A Malware Analysis Approach for Identifying Threat Actor Correlation Using Similarity Comparison Techniques

Ahmad Naim Irfan, Suriayati Chuprat, Mohd Naz'ri Mahrin, Aswami Ariffin
Universiti Teknologi Malaysia, Malaysia

***Abstract*—Cybersecurity is essential for organisations to protect critical assets from cyber threats in the increasingly digital and interconnected world. However, cybersecurity incidents are rising each year, leading to increased workloads. Current malware analysis approaches are often case-by-case, based on specific scenarios, and are typically limited to identifying malware. When cybersecurity incidents are not handled effectively due to these analytical limitations, operations are disrupted, and an organisation's brand and client trust are negatively impacted, often resulting in financial loss. The aim of this research is to enhance the analysis of Advanced Persistent Threat (APT) malware by correlating malware with its associated threat actors, such as APT groups, who are the perpetrators or authors of the malware. APT malware represents a highly dangerous threat, and gaining insight into the adversaries behind such attacks is crucial for preventing cyber incidents. This research proposes an advanced malware analysis approach that correlates APT malware with threat actors using a similarity comparison technique. By extracting features from APT malware and analysing the correlation with the threat actor, cybersecurity professionals can implement effective countermeasures to ensure that organisations are better prepared against these sophisticated cyber threats. The solution aims to assist cybersecurity practitioners and researchers in making informed decisions by providing actionable insights and a broader perspective on cyber-attacks, based on detailed information about malware tied to specific threat actors.**

***Keywords*—Malware analysis; APT group; threat actor correlation; CTI**

I. Introduction

The increasing number of cybersecurity incidents is a significant challenge faced by organisations worldwide. Throughout the year, many organisations must deal with cyber incidents involving malware. According to a report by Trend Micro, there has been a 382% increase in blocked malicious files, such as malware [1] deployed by threat actors, These threats are continuously adapting to organisations' cyber defences. Threat actors can bypass these defences due to their ever-improving modus operandi [2] [3] including the malware they use [1], which targets multiple devices, [4] such as computers and smartphones. [5] [6]. Threat actors aim to avoid detection by making it increasingly difficult to identify malicious files. This is particularly evident in cases of ransomware, where cyber-attacks are becoming more sophisticated [7]. As threat actors deploy new diversion and evasion techniques, they are able to avoid detection, highlighting the growing complexity of cyber threats [8].

Identifying malicious activities is crucial when dealing with malware found during cyber incidents, such as data breaches [12]. Cyber-attacks carried out by threat actors, particularly APT groups, are highly sophisticated and have a severe impact on victims. For example, the Lazarus Group, a state-sponsored threat actor, was reported to have attacked Automated Teller Machines (ATMs) and banks in India [13]. In addition to causing disruption, financial gain and espionage are common motivations for APT groups to execute cyber-attacks. These attacks are high-stakes because APT groups are highly motivated, skilled, and resourceful, with perpetrators often not stopping until they meet their objectives. Moreover, APT groups commonly employ stealth, anti-analysis techniques, and covert communication to evade detection, making malware analysis both difficult and time-consuming [14]. Dealing with APT groups also takes considerable time due to the scale of the cyber-attacks, which can affect targets across multiple organisations and countries.

The identification of malware, especially APT malware, is unique as it involves multiple factors and often depends on a case-by-case basis. Factors such as file type, extracted malware data, and the purpose of the malware analysis all play a role in shaping the malware analysis approach. As a result, many researchers have developed specific approaches based on these factors, such as classifying malware by type. However, current malware analysis approaches primarily focus on identifying malware itself. There is an opportunity to broaden the purpose of malware analysis by also identifying the threat actor responsible for developing the malware. Correlating malware with its associated threat actor is valuable for identifying shared features. The identification of similar features helps in discovering links between threat actors, where malware from the same actor can be correlated.

One of the factors contributing to the rising number of cybersecurity incidents is the complexity of malware analysis, which is unique and depends on a case-by-case basis. Previous research on malware analysis approaches typically focuses on factors such as the data used in experiments, features extracted, the purpose of the analysis, the medium of analysis, and how results are measured. However, current malware analysis approaches are often limited, as they primarily focus on identifying malware rather than the threat actor behind it.

Therefore, the aim of our research is to enhance an APT malware analysis approach using a similarity comparison technique to identify the threat actor.

II. Malware Analysis Approach Considerations

Malware is one of the key artefacts found in cyber-attacks and serves as a valuable source of Cyber Threat Intelligence (CTI) [9], [10], [11], data. It contains harmful code with unique signatures and behaviours [15]. Identifying these signatures and behaviours is challenging, as they are often unique to the design of the malware author. However, some malware is derived from known variants, where the signature and behavioural patterns have already been identified [16]. Current human capabilities and technologies, such as antivirus software, rely on predefined malware signature databases that require constant updates to detect new threats. The large volume of malware makes it impractical to analyse each piece manually, which is why automated technology is used to conduct these analyses.

A. Malware Group

Malware is commonly grouped by its type. Since there are many types of malware, categorising them in this way helps identify new variants within the same malware family or discover entirely new ones. Examples of common malware types include backdoors, botnets, ransomware, spyware, keyloggers, rootkits, viruses, and worms [17]. The advantage of classifying malware by type is that it enables the identification of malware families. Grouping malware in this manner improves detection accuracy, as similar types tend to share common traits. Malware type identification is achieved by analysing patterns in malware behaviour and grouping them based on these similarities.

Currently, there is a growing body of research focused on attributing malware, as it has become increasingly sophisticated. This requires analysing malware from different perspectives, such as identifying traits to group malware by platform. In addition to traditional Personal Computer (PC) malware, malware is now being developed for a wide range of platforms, including Android malware for mobile devices, Industrial Control Systems (ICS) malware for Operational Technology (OT) systems, and Internet of Things (IoT) malware for appliances connected to the internet. This approach allows malware to be grouped according to the platform it is designed to target. For example, IoT malware refers to any type of malware developed to compromise a network of connected devices, as well as the technology that facilitates communication between these devices, the cloud, and other devices within the network.

In addition to being grouped by platform, malware is also classified based on its authors, linking it to the respective threat actor. Connecting the malware to the threat actor helps gain insights into the objectives behind an attack and understand the motive of the threat actor [29]. This information is then used to build a profile of the threat actor, detailing the tools, targets, and preferred attack vectors. Having a profile of the threat actor aids in anticipating future attacks by enabling necessary preparations to enhance the organisation's cybersecurity posture. For example, incorporating known signatures and the behavioural traits of the malware into cybersecurity controls to detect or block possible threats identified [30].

The challenge in the current ecosystem is identifying specific malware groupings, rather than categorising by type, as some malware exhibits the functionalities of two or more types. For example, China Chopper is a piece of malware that displays the capabilities of a trojan, infostealer, and password brute-force attack tool, among others [18] [19]. This demonstrates that sophisticated malware has a range of functionalities, making it difficult to group strictly by type. However, despite this complexity, malware still exhibits attributes that are linked to specific threat actors, such as APT group [20]. Grouping malware by its authors enables malware analysts to attribute it to a specific threat actor group or link it to a particular threat campaign [31]. This practice enhances threat detection by providing critical insights into adversarial motives, which, in turn, facilitates proactive defence measures.

B. Malware Analysis Environment

There are various ways to build malware analysis environments, depending on the data being analysed and the specific experimental scenario. One option is to use a dedicated physical machine for performing the analysis. However, this approach is time-consuming and inflexible, as cleaning the machine and reinstalling tools after each analysis session is cumbersome. After each analysis, the machine must be cleaned, and tools need to be reinstalled. An alternative approach is to use hypervisors and preinstalled tools [21]. In this setup, the machine is simulated through virtual machines (VMs). VMs offer several advantages, including network configurations that allow for host-only connections, which prevent the machine from connecting to the internet. VMs also include a snapshot function, enabling users to capture the system's state once the machine and applications are properly configured. This snapshot is used to revert to the captured state whenever required.

C. Related Work

Related works on malware analysis approaches typically use either generic malware or APT malware for experiments. Research on APT malware often involves classification to identify APT attacks based on common features extracted from malware samples belonging to different APT groups. Additionally, features extracted from APT malware are used to distinguish between APT and non-APT malware. However, these studies do not specifically analyse PE format APT malware. Given that Windows OS is widely targeted in cyber-attacks, a dedicated extraction and analysis approach is required to gain a deeper understanding of PE-based APT malware. A comparison of related research works is presented in Table I.

TABLE I. MALWARE ANALYSIS APPROACH COMPARISON STUDY

Research Work	Malware Analysis Approach				
	Data used in Experiment	Features Extracted	Analysis Purpose	Analysis Medium	Result Measurement
Torabi, S., Dib, M., Bou-Harb, E., Assi, C., & Debbabi, M. (2021)[22]	IoT Malware (Collected using IoT-based honeypot)	Strings	Visualise Covid-related malware clusters based on strings attribute	Similarity Measurement Based on Strings Attribute to determine covid related IoT malware	Members, percentage, density
(Xu et al., 2021)[23]	APT Malware (cyber-research/APT Malware Github)	API calls	APT Malware Classification (If it is an APT malware)	Adaboost feature selection and LightGBM	Accuracy, precision, recall, F1 score
(Hu & Hsieh, 2021)[24]	APT Malware (cyber-research/APT Malware Github)	Hexadecimal and ASCII codes (APT PE samples were converted to PNG images with a fixed width of 256 pixels)	APT Malware Classification (If it is an APT malware)	Convolutional Neural Network	Avg. Train Accuracy and Max Train Accuracy
(X. Han et al., 2021)[25]	APT Malware (cyber-research/APT Malware Github)	Binary code collection and network behaviour (Grayscale image conversion)	APT Malware Classification (If it is an APT malware)	Convolutional Neural Network	Accuracy, precision, recall, F1 score
(Do Xuan & Huong, 2022)[26]	APT malware downloaded from Interactive Online Malware Sandbox (any run app)	Processes from Event ID	Classify an APT or non-APT malware	Graph Neural Network	Experimental Scenarios to measure to measure effectiveness, Accuracy, precision, recall, F1 score
Enhanced Malware Analysis Approach	Only PE file type of APT Malware (cyber-research/APT Malware Github) and vxunderground	Strings and Import Address Table	Determine features of APT malware and using similarity comparison technique to correlate with threat actor	Similarity Measurement based on strings and IAT attribute	Experiment Scenarios based on Similarity Concept

Table I presents the enhanced approach we propose, which uses the similarity comparison technique to correlate APT malware with its author. The enhanced approach is simulated through experimental scenarios designed to evaluate the results. Table I also highlights that this enhanced approach is specifically tailored for analysing PE file-type APT malware and expands the use of the similarity comparison technique, as well as the developed experimental scenarios. Therefore, this research aims to combine various techniques and methods to analyse APT malware and extract information for Cyber Threat Intelligence (CTI) purposes.

III. ENHANCED MALWARE ANALYSIS APPROACH

The enhanced approach to analysing APT malware follows the entire process flow, from feature extraction to data analysis. This approach is presented visually to demonstrate the process, which is replicable using the preferred tools and methods. The enhanced approach is illustrated in Fig. 1.

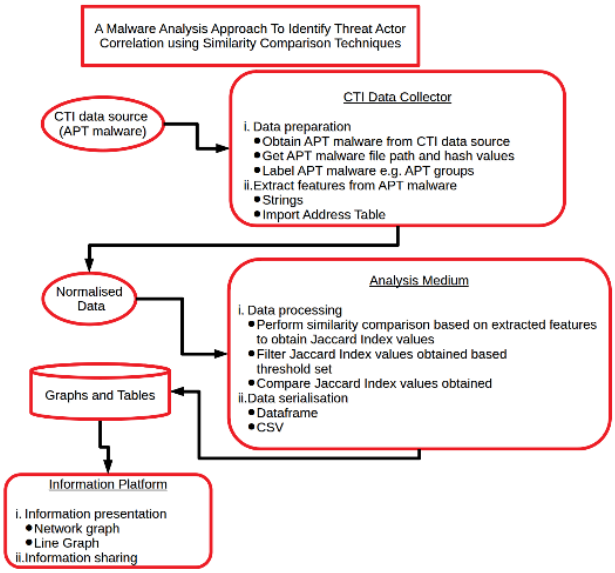

Fig. 1. Enhanced APT malware analysis approach.

Fig. 1 shows the enhanced approach for analysing APT malware to identify threat actor correlation using similarity comparison techniques. The approach takes APT malware files as input via the CTI data collector component, which performs data extraction and preparation. The normalised data is then processed and serialised by the analysis medium component. Finally, the graphs and tables generated by the analysis medium are presented and shared through the information platform component.

The first step in the approach for the CTI data collector component is to identify the APT malware source for data collection. CTI data sources are categorised into open-source, closed, or subscription-based categories. Open-source repositories, maintained by the community, include forums and websites that offer free access to content. However, the quality and availability of this content depend on how well the repository is maintained. Closed and subscription-based repositories, on the other hand, are accessible only to a select group, typically based on membership arrangements that may require payment or affiliation with an organisation. To ingest APT malware from a source, the repository typically provides a download feature to store data files locally or an API to pull data to an external storage location, such as a server or cloud.

Once APT malware is in the designated storage location, the contents of the files are extracted to obtain the relevant data. Data extraction is performed using specialised tools, which may be open-source or proprietary. Alternatively, a custom program can be written to perform the extraction process. This approach offers greater flexibility, as the features are not restricted by the limitations of open-source or proprietary data extraction tools. To develop a data extraction program, programming languages and scripting libraries typically include modules to read files and extract data based on the file format.

Initial analysis is performed on the extracted APT malware data. This step helps identify key content within the data and prepares it for subsequent processing. By reviewing the extracted data, relevant attributes are selected for analysis, and appropriate data structures are chosen based on these observations. Additionally, this step includes data labelling, which adds context to the content by tagging relevant data. Observing the data also provides an overview of its contents, allowing for the identification and removal of noise through sanitisation. Sanitisation involves filtering out irrelevant file types and removing empty rows to refine the data.

Once the data is prepared during the data collection phase, analysis mediums are used to transform it into meaningful information. At this point, the data is still in its raw format, and processing is necessary to derive actionable insights. Analysis mediums, typically built using programming languages, scripts, or available tools, perform the required processing and generate output results. Libraries and modules play a crucial role in this process, as they save time in developing the analysis mediums and allow the developer to focus more on the logic needed to process the data.

The first decision when starting data processing is to choose the appropriate algorithm for the task. In artificial intelligence, the two main options are machine learning and deep learning. The next decision is to select the data processing algorithm, which is based on the data preparation step completed earlier. A data processing algorithm is a series of instructions designed to process the data. Libraries are often used to incorporate common functions such as validation, sorting, summarising, and aggregation into the algorithm. These libraries vary across programming languages or scripts, and custom functions are written to perform tasks beyond the available scope. Pseudocode is frequently shared by the community to assist in building custom functions, and many libraries are specifically designed to support AI.

Once the data is processed, serialisation is performed to convert an object into a stream of bytes for storing the results. Common formats for storing results include CSV for tabular data and PNG for images. Serialised data are records kept for future reference when needed. Choosing the appropriate format for saving data is crucial to ensure that it is both preserved and easily shareable through the information platform.

All results obtained from the analysis are gathered on the information platform. Web-based platforms, such as blogs, wikis, and dashboards, are used to transmit and display these results. A web-based information platform is chosen because it allows for customisation in how information is presented and provides graphical tools to assist in visualising the data. Additionally, sharing features and APIs are available on web-based platforms to relay information to relevant parties.

Based on the results obtained, suitable ways to present the information include visuals such as graphs, histograms, bar charts, pie charts, and tables. These visuals help describe and interpret the data, assisting recipients in the decision-making process. Multiple visual options are available to present the information effectively, depending on the context. Key considerations when presenting information include the purpose, the recipient's background, and the structure of the information flow. To manage these visuals and considerations, a platform like a dashboard is often used to consolidate all the information in one view. A dashboard allows recipients to access information and make queries efficiently.

Before sharing information containing the analysis results, the parties who need to receive the information are identified. This step is crucial to prioritise information sharing based on the roles of personnel within the organisation. It functions as a call tree, alerting relevant personnel according to a layered, hierarchical communication model, ensuring the right people are notified of the threat. This enables the necessary preparations to be made in response to the threat. Communication methods include multiple channels, such as email, chat, SMS, and voice calls for emergencies. The information platform serves as a central medium, accessible to recipients, allowing them to pull the information when needed.

A. Malware Analysis Study

A malware analysis environment is established to experiment with and evaluate the proposed APT malware analysis approach. This environment integrates open-source tools and custom scripts to extract and compare malware features, facilitating the identification of similarities across various APT malware datasets. The environment is designed to align with the APT malware analysis methodology, simulating the analysis process using data derived from APT malware samples. The results of these experiments are assessed by reviewing the analysis outcomes. The APT malware analysis approach serves as the foundation for the environment's architecture, which is built using open-source technologies. Tools are integrated into the environment to perform similarity comparisons, offering valuable insights for

cybersecurity practitioners. These insights enable prompt actions, such as malware detection, as illustrated in Fig. 2.

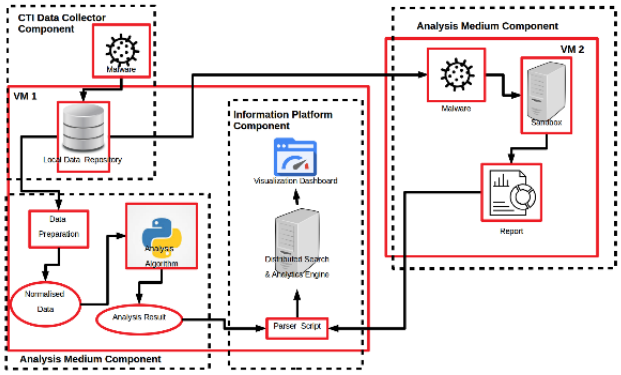

Fig. 2. Malware analysis environment.

Fig. 2 illustrates the architecture of the malware analysis environment, which implements the proposed APT malware analysis approach. Two Virtual Machines (VMs) are used in the environment to manage the integration of components separately. VM 2 runs Windows 10 for sandbox deployment, while VM 1 is used for tool installation and runs Ubuntu. The requirements and installation procedures for each tool or software used are documented on the respective tool websites. The suitability and functionality of the tools are evaluated in advance to avoid any installation or deployment issues that could hinder their operation. The tools used in the malware analysis environment are listed in Table II.

TABLE II. MALWARE ANALYSIS ENVIRONMENT

Malware Analysis Environment	Description
Host Hardware Specification	• CPU: Intel i7-13700F • RAM: 64GB (16GB assigned to VM1 and VM2)
Dataset 1 Publicly available dataset from https://github.com/cyber-research/APTMalware	The APT Malware Dataset contains over 3,500 malware samples in various file formats, such as .exe and .pdf, which are associated with 12 APT groups allegedly sponsored by five different nation-states. This dataset is primarily used for benchmarking different machine learning approaches in the context of authorship attribution. It can also serve as a valuable resource for future benchmarks or malware research.
Dataset 2 Publicly available dataset from VX Underground (https://vx-underground.org/)	The malware repository is regularly updated and maintained, with each sample properly attributed to specific cyber incidents and threat actors based on CTI reports. As such, malware from this repository serves as a suitable dataset for this research experiment. Additionally, this repository has been used as an experimental dataset in the research by Piskozub et al. (2021) [27]. The APT malware samples from 2022 and 2023 were used in this experiment.
Virtual Machine 1 (VM 1, Linux Machine)	The experiment tools installed on the virtual machine include Jupyter Lab is used to write the Python algorithms that is used for analysis durin the experiment. Elasticsearch which is a ditributed search and analysis engine is used to store experiment results obtained and the results are visualised on the Kibana (dashboard).The experiment tools installed on the virtual machine include Jupyter Lab, which is used to write the Python algorithms for analysis during the experiment, and Elasticsearch, a distributed search and analysis engine, which is used to store the experiment results. The results are then visualised on the Kibana dashboard.
Virtual Machine 2 (VM 2, Windows Machine)	Cuckoo Sandbox is a sandbox environment used for malware analysis and containment, designed to prevent outbreaks. It provides an analysis report on a given malware sample based on detection rules such as YARA (a tool commonly used in malware research and detection) to determine whether the sample is benign or malicious.

Table II describes the tools used in the malware analysis environment, which has been built according to the proposed APT malware analysis approach. This environment can be customised by replacing the existing data feeds, tools, or software with alternatives, as long as they offer the functionality outlined in the proposed approach. The tools employed in the development of the malware analysis environment are open-source and available for free download from their respective websites. Additionally, scripting is required to integrate the tools and perform tasks that require specific libraries or unique functionality. For instance, Python scripts are used to execute functions that are not provided by the selected technology providers.

B. Malware Analysis Experiment Design

The flow of the malware analysis experiment follows the enhanced malware analysis approach, which consists of six stages: data extraction from files, data preparation, data processing, data serialisation, information presentation, and information sharing. In the experiment using VM1, static analysis is performed to extract malware features such as strings and the import address table. These extracted features are then used for similarity comparison, which is conducted using the Jaccard Index, as shown in Fig. 3.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Fig. 3. Jaccard index.

Fig. 3 shows the Jaccard Index, which is used to gauge the similarity of sample sets by measuring the ratio of the intersection to the union of the two sets (Yang et al., 2023). The Jaccard Index between two sets is calculated by dividing the number of elements in their intersection by the number of elements in their union. The value of the Jaccard Index, denoted as J(A,B), lies between 0 and 1, where $0 \leq J(A,B) \leq 1$. For example, if the intersection of sets A and B is empty, then J(A,B)=0, indicating no similarity between the two sets of assembly code [28]. This is one of the similarity comparison techniques identified for assessing similarities in malware attributes.

Dataset 1, sourced from the GitHub link mentioned in Table II, contains 3,594 APT malware samples. During the data preparation step in the CTI data collector component, 2,887 PE files are filtered out of the 3,594 APT malware samples in the dataset. For Dataset 2, which is pulled from Vxunderground, 596 PE files from 2023 and 3,446 PE files from 2022 are filtered during the same data preparation step in the CTI data collector component.

Two attributes are selected from the observations during the data preparation step. These chosen attributes—strings and the Import Address Table (IAT)—are analysed during the experiment to identify similarities in APT malware. Based on these two attributes, three experiment scenarios are designed to help security professionals, researchers, and organisations understand how to identify similarities between different malware samples. Table III describes the role of these attributes and outlines the execution of the experiment scenarios.

TABLE III. EXPERIMENT SCENARIO IMPLEMENTATION

Experiment Scenarios	Implementation
String comparison	Strings are ASCII and Unicode printable sequences of characters embedded within a file. The strings attribute refers to the human-readable text embedded within a binary file, often revealing useful information such as URLs, file paths, error messages, and even internal function names. Malware analysts frequently examine these strings to gain insights from the malware, such as its behaviour and command-and-control (C2) information. The string comparison scenario is an experiment designed to analyse APT malware and calculate the Jaccard Index between combinations of the 2,888 PE files identified. In this scenario, only Dataset 1 is used to identify the similarity between two distinct samples based on string attribute.
IAT Comparison	The Import Address Table (IAT) is part of a Windows module—either an executable or a dynamic link library (DLL)—that records the addresses of functions imported from other DLLs. The IAT provides insight into the specific system resources and APIs used by the malware, which is valuable for identifying patterns across different samples.. The IAT comparison scenario is an experiment designed to analyse APT malware and calculate the Jaccard Index between combinations of the 2,888 PE files identified. In this scenario, only Dataset 1 is used to identify the similarity between two distinct samples based on the IAT attribute.
Similarity Comparison on Dataset 1 and Dataset 2	In this experiment scenario, the strings and IAT attributes are extracted from APT malware in both Dataset 1 and Dataset 2. The similarity comparison is then performed by calculating the Jaccard Index between the APT malware samples in Dataset 1 and Dataset 2 based on these two extracted attributes. The similarity comparison is performed to identify any correlation between the two different APT malware datasets based on the string and IAT attributes. The comparison between Dataset 1 and Dataset 2 begins with the year 2023, followed by the year 2022.

Table III illustrates the experiment flow, which includes three scenarios. These experiment scenarios are designed based on the dataset and the chosen attribute similarity comparisons, which include strings and IAT, as described in detail. The scenarios are executed to demonstrate how the normalised data are analysed and represented in graphs and tables. The sample data collected are shown in Table IV.

TABLE IV. SAMPLE DATA COLLECTED

Malware Hash	Malware Label	String Attributes	IAT Attributes
00be6858156b0be404b4fa4852ffc550c25565236beaa4cb13ffe288bcb48d8e	APT 1	Syntax error! Usage: getf/putf FileName <N> Mozilla/5.0 So long! exit Shell started,wait to terminate it..... Service is running already! Service started! StartService failed! CreateProcess failed! Program started! Syntax error!	CloseServiceHandle ControlService CreateFileA CreatePipe CreateProcessA'", CreateProcessAsUserA CreateThread CreateToolhelp32Snapshot

		Usage: start </p|/s> <filename|ServiceName>	

Table IV lists sample data extracted from the APT malware. The fields identified for analysis, which are relevant to the experiment, include Hash, Label, Strings, and IAT. The comparison is then performed by calculating the Jaccard index values for the Strings and IAT attributes. This comparison, using the Jaccard index, is based on the bag of features concept, as shown in Fig. 4.

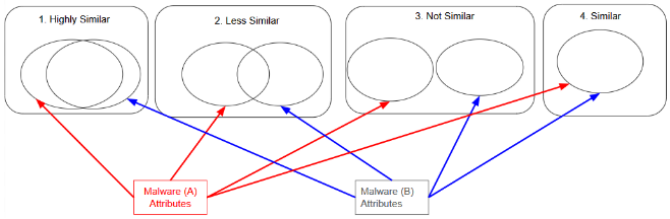

Fig. 4. Malware attributes similarity concept.

Fig. 4 illustrates the concept of attribute similarity used in malware analysis, where the similarity of malware attributes is calculated using a similarity coefficient, such as the Jaccard index. Based on the threshold of 0.8 for Jaccard index calculation: Diagram 1 represents an instance where the calculated Jaccard index value is greater than 0.8; Diagram 2 shows an instance where the Jaccard index value is less than 0.8; Diagram 3 depicts an instance where the Jaccard index value is 0. Finally, Diagram 4 illustrates an instance where the malware attributes exactly match, and the Jaccard index value is 1.

IV. Malware Analysis Results

Experiments are conducted to demonstrate the results obtained from the implementation of the enhanced approach. The results are based on three scenarios, using the malware attribute similarity concept described in Section 3B. Two distinct samples, identified by hash value, are obtained from the malware dataset and labelled as Malware 1 and Malware 2. For each comparison performed based on the designed scenario, the Jaccard Index value is calculated to identify the similarity between the compared samples.

A. Strings Similarity Comparison

The string comparison scenario in the experiment compares the string attributes of each APT malware sample. For each APT malware analysed, the string values are extracted and saved in a CSV file. A sample of the results from the similarity comparison of string attributes extracted from APT malware is presented in Table V.

TABLE V. Sample Jaccard Index Results for Strings Attribute

Malware 1 Hash	Malware 1 Label	Malware 2 Hash	Malware 2 Label	String Jaccard Index
0fbb47373b8bbefdfd9377dc2	AP	0fbb47373b8bbefdfd9377dc2	AP	1
6b6418d2738e6f688562885f4d2a1a049e4948e	T1	6b6418d2738e6f688562885f4d2a1a049e4948e copy	T1	
6c8eb3365b7fb7683b9b465817e5cb87574026e306c700f3d103eba056777720	AP T29	6c8eb3365b7fb7683b9b465817e5cb87574026e306c700f3d103eba056777720	AP T29	1

Table V shows sample Jaccard index results for the string attribute. The first row presents a sample where the Jaccard index value for the string attribute is 1. This result indicates that the malware is identical, as it represents the same sample with a similar hash value.

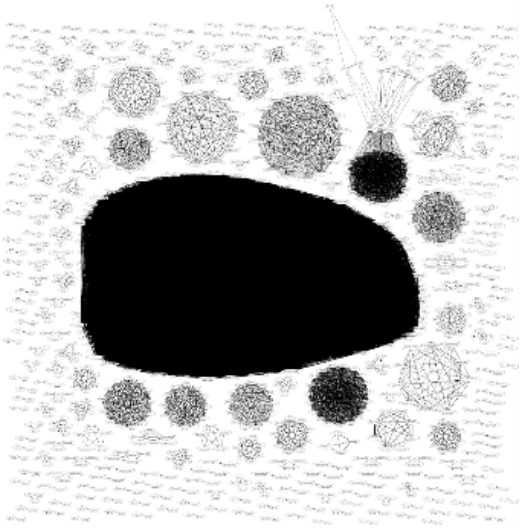

Fig. 5. Strings network graph.

Fig. 5 shows the results of the string attribute comparison between malware from different APT groups. Nodes without at least one edge are removed from the graph to reduce clutter.

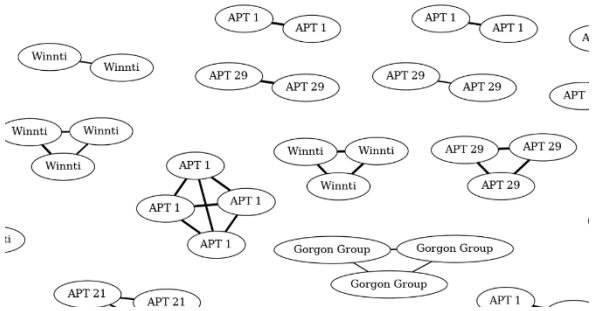

Fig. 6. Close up of strings network graph.

Fig. 6 provides a close-up of Fig. 5. The APT malware samples are grouped based on the string attribute. Observations of the results reveal that some malware, even from the same APT group, are not clustered together. Additionally, no malware from different APT groups shows any connections.

B. Imports Address Table Similarity Comparison

The IAT comparison scenario in the experiment compares the IAT attributes of each APT malware sample. For each APT malware analysed, the IAT values are extracted and saved in a CSV file. A sample of the results from the similarity comparison of IAT attributes extracted from APT malware is presented in Table VI.

TABLE VI. SAMPLE JACCARD INDEX RESULTS FOR IAT ATTRIBUTE

Malware 1 Hash	Malware 1 Label	Malware 2 Hash	Malware 2 Label	IAT Jaccard Index
0fbb47373b8bbefdfd9377dc26b6418d2738e6f688562885f4d2a1a049e4948e	APT1	0fbb47373b8bbefdfd9377dc26b6418d2738e6f68856285f4d2a1a049e4948e	APT1	1
1b3ee0274ae0ac0b83dba7f95f00e2381a5d3596d136eb1fac842a07d8d25262	APT1	6bb764f3a5ca57f9bcc72aa0c34dab64e870e22c6400f6b3f62d5986104dc68f	APT1	0.82828282828 2828
6c7e768e48b9b225b7b9f84528c53c2e6f9b639ce2e7919fe0dff9aad07ea4f5	APT29	6c8eb3365b7fb7683b9b465817e5cb87574026e306c700f3d103eba056777720	APT29	0.94845360824 7423
6c8eb3365b7fb7683b9b465817e5cb87574026e306c700f3d103eba056777720	APT29	6c8eb3365b7fb7683b9b465817e5cb87574026e306c700f3d103eba056777720	APT29	1

Table VI shows two identical samples in rows 1 and 4, which are malware from the same APT group with a Jaccard Index value of 1. Rows 2 and 3 show malware with different hashes but from the same APT group, with Jaccard Index values of 0.83 and 0.95, respectively. The closer the Jaccard Index values are to 1, the greater the similarity in the IAT attributes.

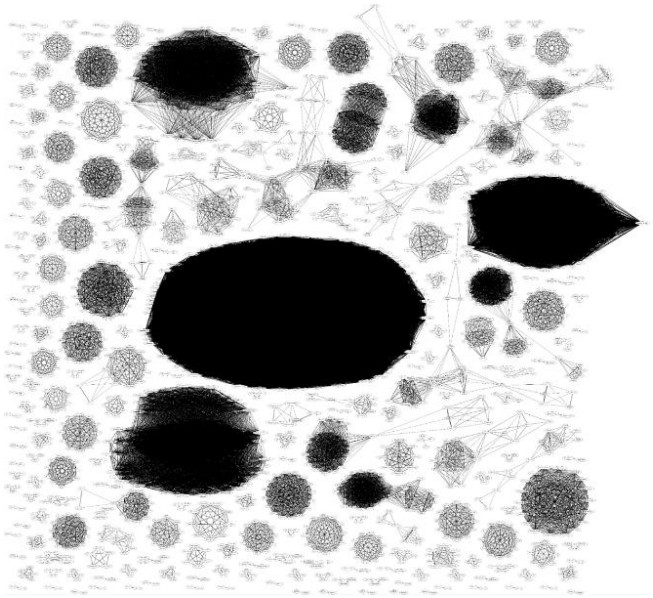

Fig. 7. IAT Network graph.

Fig. 7 shows the results of the IAT comparison between malware from different APT groups. The IAT network graph differs from the string network graph shown in Fig. 5. A close-up of the IAT network graph is presented in Fig. 8.

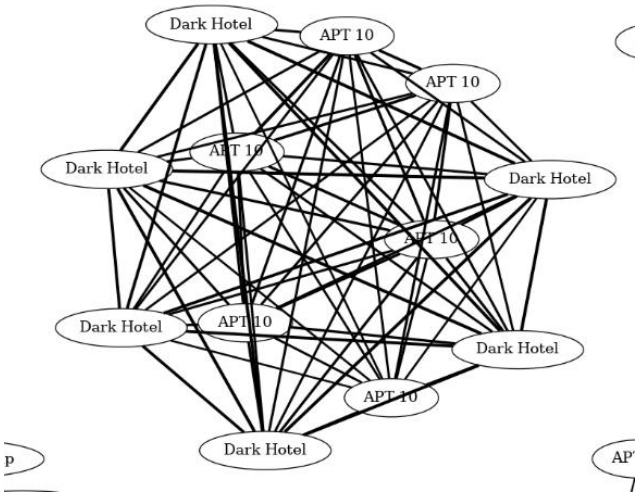

Fig. 8. Close up of IAT network graph.

In contrast to Fig. 6 from the string comparison experiment, Fig. 8 shows that in the IAT network graph, there are overlaps in IAT values for malware from different APT groups. This example demonstrates that there are overlapping attributes between malware from two different APT groups, which are grouped based on attributed threat actors.

C. Strings and Import Address Table Similarity Comparison

The similarity comparison scenario compares features from Dataset 2 (2023) and Dataset 2 (2022) with features collected from APT malware in Dataset 1. In this scenario, the strings and IAT attributes extracted from Dataset 2 are compared with the corresponding strings and IAT attributes extracted from malware samples of 12 APT groups in Dataset 1.

The similarity comparison of 596 samples from Dataset 2 (2023) with 2,887 samples from Dataset 1 took 24 minutes and 19 seconds, resulting in 1,720,652 similarity comparisons. A summary of the results is shown in Table VII.

TABLE VII. SUMMARY OF SIMILARITY COMPARISON RESULTS FOR DATASET 1 AND DATASET 2 (2023)

Similarity Comparison	Greater than 0.8	Greater than 0.5	Lower than 0.5	Lower than 0.2
Strings	0	121	1,720,531	1,715,434
IAT	9643	33673	1,684,723	1,472,266

The results of the string similarity comparison show that no Jaccard Index value exceeds 0.8. However, further examination of the results reveals that 28 samples have a Jaccard Index value greater than 0.6, with some samples exceeding 0.5. The outcome of the string similarity comparison is presented in Fig. 9.

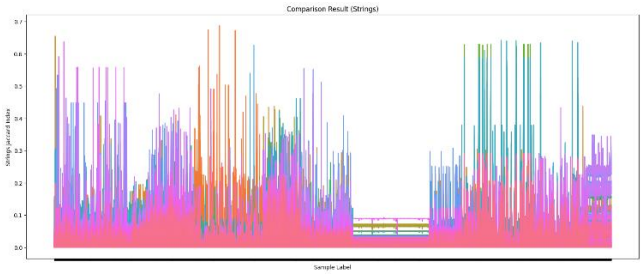

Fig. 9. Similarity comparison result line graph (Strings).

Fig. 9 provides an overview of the string similarity comparison results. The graph shows that, out of the 1,720,652 similarity comparisons performed, none of the Jaccard index values exceed 0.7. Since the Jaccard Index values in the results are below the set threshold, it is likely that there are no significant similarities between the malware in Dataset 2 (2023) and Dataset 1.

A different result was obtained for the IAT similarity comparison, with 9,643 samples having a Jaccard Index value greater than 0.8, as shown in Table VII and represented in Fig. 10.

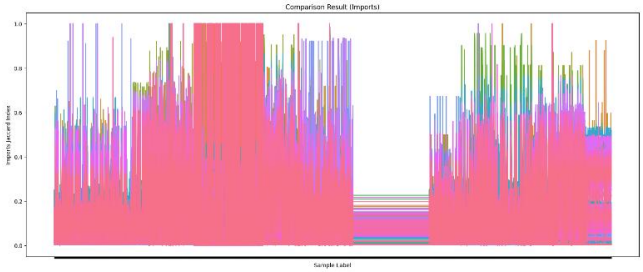

Fig. 10. Similarity comparison result line graph (IAT).

Fig. 10 provides an overview of the 1,720,652 IAT similarity comparison results. A correlation of 9,643 samples that scored a Jaccard Index higher than 0.8 is highlighted in the IAT network graph shown in Fig. 11.

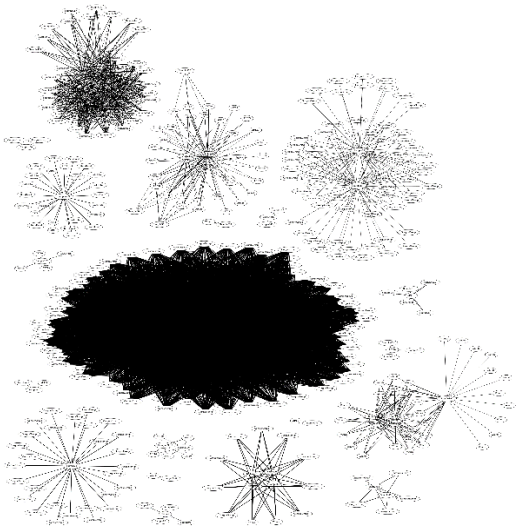

Fig. 11. IAT network graph.

Fig. 11 depicts the results of the IAT comparison between Dataset 2 (2023) and Dataset 1. The IAT network graph shows multiple correlations. One of the correlations identified is shown in Fig. 12, which provides a close-up of the IAT network graph from Fig. 11.

Fig. 12. Close up IAT network graph.

Fig. 12 shows that Sample 160, Sample 161, and Sample 162 are correlated with malware attributed to APT 10 and APT 28, based on the IAT similarity comparison. Additionally, Sample 368 and Sample 371 are correlated with malware identified as being used in APT 28 malicious operations. This suggests that, since the results scored higher than 0.8, the IAT attributes for these samples overlap with those of APT 10 and APT 28 malware.

The experiment continued with the similarity comparison of 3,446 samples from Dataset 2 (2022) with 2,887 samples from Dataset 1, which took 2 hours, 52 minutes, and 45 seconds. This resulted in 9,948,602 similarity comparisons. A summary of the results is shown in Table VIII.

TABLE VIII. SUMMARY OF SIMILARITY COMPARISON RESULTS FOR DATASET 1 AND DATASET 2 (2022)

Similarity Comparison	Greater than 0.8	Greater than 0.5	Lower than 0.5	Lower than 0.2
Strings	4	299	9,948,301	9,901,641
IAT	48,110	196,401	9,741,018	8,380,710

Table VIII shows that for the string similarity comparison, 299 samples scored above 0.5, and four samples scored above 0.8. The results of the string similarity comparison are represented in Fig. 13.

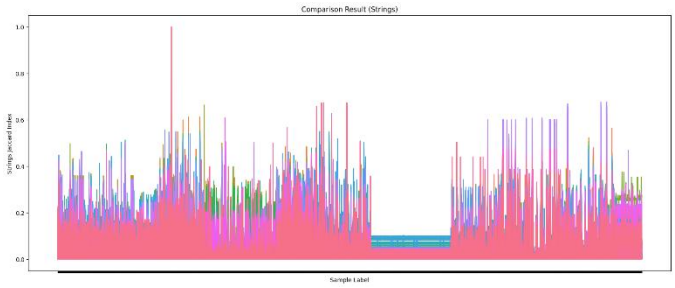

Fig. 13. Similarity comparison result line graph (Strings).

Based on Fig. 13, it is clear that there is a single instance where the Jaccard Index value is 1.0. The results of the line graph for the IAT similarity comparison are shown in Fig. 14.

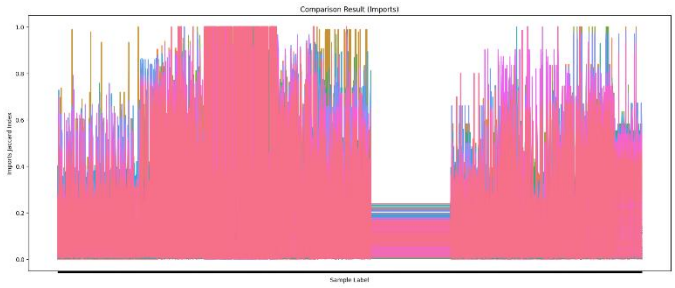

Fig. 14. Similarity comparison result line graph (IAT).

Fig. 14 depicts the visualisation of the results for 48,110 samples with Jaccard Index values greater than 0.8 in the IAT similarity comparison. We then further investigate the results from Table VIII and Fig. 13 to better understand the findings. The 4 samples with Jaccard Index values greater than 0.8 for the string similarity comparison are filtered from the others. The results for the string similarity comparison above 0.8 are shown in Table IX.

TABLE IX. STRINGS SIMILARITY COMPARISON RESULT FOR DATASET 1 AND DATASET 2 (2022)

Sample Hash	Sample Label	Malware Hash	Malware Label	String Jaccard Index	IAT Jaccard Index
c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	Sample243	c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	APT10	1.0	1.0
fa7eee6e322bfad1bb0487aa1275077d334f5681f0b4ede0ee784c0ec1567e01	Sample809	c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	APT10	1.0	1.0
c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	Sample3063	c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	APT10	1.0	1.0
c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	Sample3427	c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	APT10	1.0	1.0

Table IX lists the four samples for which the Jaccard Index values for strings are greater than 0.8. Observations from both the string and IAT similarity comparisons show that the Jaccard Index obtained is high, with a score of 1.0. This indicates that Sample 243, Sample 809, Sample 3063, and Sample 3427 exactly match APT10 malware, which has the hash"c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e". This correlation is also reflected in the strings network graph shown in Fig. 15.

Fig. 15. Correlation identified in strings and imports network graph.

Based on the correlation shown in Fig. 15, a relationship between APT 10 malware and Sample 809, which has the hash "fa7eee6e322bfad1bb0487aa1275077d334f5681f0b4ede0ee784c0ec1567e01," is identified. Since Sample 243, Sample 3063, and Sample 3427 have the same hash as the APT 10 malware, they are not shown in the network graph in Fig. 15. This also explains the result obtained in Fig. 13, which indicates the similarity between APT 10 malware and those samples. Therefore, only the correlation of Sample 809 to APT 10 malware, based on the string similarity comparison, is shown, as only Sample 809 has a different hash. This suggests that the incident involving Sample 243, Sample 809, Sample 3063, and Sample 3427 is likely linked to APT 10, since these samples share similar attributes with malware already attributed to this APT group, which is believed to be linked to China.

Based on the information obtained, these findings can be used for CTI (Cyber Threat Intelligence) purposes. The report accompanying Dataset 2 from Vxunderground states that Sample 243 is Nbtscan, discovered by Avast; Sample 809 is NBTScan, discovered by Symantec; Sample 3063 is a NetBIOS scanner, discovered by Trend Micro; and Sample 3427 is F01A9A2D1E31332ED36C1A4D2839F412, discovered by Kaspersky, where only the MD5 value is provided in the IOC section. All of these reports attribute the samples to APT groups possibly linked to China, such as Mustang Panda and Earth Lusca. Therefore, the analysis and results obtained in this research provide a valid correlation.

By knowing that the malware found is possibly attributed to a specific APT group, organisations can better prepare to defend against the threat. For example, based on our results, if the sample found in an organisation is linked to APT 10, threat hunting efforts can focus on looking for IOCs (Indicators of Compromise) and activities associated with APT 10 or related APT groups, based on past incidents involving those groups. Our research demonstrates how string and IAT attributes can be used in similarity comparison scenarios, with extracted features being correlated to the threat actor through visual information presentation.

V. CONCLUSION

In general, the enhanced APT malware analysis approach extracts attributes from PE files and uses these to correlate with threat actors. This helps identify the origin of malware through the Jaccard Index, a similarity comparison technique used to establish threat actor correlations. The information

obtained can be leveraged to develop countermeasures against cyber threats. The development of additional malware analysis systems and experiments performed in this research includes technical discussions and examples to deepen understanding of the formulated APT malware analysis approach. This solution aims to assist cybersecurity practitioners and researchers in making informed decisions by providing actionable insights and a comprehensive perspective on cyber-attacks, based on the analysis of artefacts from APT groups.

Our experiment identified correlations between four samples and malware attributed to APT 10. Our analysis of the results also validates the findings obtained during the experiment. The enhanced APT malware analysis approach, the malware analysis environment, and the experimental scenarios developed in this research provide a foundation for discovering threat actor correlations. Our work provides a foundation for correlating malware with the threat actor, and the malware analysis approach can be used in designing other experimental scenarios.

Extending the experimental scenarios is one possible avenue for future work. Developing additional scenarios would uncover more insights from the APT malware dataset. Additionally, the malware analysis environment could be improved by using hardware with higher specifications, which would enable faster analysis, and by incorporating tools that are more preferred or offer better functionality.

Another direction for future work is refining the dataset used, or adopting a different dataset that is better suited to the experiment. Our research used Dataset 1, which, although relatively outdated, is well-structured, making it easier to label samples with the attributed threat actor. Moving forward, we plan to use Dataset 2, which is more recent but requires additional effort for labeling. If publicly available datasets were better structured or properly labeled, the sample analysis process would be much easier, and the labeling step would be significantly streamlined.

Apart from that, other similarity comparison techniques could be explored for future work, incorporating AI—such as machine learning or deep learning algorithms—into the approach. Our current work, using the Jaccard Similarity Index, aims to conduct a preliminary analysis of the dataset and obtain results that will help develop the malware analysis approach with similarity comparison techniques, as well as design the malware analysis environment. Both the enhanced approach and the malware analysis environment are integral for analysing APT malware to extract information for identifying the threat actor.

ACKNOWLEDGMENT

This research was fully funded by the Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (5F511) awarded to Suriayati Chuprat, Universiti Teknologi Malaysia.

REFERENCES

[1] CrowdStrike, “CrowdStrike 2023 Global Threat Report.” Accessed: May 27, 2023. [Online]. Available: https://www.crowdstrike.com/global-threat-report/

[2] S. Cobb and A. Lee, “Malware is called malicious for a reason: The risks of weaponizing code,” Int. Conf. Cyber Conflict, CYCON, vol. 2014, pp. 71–84, 2014, doi: 10.1109/CYCON.2014.6916396.

[3] C. Rong, G. Gou, C. Hou, Z. Li, G. Xiong, and L. Guo, “UMVD-FSL: Unseen Malware Variants Detection Using Few-Shot Learning,” Proc. Int. Jt. Conf. Neural Networks, vol. 2021-July, pp. 9–16, 2021, doi: 10.1109/IJCNN52387.2021.9533759.

[4] L. M. Zagi and B. Aziz, “Searching for malware dataset: A systematic literature review,” 2020 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2020 - Proc., pp. 375–380, 2020, doi: 10.1109/ICITSI50517.2020.9264929.

[5] A. Amamra, C. Talhi, and J. M. Robert, “Smartphone malware detection: From a survey towards taxonomy,” Proc. 2012 7th Int. Conf. Malicious Unwanted Software, Malware 2012, pp. 79–86, 2012, doi: 10.1109/MALWARE.2012.6461012.

[6] R. Nigam, R. K. Pathak, A. Kumar, and S. Prakash, “PCP Framework to Expose Malware in Devices,” Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020, pp. 651–654, 2020, doi: 10.1109/ICESC48915.2020.9155593.

[7] I. Bello et al., “Detecting ransomware attacks using intelligent algorithms: recent development and next direction from deep learning and big data perspectives,” J. Ambient Intell. Humaniz. Comput., vol. 12, no. 9, pp. 8699–8717, 2021, doi: 10.1007/s12652-020-02630-7.

[8] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, “Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions,” Appl. Sci., vol. 12, no. 1, 2022, doi: 10.3390/app12010172.

[9] V. Mavroeidis and J. Brule, “A nonproprietary language for the command and control of cyber defenses – OpenC2,” Comput. Secur., vol. 97, p. 101999, 2020, doi: 10.1016/j.cose.2020.101999.

[10] F. Martinelli, F. Mercaldo, V. Nardone, A. Santone, A. K. Sangaiah, and A. Cimitile, “Evaluating model checking for cyber threats code obfuscation identification,” J. Parallel Distrib. Comput., vol. 119, pp. 203–218, 2018, doi: 10.1016/j.jpdc.2018.04.008.

[11] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, “A Cyber-Kill-Chain based taxonomy of crypto-ransomware features,” J. Comput. Virol. Hacking Tech., vol. 15, no. 4, pp. 277–305, 2019, doi: 10.1007/s11416-019-00338-7.

[12] Y. Roumani, “Detection time of data breaches,” Comput. Secur., vol. 112, p. 102508, 2022, doi: 10.1016/j.cose.2021.102508.

[13] ENISA, ENISA Threat Landscape 2022, no. November. 2022. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

[14] A. Sharma, B. B. Gupta, A. K. Singh, and V. K. Saraswat, “Orchestration of APT malware evasive manoeuvers employed for eluding anti-virus and sandbox defense,” Comput. Secur., vol. 115, p. 102627, 2022, doi: 10.1016/j.cose.2022.102627.

[15] Y. Zhou and X. Jiang, “Dissecting Android malware: Characterization and evolution,” Proc. - IEEE Symp. Secur. Priv., no. 4, pp. 95–109, 2012, doi: 10.1109/SP.2012.16.

[16] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov, “Learning and Classification of Malware Behavior,” in Detection of Intrusions and Malware, and Vulnerability Assessment, vol. 7591, no. July, U. Flegel, E. Markatos, and W. Robertson, Eds., in Lecture Notes in Computer Science, vol. 7591. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 108–125. doi: 10.1007/978-3-540-70542-0_6.

[17] C. P. Chenet, A. Savino, and S. Di Carlo, "A Survey on Hardware-Based Malware Detection Approaches," IEEE Access, vol. 12, pp. 54115–54128, 2024, doi: 10.1109/ACCESS.2024.3388716.

[18] T. Lee, I. Ahl, and D. Hanzlik, "Breaking Down the China Chopper Web Shell - Part I." Accessed: Jan. 17, 2024. [Online]. Available: https://www.mandiant.com/resources/blog/breaking-down-china-chopper-web-shell-part-i

[19] TONY LEE, IAN AHL, and DENNIS HANZLIK, "Breaking Down the China Chopper Web Shell - Part II | Mandiant," Mandiant, 2013, [Online]. Available: https://www.mandiant.com/resources/blog/breaking-down-the-china-chopper-web-shell-part-ii

[20] L. Rochberger, T. Fakterman, and R. Falcone, "Unit 42 Researchers Discover Multiple Espionage Operations Targeting Southeast Asian Government." [Online]. Available: https://unit42.paloaltonetworks.com/analysis-of-three-attack-clusters-in-se-asia/

[21] M. Figueroa, "Building a Custom Malware Analysis Lab Environment." [Online]. Available: https://www.sentinelone.com/labs/building-a-custom-malware-analysis-lab-environment/

[22] S. Torabi, M. Dib, E. Bou-Harb, C. Assi, and M. Debbabi, "A Strings-Based Similarity Analysis Approach for Characterizing IoT Malware and Inferring Their Underlying Relationships," IEEE Netw. Lett., vol. 3, no. 3, pp. 161–165, 2021, doi: 10.1109/lnet.2021.3076600.

[23] N. Xu, S. Li, X. Wu, W. Han, and X. Luo, "An APT Malware Classification Method Based on Adaboost Feature Selection and LightGBM," in Proceedings - 2021 IEEE 6th International Conference on Data Science in Cyberspace, DSC 2021, 2021. doi: 10.1109/DSC53577.2021.00101.

[24] Y. H. F. Hu and C. C. G. Hsieh, "A Study of Classifying Advanced Persistent Threats with Multi-Layered Deep Learning Approaches," in 19th IEEE International Symposium on Parallel and Distributed Processing with Applications, 11th IEEE International Conference on Big Data and Cloud Computing, 14th IEEE International Conference on Social Computing and Networking and 11th IEEE Internation, 2021. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00220.

[25] X. Han, C. Li, X. Li, and T. Lu, "Research on APT Attack Detection Technology Based on DenseNet Convolutional Neural Network," in Proceedings - 2021 International Conference on Computer Information Science and Artificial Intelligence, CISAI 2021, 2021. doi: 10.1109/CISAI54367.2021.00091.

[26] C. Do Xuan and D. Huong, "A new approach for APT malware detection based on deep graph network for endpoint systems," Appl. Intell., 2022, doi: 10.1007/s10489-021-03138-z.

[27] M. Piskozub, F. De Gaspari, F. Barr-Smith, L. Mancini, and I. Martinovic, "MalPhase: Fine-Grained Malware Detection Using Network Flow Data," in ASIA CCS 2021 - Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021. doi: 10.1145/3433210.3453101.

[28] R. Murali, P. Thangavel, and C. Shunmuga Velayutham, "Evolving malware variants as antigens for antivirus systems," Expert Syst. Appl., vol. 226, 2023, doi: 10.1016/j.eswa.2023.120092.

[29] Gray, Jason, Daniele Sgandurra, and Lorenzo Cavallaro. "Identifying authorship style in malicious binaries: techniques, challenges & datasets." *arXiv preprint arXiv:2101.06124*

[30] Albtosh, Luay Bahjat. "Malware authorship attribution: Unmasking the culprits behind malicious software." *World Journal of Advanced Research and Reviews* 23, no. 3 (2024)

[31] Xiang, Xiayu, Hao Liu, Liyi Zeng, Huan Zhang, and Zhaoquan Gu. "IPAttributor: Cyber Attacker Attribution with Threat Intelligence-Enriched Intrusion Data." *Mathematics* 12, no. 9 (2024): 1364