




ArkThor: Threat Categorization Based on Malware's C2 Communication

Mohammed Jawed¹, Sriram Parameshwaran², Nitesh Kumar³^a, Anand Handa³^b
and Sandeep K. Shukla³^c

¹International Atomic Energy Agency (IAEA), Austria

²McAfee India Pvt Ltd, India

³C3i Hub, Indian Institute of Technology, Kanpur, India

Keywords: Threat Categorization, Command-and-Control(C2) Communication, .Pcap Files, Network Security, Threat Detection, Threat Mitigation, Machine Learning, RabbitMQ, User Interface, APIs, SQLite Database, Containerization, Scapy, Python, Rule-Engine.

Abstract: In today's digital world, network security is of utmost importance. Cyber-attacks are becoming more sophisticated and complex, making it increasingly difficult to detect and prevent them. Command-and-Control (C2) communication is a common technique used by attackers to control infected hosts and steal sensitive information. Therefore, it is crucial to identify and categorize network threats accurately to prevent and mitigate cyber-attacks. However, traditional methods of threat categorization are often insufficient in identifying and classifying these communications. This work aims to develop a threat categorization tool based on C2 communication in archived/live stream .pcap files that can help organizations more effectively detect and respond to cyber threats. The resulting tool, ArkThor, represents safety and strength and is a cutting-edge threat categorization engine designed to empower organizations to stay ahead of emerging threats in the cybersecurity landscape.


1 INTRODUCTION


ArkThor is an innovative threat categorization tool that offers a range of advanced features designed to improve threat detection capabilities and enhance overall cybersecurity infrastructure. ArkThor is available for free on Github (Handa and Kumar, 2023) and DockerHub. Here are some of the key features and contributions of ArkThor:


- **Threat Categorization Based on C2 Communication:** ArkThor is designed to categorize threats based on Command-and-Control (C2) communication in archived or live stream .pcap files. This allows the system to categorize the network threat into various categories, including BOK-BOT (Acronis, 2023), IcedID (Checkpoint, 2023), Graftor (F-secure, 2023), STRRat (Blackberry, 2023), Cobalt Strike (Malwarebytes, 2023), and more. The core engine of ArkThor consists of

three independent modules - packet processing, rule parser, and rule authoring - that work together to provide a comprehensive threat categorization engine. This approach provides organizations with a more accurate and effective way to detect and respond to cyber threats.

- **3-Distinct Layers:** ArkThor is built with three distinct layers that enable organizations to identify and mitigate threats before they cause damage. The presentation layer provides end-users with a comprehensive set of information, enabling them to gain valuable insights into the threat landscape and take proactive measures to prevent and mitigate cyber-attacks. The middle layer consists of APIs, a database, and a message broker that work together to connect the outside world to the core layer. Finally, the core layer includes three independent modules, namely packet processing, rule parser, and rule authoring, that analyze the .pcap file based on C2 communication.
- **User-Friendly Interface:** ArkThor includes a user-friendly interface that provides a simple and

^a <https://orcid.org/0000-0003-0998-0925>

^b <https://orcid.org/0000-0003-0075-1165>

^c <https://orcid.org/0000-0001-5525-7426>

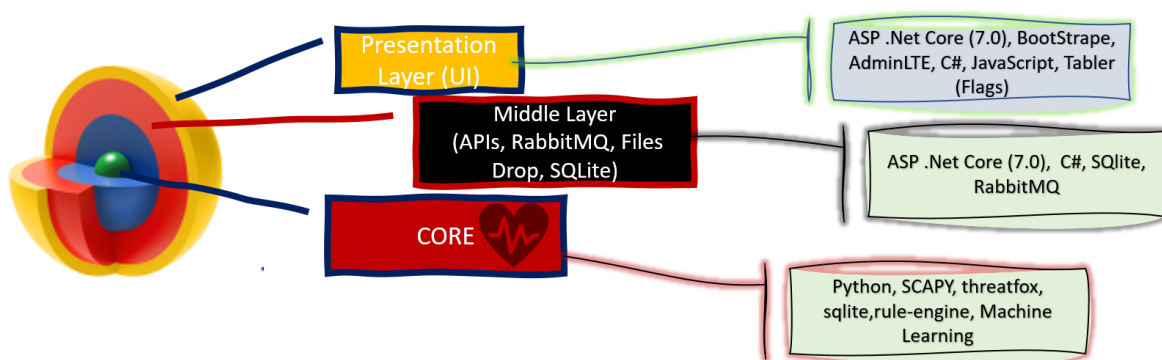


Figure 1: ArkThor Layers.

intuitive way for users to interact with the system. The interface offers various views and visualizations that enable users to quickly identify threats and take appropriate action.

- **Flexible and Scalable:** ArkThor includes various APIs that provide flexibility and scalability to organizations of all sizes. The system can be easily integrated into existing infrastructures, allowing organizations to customize the tool to meet their specific needs.
- **Containerization and Microservice Architecture:** ArkThor is built with containerization and microservice architecture that enables it to be easily deployed and used as a plug-and-analyze solution in any organization. The containerization also allows the system to be easily updated and maintained, ensuring that it remains up-to-date and effective.
- **Machine Learning Integration:** In the future, ArkThor aims to integrate a machine learning model to train the core module, further enhancing the threat categorization engine’s accuracy and effectiveness. This feature will enable the system to learn from past incidents and adapt to new threats in real-time.

Our Contributions: The contributions of ArkThor to the field of cybersecurity are significant. The tool provides organizations with an advanced system for improving their threat detection capabilities and overall security posture. By categorizing threats based on C2 communication, ArkThor enables organizations to better understand the nature of the threat and take appropriate action. Additionally, the system’s flexibility and scalability make it a valuable tool for organizations of all sizes, from small businesses to large enterprises.

2 METHODOLOGY

The ArkThor product is build using the following methodology and steps:

Core Engine Development: The first step in the development process is to create the core engine. This involves designing and building three independent modules – packet processing, rule parser, and rule authoring. The packet processing module uses Scapy (Scapy, 2023), an opensource library, to process packets. The rule parser module loads the ArkThor format rules and matches them with the output of the packet processing module. The rule authoring module contains rule components that can convert open-source rules or human-authored rules.

UI Development: The user interface (UI) of the product is built using ASP.NET Core, Javascript and Bootstrap. The UI includes a dashboard, various pages for displaying analysis results, and a measurement page that shows valuable KPIs in the form of pie charts, bar graphs, and knobs.

API Development: To provide flexibility and scalability, we use various APIs. These APIs allow users to interact with the product programmatically and access the data in the product’s SQLite database.

Database Management: We use a SQLite database to store analysis records, configuration settings, and other data. The database is managed using SQLite commands and queries.

Message Queue Integration: We use RabbitMQ (RabbitMQ, 2023), a message queue system, to handle communication between different modules and components. This allows for efficient and reliable communication between different parts of the product.

Containerization: We containerize our solution using Docker, making it easy to deploy and use in any organization.

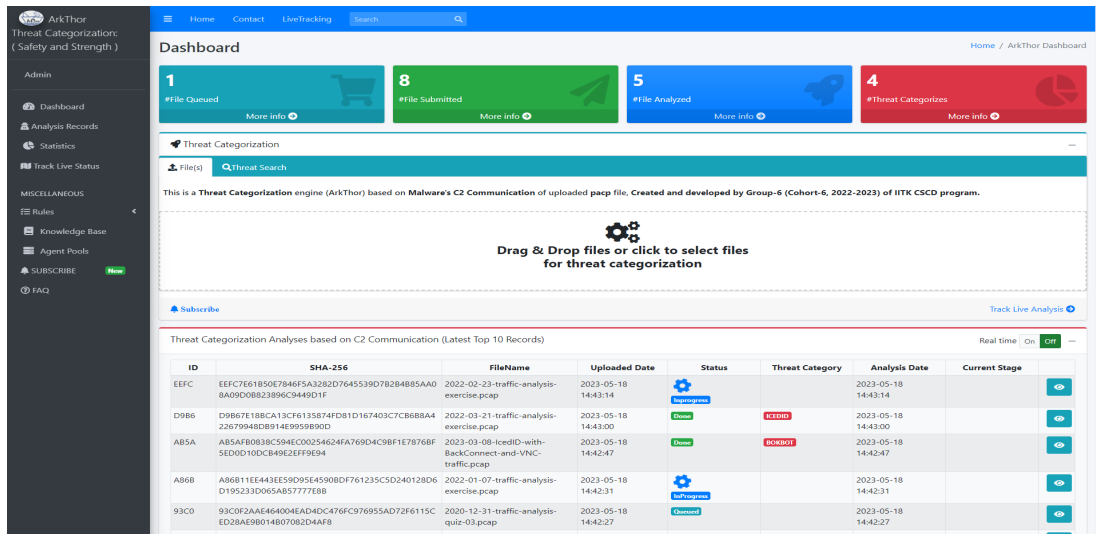


Figure 2: ArkThor Dashboard.

Testing and Deployment: We test ArkThor to ensure that it meets the requirements and works as expected. It is then deployed to the production environment using Docker, which allows for easy deployment and management of the product.

Machine Learning: In the future, we will collect a dataset of labeled network traffic that includes various types of threats by using public datasets and creating our own by collecting traffic from the C3i IIT Kanpur (CSE Department IIT Kanpur, 2023) network and labeling it based on the threat type. After obtaining the labeled dataset, we will train a machine learning model using one of the available algorithms, such as decision trees, random forests, or neural networks. The trained model will be integrated into ArkThor's core engine, allowing it to categorize threats using both rule-based and machine learning-based methods.

When a new network packet is captured, ArkThor's packet processing module will extract relevant features from the packet and send it to the rule parser. The rule parser will then apply the predefined rules and the machine learning model to categorize the threat. The categorization result will be passed to the APIs and then to the user interface, where it can be displayed and analyzed.

3 FEATURES AND FUNCTIONALITIES OF ArkThor UI

The ArkThor user interface comprises several noteworthy features and functionalities, including:

3.1 Dashboard

The ArkThor product has a dashboard as the primary interface, providing users with an overview of their measurements, such as the number of files queued, the total number of files submitted for threat categorization, the number of files analyzed by the ArkThor core engine, and the number of distinct threats categorized by the engine.

Additionally, users can search through the internal database for threats and upload files for threat categorization using select file as well as the drag-and-drop feature. The dashboard will also display the latest top 10 analysis records, each with unique properties, and users can navigate to view more detailed file analysis information.

The file upload process will involve an internal check against criteria such as file extension, size limit, and file signature, followed by passing the file properties to the ArkThor APIs for storage in an internal SQLite Database and a copy of the file will be created in the drop location and SH256 of the uploaded file will be pushed to RabbitMQ Queue.

The core engine will then pick up the file for threat categorization using custom rules. Additionally, users can switch the table displaying the latest top 10 files analysis track records to real-time mode for auto-updating every 5 minutes using ON/OFF switch. Other functionalities of the ArkThor product can be accessed through the left navigation. Figure 2 shows the ArkThor dashboard.

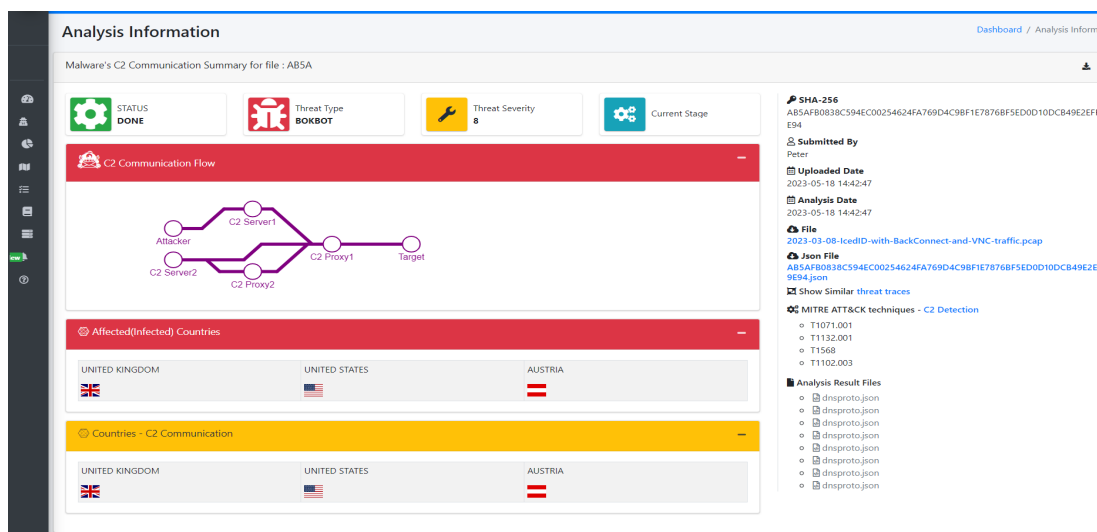


Figure 3: ArkThor File Analysis Information.

3.2 Analysis Information

ArkThor users can access a comprehensive analysis report of the analyzed file from various pages, including the dashboard, records, or ArkThor Live tracking board. To make it easier to understand the information, we divide the webpage into three different sections. Figure 3 shows the insights about ArkThor analysis.

The top section displays critical information about the analyzed file, such as its final status, threat category type, and threat severity. The right section includes informative details such as the SHA-256 of the analyzed file, the user who submitted the file for analysis, the file upload time, the analysis report completion time by the Core Engine, the RAW analyzed file (which can be downloaded by the user), the final JSON analyzed result (available for user to download locally by selecting), and the ability to select similar threat category files from the internal ArkThor database. Additionally, users can view precise information on the MITRE ATT&CK techniques (Mitre, 2023) used by the attacker in C2 communication.

The middle section is divided into three subsections, each containing critical information relevant to the analyzed file. The first subsection, “C2 Communication Flow,” displays all the communication dots on a flow diagram from the attacker to the target. The second subsection, “Affected Countries,” includes a list of country names affected by the output threat category, with respective flags for better presentation. The third subsection displays the names of the countries and their corresponding flags involved in C2 communication.

Furthermore, users can download all this information as a PDF file by selecting the download icon located in the top right corner of the page.

3.3 ArkThor Analysis Records

ArkThor’s analysis records feature offers a powerful functionality for users to access their analyzed records. Figure 4 shows a few of the ArkThor’s analysis records. It can be easily accessed from the left navigation pane of the ArkThor Home page. This feature allows users to search for records based on upload date range, making it simple to locate the records they require. The results are displayed in a convenient tabular format, showcasing essential properties of the records, which helps users quickly identify the relevant records they need. By selecting a record, users can view more detailed information about that specific record. This feature offers a comprehensive view of the analyzed records and streamlines the record retrieval process for maximum efficiency.

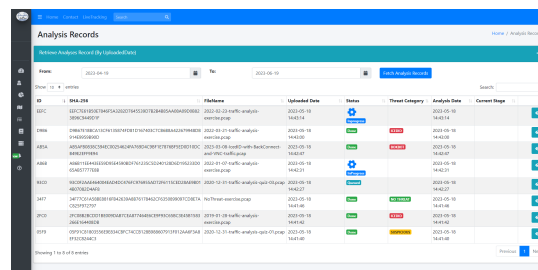


Figure 4: Analysis Records.

3.4 Statistics - Measurement

We develop a statistics or measurement page that shows valuable Key Performance Indicators (KPIs) through the use of pie charts, bar graphs, and knobs. These visual aids help provide insightful and meaningful data to organizations based on the analysis records available in the ArkThor database. By presenting data in a clear and concise manner, users can quickly interpret and identify patterns and trends that can help them make informed decisions. The data available on the statistics or measurement page in the ArkThor product is sourced from the ArkThor APIs, but organizations are not limited to only using these visualizations. If preferred, organizations can utilize other available tools such as Kibana (Elasticsearch, 2023), Grafana (GrafanaLabs, 2023), or any other data visualization platforms to analyze the data available in the ArkThor database. The goal of ArkThor is to provide a comprehensive and flexible cybersecurity solution that can integrate with existing systems and tools, and the inclusion of APIs and the ability to export data is designed to facilitate this flexibility. Therefore, organizations can choose the best approach for their unique needs and use the data as they see fit. Figure 5 depicts various ArkThor's statistics.

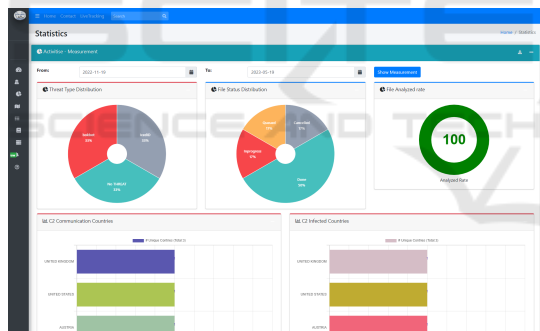


Figure 5: Statistics - Measurements.

3.5 Track Live Status (ArkThor Board)

The ArkThor product also features a dedicated page called the "ArkThor Board," which provides users with valuable insights into the analysis of files using the ArkThor system. This feature was developed with the idea that organizations may want to display all relevant information on a big screen for easy viewing and monitoring. The ArkThor Board is designed to show analysis information in real-time, with automatic refresh feature in every 180 seconds (configurable). To ensure the best visual appearance and usability, the ArkThor Board includes various indicators and visualizations that make it easy for users to quickly interpret and understand the data being pre-

sented, whether on a big screen or a mobile device. The inclusion of these features enhances the overall value and usefulness of the ArkThor product, allowing organizations to stay on top of their cybersecurity posture with ease and efficiency. Additionally, The ArkThor Board also includes information on the current status of different ArkThor Engine's modules. This information helps users to understand whether the system is online or offline and whether any specific modules require attention. Figure 6 depicts the live status using ArkThor Board.



Figure 6: ArkThor Board.

3.6 Subscribe

The subscribe feature in ArkThor product allows users to receive periodic email notifications that contain an executive summary of threat categorization, cyber security, and cyber defense. This feature provides users with valuable information about the current state of cyber threats and helps them stay up-to-date with the latest developments in the field. By subscribing to this feature, users can ensure that they receive timely and relevant information about the latest threats and vulnerabilities, as well as updates to the ArkThor product itself. The email notifications contain a concise summary of the most important information, making it easy for users to quickly scan and digest the information without having to spend a lot of time reading lengthy reports. Users can easily manage their subscription preferences, including the frequency and content of the email notifications they receive. They can choose to receive notifications daily, weekly, or monthly, depending on their preferences and the level of information they require. The subscribe feature is a valuable tool for anyone who wants to stay informed about the latest developments in cyber security and cyber defense. Whether you are an IT professional, a security analyst, or a business owner, this feature provides you with the information you need to protect your systems and stay ahead of emerging threats. Figure 7 shows the ArkThor's subscribe feature.

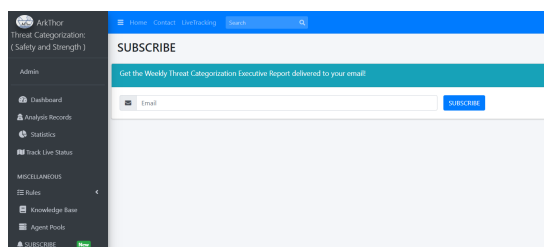


Figure 7: ArkThor subscribe feature.

3.7 Core Control

Users can utilize the ArkThor feature to update the IP2ASN Database of the Core Engine with the most recent IP address information available from the open-source database located at <https://iptoasn.com/>. Additionally, users can employ this feature to retrieve the latest IOC database from threatfox MISP. By simply clicking the refresh button, ArkThor will obtain the most up-to-date IOC details from threatfox and convert them into arkthorule, which will be utilized during file analysis. The feature is accessible under miscellaneous tab and the Figure 8 depicts the feature.

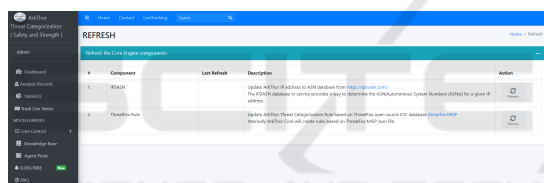


Figure 8: ArkThor Core Control - Refresh feature.

3.8 Core Control-View/Edit Core Config

We introduce the “config.json” file in the CORE engine to enable users to have control over its configuration. This file allows users to customize the behavior of the CORE engine, such as running it as a standalone tool or converting Threatfox IoC (threatfox, 2023) to ArkThor rules within a specified timeframe, among other functionalities.

You can find detailed explanations about these features in the “Deep Dive into ArkThor Core Engine Configuration File” section. To access this feature, navigate to the left navigation pane, go to MISCELLANEOUS, select Core Control, and then choose View/Edit Core Config. After making the necessary edits or updates, click the SAVE button to apply the changes to the CORE engine. Figure 9 shows the feature.

In ArkThor, there are several APIs that have been developed using ASP.NET Core and C#. These APIs

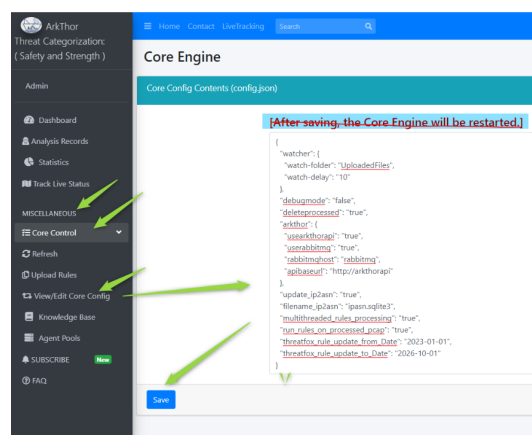


Figure 9: ArkThor Core Config Contents feature.

allow for the receipt of .pcap files, which are then stored in a database. Additionally, a physical copy of the .pcap file is created and stored on a share location. Finally, the SHA256 of the .pcap file is stored on RabbitMQ for CORE engine. The APIs are available in several different types, including POST, GET, and PUT. These APIs are designed to work with ArkThor and include functionality such as uploading JSON results, uploading .pcap files, uploading support files, updating status and threat type, retrieving measurements, creating file records, and more. By leveraging these APIs, users can easily integrate ArkThor into their existing workflows and gain access to its powerful analysis capabilities.

3.9 End-to-End Working

Figure 10 shows the end-to-end working of ArkThor. The following are the steps which explains an end-to-end working of ArkThor:

Step-1: User Uploads .pcap File to the UI – The user selects a .pcap file from their local device and uploads it through the UI.

Step-2: UI Sends the .pcap File to the API – Once the user has uploaded the file, the UI sends it to the API for storage and analysis. The API provides a RESTful endpoint that accepts .pcap files as input.

Step-3: API Saves the .pcap File in SQLite Database and on Local Drive – Upon receiving the .pcap file, the API saves it in a shared directory for later use by CORE Engine. This directory can be specified in the configuration file of the API otherwise by default it will save at same location on API under “UploadedFiles” folder. The API also stores the file in a SQLite database table to keep track of all the files that have been analyzed. This table can contain information such as the file name, file size, upload date, uploaded by, SHA256 hash value, and the

analysis status, C2 communication countries, threat type, severity and much more data related to uploaded file. After saving the .pcap file, the API generates the SHA256 hash value of the file and sends it to RabbitMQ as a message. This message serves as a notification to the core model that a new .pcap file is available for analysis.

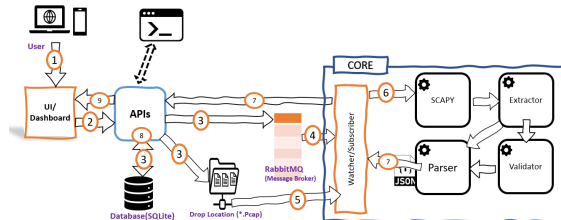


Figure 10: End-to-End Working.

Step-4: Core (Watcher Module) Reads the SHA256 from RabbitMQ – The watcher module is a separate application or process that runs in the background, continuously listening to the RabbitMQ message queue for new tasks.

Step-5: Core (Watcher Module) Picks the .pcap File for Analysis – Upon receiving a new message containing the SHA256 hash value, the watcher module retrieves the corresponding .pcap file from the shared directory by matching the hash values and pass on to Scapy for threat categorization.

Step-6: Core Engine Analyzes the .pcap File for Threat Categorization – All necessary checks for validating the .pcap are done first, after validation, the .pcap is loaded with Scapy. Scapy runs first in stream mode to extract all the stream HTTP artifacts. Scapy then runs in packet capture mode to extract UDP artifacts. Once extracted, all component results are stored in their respective json. The rule engine is now invoked by the watcher. Rule engine then parses all the available rules and runs them over the generated json artifacts. Results are aggregated and is given to aggregator which returns back with the valid threat family formulated by the rules. The analysis results are formatted as a JSON object that contains the .pcap file SHA256, the threat type, MITRE ATT&CK technique, and any other relevant information such as the severity score, the analyzed time, and so on.

Step-7: Core Engine Submits the JSON Object to the API as Well as Status – Once the analysis is complete, the core engine sends the JSON object containing the analysis results to the API through a RESTful endpoint.

Step-8: The API saves the JSON object in a SQLite database table for later retrieval based on SHA256 of uploaded file.

Step-9: UI Fetches the Analysis Results from

the API and Displays the Results to the User – The UI retrieves the analysis results from the API through a RESTful endpoint by passing SHA256 of file. The UI parses the JSON object and displays the analysis results to the user in a user-friendly format, such as a table or a chart. The user can interact with the UI to view the analysis results of different .pcap files, filter the results based on different criteria, or export the results to a PDF file and view measurements.

4 RELATED WORK AND EXISTING TECHNOLOGIES

There are several existing technologies and tools in the field of threat categorization, such as Snort (Snort, 2023), Suricata (Suricata, 2023), and Bro (Zeek, 2023). Snort is a widely-used intrusion detection system (IDS) that can categorize network threats based on pre-defined rules. Similarly, Suricata is an open-source IDS and Intrusion Prevention System (IPS) that uses signature-based detection to categorize network threats. Bro is another open-source network security monitoring tool that can detect and categorize network threats.

However, these traditional methods of threat categorization have some limitations when it comes to detecting and categorizing Command-and-Control (C2) communication patterns used by attackers to control infected hosts and steal sensitive information. This is where ArkThor comes in, as it is specifically designed to identify and categorize C2 communication patterns in live stream or archived .pcap files.

In addition to traditional IDS and IPS tools, there are also machine learning-based solutions that can be used for threat categorization. For example, there are several research papers that propose the use of machine learning algorithms for categorizing network threats. These algorithms can learn from previous data and identify patterns in network traffic to detect and categorize threats. However, machine learning-based solutions can also have some limitations, such as requiring a large amount of training data and being vulnerable to adversarial attacks. This is why ArkThor combines traditional rule-based detection methods with the flexibility of machine learning-based solutions, creating a powerful and accurate threat categorization engine.

Moreover, containerization is a key feature of ArkThor that provides significant benefits for organizations of all sizes. By containerizing all components of the tool, including the user interface, APIs, core engine, and RabbitMQ, ArkThor becomes easy to deploy and use in any organization, regardless of their

level of expertise. Containerization enables the tool to be delivered as a plug-and-analyze solution, allowing organizations to quickly integrate it into their existing infrastructure and start identifying and mitigating threats. One of the main advantages of containerization is that it ensures the tool's compatibility with a wide range of environments, including different operating systems and cloud platforms. Containers provide a lightweight and portable way to package and distribute software, allowing organizations to easily deploy ArkThor on-premises, in the cloud, or in hybrid environments. This flexibility ensures that the tool is accessible to organizations of all sizes, from small businesses to large enterprises.

5 CONCLUSION

Threat categorization is a critical task in modern cybersecurity, and it requires accurate and efficient detection of various types of threats. While traditional methods of threat categorization, such as signature-based detection and rule-based detection, have been effective to some extent, they are not always sufficient for identifying and categorizing complex threats. This is where ArkThor comes in, providing a powerful and flexible solution for threat categorization by combining traditional rule-based detection methods with machine learning-based methods.

Through its containerized architecture and plug-and-analyze solution, ArkThor is accessible to organizations of all sizes and levels of expertise. By using machine learning algorithms, ArkThor can learn from previous data and identify patterns in network traffic to detect and categorize threats more accurately. This combination of traditional and modern detection methods results in a powerful and reliable threat categorization engine.

In the future, we plan to continue to improve the accuracy and efficiency of ArkThor's threat categorization capabilities by incorporating more advanced machine learning algorithms and improving the rule-based detection methods. We also plan to expand ArkThor's capabilities to include the detection and categorization of more types of threats, including those related to IoT devices and cloud environments.

To summarize, ArkThor is a valuable tool for organizations looking to enhance their threat detection and response capabilities. Its containerized architecture and combination of traditional and modern detection methods make it accessible and powerful for organizations of all sizes and levels of expertise.

ACKNOWLEDGEMENTS

This work is carried out as a capstone project under a certification program conducted by C3i Hub, IIT Kanpur, India, and Talensprint Pvt. Ltd. Bengaluru, India.

REFERENCES

- Acronis (2023). Bokbot-from-banking-trojan-to-backdoor. <https://www.acronis.com/en-us/cyber-protection-center/posts/icedid-bokbot-from-banking-trojan-to-backdoor/#:~:text=IcedID%2C%20also%20known%20as%20BokBot,attachments%20to%20infect%20victims'%20machines.>
- Blackberry (2023). Strat malware. <https://blogs.blackberry.com/en/2021/10/threat-thursday-strat-malware.>
- Checkpoint (2023). Iceid malware. [https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-malware/icedid-malware/.](https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-malware/icedid-malware/)
- CSE Department IIT Kanpur, I. (2023). C3i center. [https://www.security.cse.iitk.ac.in/.](https://www.security.cse.iitk.ac.in/)
- Elasticsearch (2023). Elastic stack. <https://www.elastic.co/kibana.>
- F-secure (2023). Graftor malware. <https://www.f-secure.com/v-descs/trojan-w32-graftor.shtml.>
- GrafanaLabs (2023). Grafana. <https://www.grafana.com.>
- Handa, M. J. S. P. A. and Kumar, N. (2023). Arkthor is live. <https://github.com/JawedCIA/ArkThor/wiki#ArkThor-Demystified.>
- Malwarebytes (2023). Cobalt strike malware. <https://www.malwarebytes.com/blog/detections/trojan-cobaltstrike.>
- Mitre (2023). Mitre att&ck. [https://attack.mitre.org/.](https://attack.mitre.org/)
- RabbitMQ (2023). Rabbitmq. [https://www.rabbitmq.com/.](https://www.rabbitmq.com/)
- Scapy (2023). Scapy. [https://scapy.net/.](https://scapy.net/)
- Snort (2023). Snort - network intrusion detection & prevention system. [https://www.snort.org/.](https://www.snort.org/)
- Suricata (2023). Suricata. [https://suricata.io/.](https://suricata.io/)
- threatfox (2023). Threatfox by abuse. [https://threatfox.abuse.ch/.](https://threatfox.abuse.ch/)
- Zeek (2023). An open source network security monitoring tool. [https://zeek.org/.](https://zeek.org/)