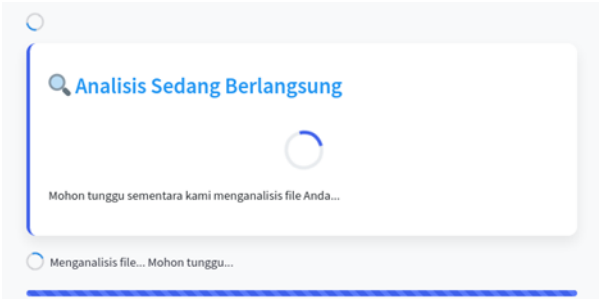


Figure 6. File Analysis Process

Figure 6 shows the file analysis process. Users only need to wait a few moments to get the results.

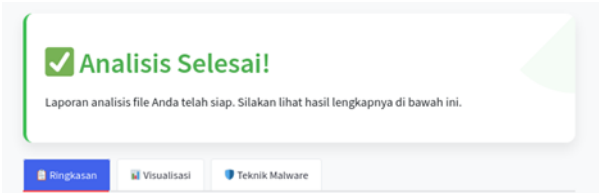


Figure 7. File Analysis Complete

Figure 7 indicates that the file has been analysed and the analysis report is ready for the user to read.

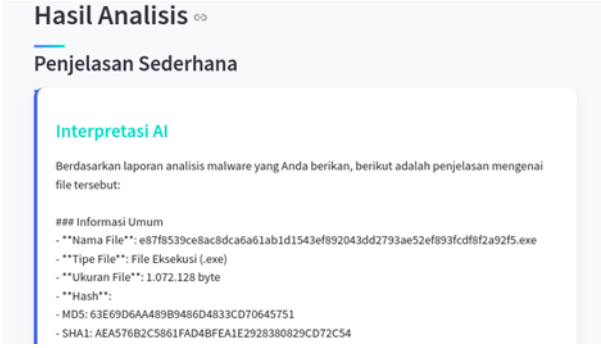


Figure 8. File 1 Analysis Results

Figure 8 provides general information about the file name, file type, file size, and file hash consisting of MD5, SHA1, and SHA256, which are shown in the following figure.

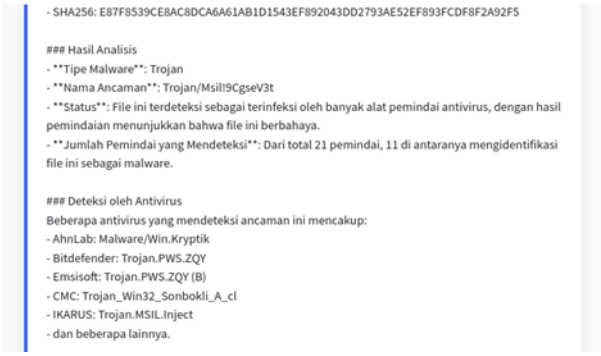


Figure 9. File 2 Analysis Results

Figure 9 shows the results, which provide information on the type of malware, threat name, status, number of detections, and a list of detections by antivirus programs, including AhnLab, BitDefender, Emisisoft, CMC, and IKARUS.

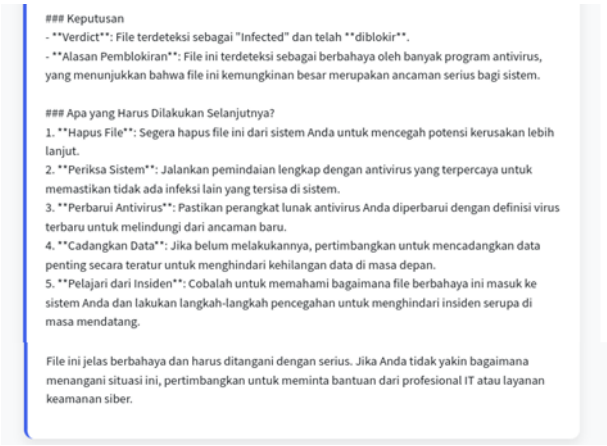


Figure 10. File 3 Analysis Results

Figure 10 shows the results, which provide decision information consisting of the Verdict and Reason for Blocking. There are also results that provide security recommendations for users, including delete the file, check the system, update the antivirus, back up data, and learn from the incident.

Figure 11. Technical Report File 1

Figure 11 shows a technical report that provides detailed information about the contents of the Trojan program, including the last time the Trojan was analysed, the malware family group, the malware type, and the analysis history.

Figure 12. Technical Report File 2

Figure 12 shows a follow-up technical report from the previous figure, providing information such as the name of

the threat, sandbox, file ID, data ID, process info, progress percentage, and reason for blocking.

```
7 : "Trojan.MSIL"
8 : "W32/MSIL_Kryptik.KUK.gen!Eldorado"
9 : "Trojan.Win32.MSIL_Heur.A"
10 : "Malware"
]
"result" : "Blocked"
"post_processing" : {
  "actions_failed" : ""
  "actions_ran" : ""
  "converted_destination" : ""
  "converted_to" : ""
  "copy_move_destination" : ""
}
"verdicts" : [
  0 : "Infected"
]
"profile" : "multiscan"
```


Figure 13. Technical Report File 3

Figure 13 shows a follow-up technical report from the previous figure, providing information in the form of blocking results, verdicts, and profiles in the analysis.

```
"blocked_reason" : "Infected"
}
"scan_results" : {
  "scan_details" : {
    "AhnLab" : {
      "scan_result_i" : 1
      "scan_time" : 71
      "threat_found" : "Malware/Win.Kryptik"
      "def_time" : "2025-05-02T07:54:16.770Z"
    }
    "Avira" : {
      "scan_time" : 15
      "def_time" : "2025-05-02T07:52:41.157Z"
      "scan_result_i" : 0
      "threat_found" : ""
    }
```


Figure 14. Technical Report File 4

Figure 14 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely AhnLab and Avira.

```
"Bitdefender" : {
  "scan_result_i" : 1
  "scan_time" : 47
  "threat_found" : "Trojan.PWS.ZQY"
  "def_time" : "2025-05-02T07:53:52.473Z"
}
"Bkav Pro" : {
  "scan_time" : 317
  "def_time" : "2025-05-02T07:12:39.252Z"
  "scan_result_i" : 0
  "threat_found" : ""
}
"ClamAV" : {
  "scan_time" : 1036
  "def_time" : "2025-05-02T08:03:56.005Z"
  "scan_result_i" : 0
```


Figure 15. Technical Report File 5

Figure 15 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely BitDefender, Bkav Pro, and ClamAV.

```
  "threat_found" : ""
}
"CMC" : {
  "scan_result_i" : 1
  "scan_time" : 6
  "threat_found" : "Trojan_Win32_Sonbokli_A_cl"
  "def_time" : "2025-05-02T07:58:36.461Z"
}
"CrowdStrike Falcon ML" : {
  "scan_result_i" : 1
  "scan_time" : 85
  "threat_found" : "win/malicious_confidence_100"
  "def_time" : "2025-05-02T07:32:42.906Z"
}
"Emsisoft" : {
  "scan_result_i" : 1
  "scan_time" : 38
```


Figure 16. Technical Report File 6

Figure 16 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely CMC, CrowdStrike Falcon ML, and Emisoft.

```
  "threat_found" : "Trojan.PWS.ZQY (B)"
  "def_time" : "2025-05-02T07:25:39.650Z"
}
"IKARUS" : {
  "scan_result_i" : 1
  "scan_time" : 8
  "threat_found" : "Trojan.MSIL.Inject"
  "def_time" : "2025-05-02T07:58:50.598Z"
}
"K7" : {
  "scan_result_i" : 3
  "def_time" : "2025-05-02T00:00:00.000Z"
  "threat_found" : ""
}
"Lionic" : {
  "scan_time" : 60
```


Figure 17. Technical Report File 7

Figure 17 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely IKARUS, K7, and Lonic.

```
  "def_time" : "2025-05-02T08:06:37.875Z"
  "scan_result_i" : 0
  "threat_found" : ""
}
"McAfee" : {
  "scan_time" : 321
  "def_time" : "2025-05-02T08:04:19.583Z"
  "scan_result_i" : 0
  "threat_found" : ""
}
"NANOAV" : {
  "scan_result_i" : 1
  "scan_time" : 39
  "threat_found" : "Trojan.Win32.Taskun.kxdbfq"
  "def_time" : "2025-05-02T07:34:35.729Z"
}
```


Figure 18. Technical Report File 8

Figure 18 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely McAfee and NANOAV.

Figure 19. Technical Report File 9

Figure 19 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely Quick Heal, TACHYON, and Varist.

Figure 20. Technical Report File 10

Figure 20 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely Webroot SMD and Xvirus Anti-Malware.

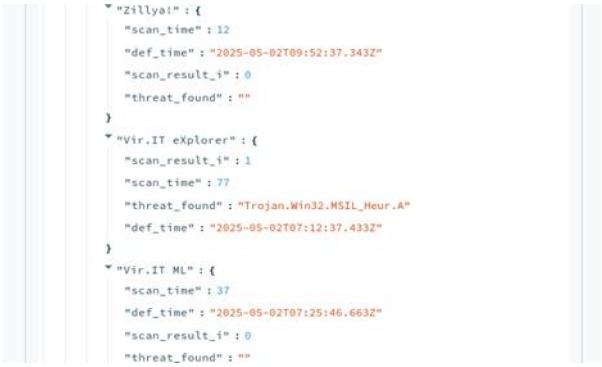


Figure 21. Technical Report File 11

Figure 21 shows a follow-up technical report from the previous figure, providing information in the form of scan results from several antivirus programs, namely Zillya and Vir.IT eXplorer, and Vir. IT ML.

Figure 22. Technical Report File 12

Figure 22 shows a follow-up technical report from the previous figure, providing information in the form of scan results, scan statistics, and file information.

Figure 23. Technical Report File 13

Figure 23 shows a follow-up technical report from the previous figure, providing information in the form of scan results in file hash identification, file type, process information, and storage status.

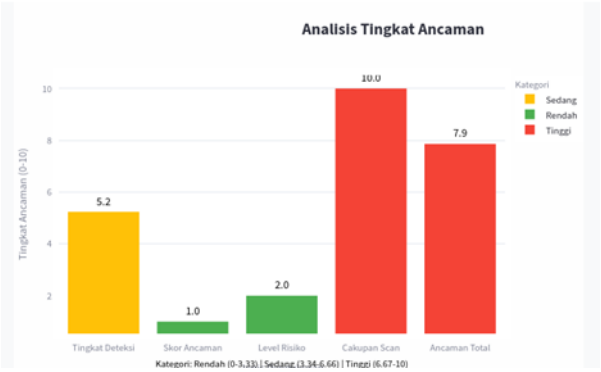


Figure 24. Threat Level Graph

Figure 24 shows a bar graph titled 'Threat Level Analysis.' This graph shows five threat assessment indicators on a scale of 0-10. The following are the calculations in the graph and their colour categories:

- Detection Level: 5.2 – Moderate category (yellow).
- Threat Score: 1.0 – Low category (green).
- Risk Level: 2.0 – Low category (green).
- Scan Coverage: 10.0 – High category (red).
- Total Threat: 7.9 – High category (red).

There is a category legend at the bottom of the graph: Low: 0 – 3.33, Medium: 3.34 – 6.66, and High: 6.67 – 10

This graph provides a visual representation of the magnitude of the detected threat based on several technical parameters.

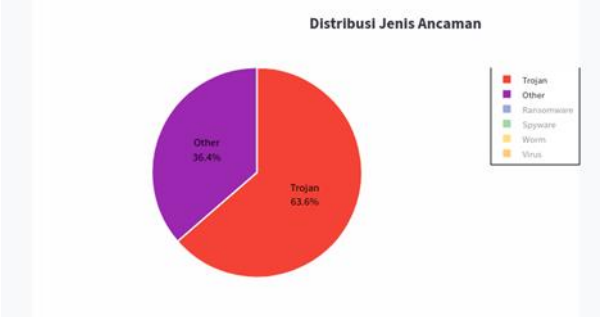


Figure 25. Threat Level Diagram

Figure 25 is a pie chart labelled 'Threat Type Distribution'; the entire chart shows that 100% of detected threats are Trojan types. The red colour dominates the whole chart, indicating that no other threat types were detected, such as Ransomware, Spyware, Worms, Viruses, or the 'Other' category. It can be concluded that only Trojan threats were detected.

Figure 26. PDF Report Results

Figure 26 shows the results of the analysis report in PDF format, which users can download and read anywhere and anytime offline.

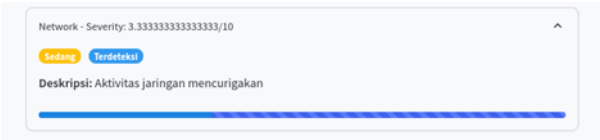


Figure 27. Malware Technique Results

Figure 27 shows a malware technique classified as Medium (yellow) describing suspicious network activity.

Figure 28. Security Consultation Feature

Figure 28 shows the security consultation feature of the malware analysis system called 'Chat with AI'. This feature lets users consult about malware analysis results or other cybersecurity topics.

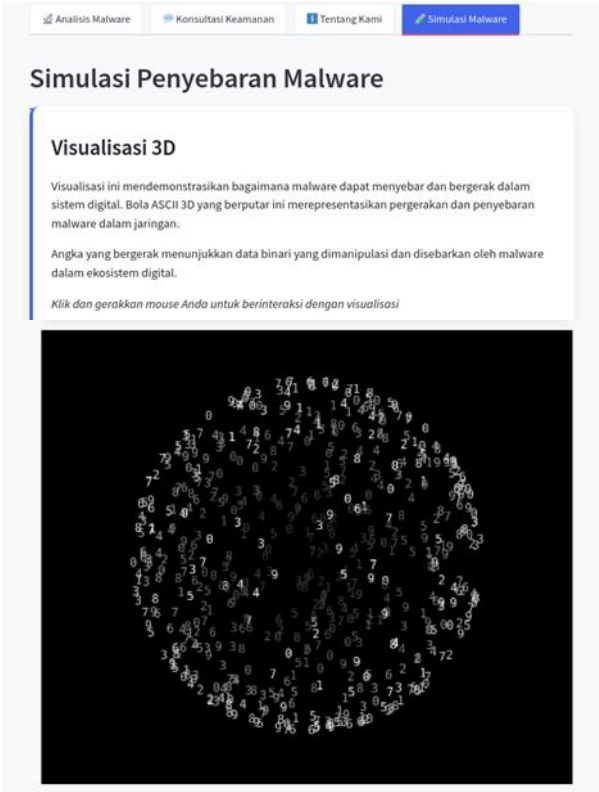


Figure 29. Malware Spread Simulation

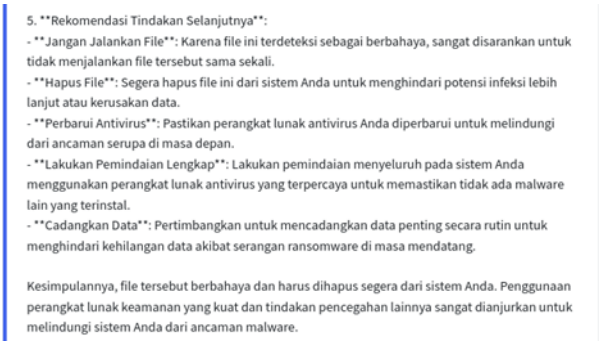
Figure 29 shows a web page titled 'Malware Spread Simulation' with the subtitle '3D Visualisation'. This page explains that 3D visualisation is required to demonstrate how malware spreads and behaves in digital systems. Specifically, the rotating 3D ASCII ball symbolises the movement of malware in the network, while the numbers that appear represent the manipulative binary data of malware in the digital ecosystem.

Figure 30. Malware Movement

Figure 30 illustrates various vectors of malware spread or movement through phishing emails, malicious downloads, security exploits, infected media, and network attacks.

In addition, the system can provide more effective recommendations for the detection, prevention and mitigation of threats caused by ransomware or other types of malware. These recommendations will include the development of stronger security solutions and better

strategies for organizations to deal with the growing threat of malware or ransomware. This includes system monitoring techniques, early detection, and implementation of more reliable data recovery protocols.


5. **Rekomendasi Tindakan Selanjutnya**:
- **Jangan Jalankan File**: Karena file ini terdeteksi sebagai berbahaya, sangat disarankan untuk tidak menjalankan file tersebut sama sekali.
- **Hapus File**: Segera hapus file ini dari sistem Anda untuk menghindari potensi infeksi lebih lanjut atau kerusakan data.
- **Perbarui Antivirus**: Pastikan perangkat lunak antivirus Anda diperbarui untuk melindungi dari ancaman serupa di masa depan.
- **Lakukan Pemindaian Lengkap**: Lakukan pemindaian menyeluruh pada sistem Anda menggunakan perangkat lunak antivirus yang terpercaya untuk memastikan tidak ada malware lain yang terinstal.
- **Cadangkan Data**: Pertimbangkan untuk mencadangkan data penting secara rutin untuk menghindari kehilangan data akibat serangan ransomware di masa mendatang.

Kesimpulannya, file tersebut berbahaya dan harus dihapus segera dari sistem Anda. Penggunaan perangkat lunak keamanan yang kuat dan tindakan pencegahan lainnya sangat dianjurkan untuk melindungi sistem Anda dari ancaman malware.

Figure 31. Threat Prevention Recommendations

The results of this research can also provide a basis for further research on more sophisticated malware or ransomware. Therefore, the results of this research are not only relevant for academics but also for practitioners and cybersecurity professionals who are directly involved in countering and investigating malware or ransomware attacks. The information obtained from this research will be a useful reference in developing new tools and techniques to improve system resilience against cyber threats, particularly from ransomware. The results obtained can also enrich the existing literature in the field of cybersecurity, contribute to knowledge in the field of malware analysis, and encourage the implementation of more innovative and adaptive solutions to increasingly complex threats in cyberspace [21].

VI. CONCLUSION

This research proposes a new solution in web-based malware analysis by utilizing Machine Learning technology. By integrating the Random Forest algorithm, MetaDefender API, and GPT API, this system can provide analysis results that are accurate, fast, and easily understood by non-technical users. The implementation of this system is expected to improve cybersecurity and help the community detect malware threats more effectively.

REFERENCES

[1] O. Adiputra and E. Setiawan, "Klasifikasi Malicious URL Menggunakan Algoritma Improved Random Forest Dan Random Forest Berbasis Web," J. Sains dan Inform., vol. 09, no. 01, pp. 8–14, 2023, [Online]. Available: https://scispace.com/pdf/klasifikasi-malicious-url-menggunakan-algoritma-improved-1301amsz.pdf.

[2] J. B. Higuera et al., "Benchmarking Android Malware Analysis Tools," Electronics, vol. 13, no. 11, pp. 1–28, 2024, doi: https://doi.org/10.3390/electronics13112103.

[3] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," IEEE Xplore, vol. 8, pp. 124579–124607, 2020, doi: 10.1109/ACCESS.2020.3006143.

[4] F. A. Rafrastara, R. A. Pramunendar, D. Prabowo, E. Kartikadarma, and U. Sudibyo, "Optimasi Algoritma Random Forest menggunakan Principal Component Analysis untuk Deteksi Malware," J. Teknol. dan Sist. Inf. Bisnis, vol. 5, no. 3, pp. 217–223, 2023.

[5] I. P. Y. A. Ariwanta, K. Y. E. Aryanto, and I. G. A. Gunadi, "Suricata accuracy optimization based on live analysis using One-Class Support Vector Machine method and Streamlit framework," J. Tek. Inform., vol. 5, no. 2, pp. 301–315, 2024, doi: https://doi.org/10.52436/1.jutif.2024.5.2.1822.

[6] J. Rafapa and A. Konokix, "Technique with Recent Variants Ransomware Detection Using Aggregated Random Forest Technique with Recent Variants," J. Appl. Mach. Learn. Secur., vol. 11, no. 2, pp. 55–67, 2024.

[7] J. P. Bororing, "Evaluating user-friendly interfaces in cybersecurity tools," Human-Centric Cybersecurity Rev., vol. 9, no. 3, pp. 101–112, 2022.

[8] M. A. Kurniawan and S. Bramasto, "Analisis Malware Menggunakan Metode Dynamic Analysis," TECHNOPEX, pp. 860–865, 2024, [Online]. Available: https://technopex.iti.ac.id/ocs/index.php/tpx24/tpx24/paper/viewFile/1796/687.

[9] M. R. Akbar and T. Sutabri, "Implementasi Teknologi AI Dalam Deteksi dan Pencegahan Serangan Malware pada Jaringan Komputer Perusahaan," IJM Indones. J. Multidiscip., vol. 2, no. 3, pp. 20–30, 2024, [Online]. Available: https://journal.csspublishing.com/index.php/ijm/article/view/700.

[10] S. M. A. Jafari, "Streamlining the Selection Phase of Systematic Literature Reviews (SLRs) Using AI-Enabled GPT-4 Assistant API," J. AI Res. Optim., vol. 6, no. 1, pp. 34–45, 2024, [Online]. Available: https://arxiv.org/abs/2402.18582.

[11] Y. W. Sitorus, P. Sukarno, and S. Mandala, "Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest," in E-Proceeding of Engineering, 2021, vol. 8, no. 6, pp. 12500–12518, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/16864.

[12] J. D. Nugraha, A. Budiyono, and A. Almaarif, "Analisis malware berdasarkan API call memory dengan metode deteksi signature-based," J. Anal. Keamanan Siber, vol. 4, no. 3, pp. 45–59, 2019.

[13] A. Z. Toscana, C. Setianingsih, and M. W. Paryasto, "Integrasi Streamlit pada Aplikasi Berbasis Web dengan Algoritma YOLO V8 dan Teknologi Drone untuk Identifikasi Jenis dan Estimasi Tinggi Pohon," in E-Proceeding of Engineering, 2024, vol. 11, no. 3, pp. 1828–1831.

[14] M. M. Alvanof, Bustami, and R. K. Dinata, "Penerapan Algoritma Random Forest dalam Deteksi dan Klasifikasi Ransomware," JETI (Jurnal Elektron. dan

Teknol. Informasi), vol. 5, no. 2, pp. 23–31, 2024, doi: https://doi.org/10.5201/jet.v5i2.488.

[15] K. Inayah and K. Ramli, "Analisis Kinerja Intrusion Detection System Berbasis Algoritma Random Forest Menggunakan Dataset Unbalanced Honeynet BSSN," J. Teknol. Inf. dan Ilmu Komput., vol. 4, no. 11, pp. 867–876, 2024, doi: 10.25126/jtiik1148911.

[16] Eliando and A. B. Warsito, "LockBit Black Ransomware On Reverse Shell: Analysis of Infection," Cogito Smart J., vol. 9, no. 2, pp. 228–240, 2023, doi: https://doi.org/10.31154/cogito.v9i2.494.228-240.

[17] O. Akinyemi, R. Sulaiman, and N. Abosata, "Analysis of the LockBit 3.0 and its infiltration into Advanced's infrastructure crippling NHS services," 2023, doi: 10.48550/arXiv.2308.05565.

[18] A. O. Ojo, "Ransomware trends and mitigation strategies: A comprehensive review," Glob. J. Eng. Technol. Adv., vol. 22, no. 3, 2025, doi: 10.30574/gjeta.2025.22.3.0038.

[19] R. B. Hadiprakoso, W. R. Aditya, and F. N. Pramitha, "Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning," CyberSecurity dan Forensik Digit., vol. 5, no. 1, pp. 1–5, 2022, doi: https://doi.org/10.14421/csecurity.2022.5.1.3116.

[20] W. Yunanri, Y. B. Fitriana, S. Isabela, and F. Hamdani, "Deteksi Serangan Malware Pada Web Aplikasi Menggunakan Metode Malware Analis Dinamis dan Statis," Digit. Transform. Technol., vol. 4, no. 1, pp. 461–470, 2024, doi: https://doi.org/10.47709/digitech.v4i1.4270.

[21] Mawardi. C, Kuswoyo. D, Falah. N, "Implementation of A Cyberpanel-Based Partial Cloud Server As A Prevention Of Security Information Management System (SIMS) Encryption," The First Jakarta International Conference on Multidisciplinary Studies Towards Creative Industries, JICOMS 2022, Jakarta, Indonesia, Nov. 2022, pp. 153, doi: https://doi.org/10.4108/eai.16-11-2022.2326064.

[22] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[23] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[24] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[25] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[26] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[27] A. Karnik, "Performance of TCP congestion control with rate feedback:TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute ofScience, Bangalore, India, Jan. 1999.

[28] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Renocongestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[29] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 199