The object of this study is multi-layered cybersecurity systems for detecting and countering advanced persistent threats through the integration of machine learning technologies, artificial intelligence, and multi-layered security systems. The task relates to the need to design adaptive detection systems capable of effectively responding to new and modified threats while improving accuracy and minimizing delays. An integrated approach was devised in the study, which combines conventional detection methods (signature analysis, correlation rules) with modern technologies such as machine learning and Artificial Intelligence assistants. Each layer of the system showed varying levels of effectiveness: for example, antivirus solutions were most effective at detecting known threats but failed to cope with modified threats, which were detected by correlation rules. Machine learning proved most effective at detecting fileless attacks and anomalous activity that other tools could not detect. It is through the combination of these methods that the detection system proved to be effective, providing a high level of protection. The results are due to the efficiency of combining several layers of defense, in which each subsequent layer compensates for the shortcomings of the previous one. Antivirus solutions detected 100 % of known threats, while correlation rules identified all modified malicious files. Overall, the system was able to detect 98 % of malicious files and 99 % of tactics, techniques, and procedures used in advanced persistent threats attacks. A unique feature of the research is the integration of the Artificial Intelligence assistant, which automates threat analysis processes and speeds up response times by leveraging historical data and the context of past incidents. This reduces the workload on cybersecurity specialists and improves the overall effectiveness of the detection system, allowing for the quick identification of new threats and a reduction in false positives. Practical application of the results is possible in various critical sectors, including financial institutions, government organizations, and energy companies. The system demonstrates high flexibility and scalability, making it possible to easily adapt to different infrastructures and types of threats

Keywords: advanced persistent threat, intrusion detection systems, machine learning, anomaly detection, large language models

UDC 004.681
DOI: 10.15587/1729-4061.2024.318336

DESIGN OF AN INTEGRATED DEFENSE-IN-DEPTH SYSTEM WITH AN ARTIFICIAL INTELLIGENCE ASSISTANT TO COUNTER MALWARE

Danyil Zhuravchak
Corresponding author
PhD*
E-mail: danyil.y.zhuravchak@lpnu.ua
Maksym Opanovych
PhD Student*
Anastasiia Tolkachova
PhD Student*
Valerii Dudykevych
Doctor of Technical Sciences, Professor*
Andrian Piskozub
Doctor of Technical Sciences, Associate Professor*
*Department of Information Security
Lviv Polytechnic National University
S. Bandery str., 12, Lviv, Ukraine, 79013

Received 20.09.2024
Received in revised form 25.11.2024
Accepted 05.12.2024
Published 25.12.2024

***How to Cite**: Zhuravchak, D., Opanovych, M., Tolkachova, A., Dudykevych, V., Piskozub, A. (2024). Design of an integrated defense-in-depth system with an artificial intelligence assistant to counter malware. Eastern-European Journal of Enterprise Technologies, 6 (2 (132)), 64–73. https://doi.org/10.15587/1729-4061.2024.318336*

1. Introduction

In today's world where cyber threats are constantly evolving, the importance of scientific research in the field of cyber security is becoming more and more relevant. In particular, research focused on advanced persistent threats (APTs) is of great practical importance because of the high level of threat these attacks pose to critical infrastructures, government organizations, and corporate systems. APT attacks are complex, long-lasting, and often aimed at gaining access to sensitive information, making them particularly dangerous for organizations of all levels.

Scientific research in this area is important because of the constant improvement of attack methods and the evolution of cyber threats that can bypass conventional defenses. Therefore, designing adaptive protection systems that can quickly respond to new and modified threats is a necessary condition for maintaining information security.

The integration of machine learning and artificial intelligence technologies into threat detection systems opens up new opportunities for improving detection methods, in particular by automating analysis and monitoring processes. Such technologies can analyze large volumes of data in real time, enabling the detection of threats in the early stages [1, 2].

The practical significance of these studies relates to increasing the reliability of cyber defense systems and minimizing the consequences of APT attacks. In particular, the use of multi-level protection systems based on the "Swiss Cheese Model" makes it possible to create additional barriers to threats, where each level of protection compensates for the shortcomings of the previous one [3]. The integration of AI chatbots and other artificial intelligence tools makes it possible to automate the processes of threat analysis and reduce the response time to attacks [4].

Therefore, research aimed at designing adaptive multi-level protection systems using machine learning and artificial intelligence technologies is relevant to ensure the proper level of cyber security, especially under the conditions of increasing complexity and frequency of APT attacks.

2. Literature review and problem statement

Work [5] reports the results of research in which machine learning methods were used, such as C5.0 decision tree, Bayesian network, and deep neural networks for timely detection and classification of APT attacks. Deep neural networks are shown to have superior performance for timely detection of APT attacks compared to other classification models. However, issues related to the possible complexity of setting up the models and the requirements for computing resources remain unresolved. This makes the use of such methods limited in some scenarios. Optimizing resource requirements or searching for alternative methods of machine learning can be an option to overcome related difficulties.

Work [6] gives the results of studies in which NodLink was developed. It is the first online system for detecting APT attacks. It provides high accuracy of detection without loss of detailed analysis. The importance of building systems capable of promptly responding to APT attacks is shown. However, the issues related to the possibility of scaling the system and ensuring the stability of work with a large number of simultaneous requests remained unresolved.

Work [7] reports the results of studies in which the authors presented TBDetector. This is a method for detecting transformer-based APT attacks. It uses provenance analysis to ensure long system execution with space efficiency. The potential of using transformers to detect slow attacks is shown. However, issues related to the complexity of configuring transformers for high-precision analysis of large volumes of data remained unresolved. This may limit the application of the method under conditions with increased performance requirements.

Work [8] reports the results of research examining the challenges of the Internet of Things (IoT) network in the context of APT attacks. The importance of using machine learning methods to counter cyber threats in IoT is shown. The difficulty of detecting APT attacks due to their small share in normal traffic is shown. However, issues related to the high dynamics of IoT networks and the need to adapt machine learning models to these conditions remain unresolved. A likely option to overcome related difficulties is to devise methods for the automatic adaptation of models to ensure constant monitoring of threats.

Work [9] gives the results of studies where XfedHunter is presented. It is a training platform with the ability to detect APTs in SDN networks. It is shown that the use of graph neural networks and deep learning contributes to the effective detection of malicious events. However, issues related to the complexity of processing a large amount of heterogeneous data remain unresolved.

Work [10] reports the results of studies in which the DNS and TCP traffic of APT attacks were analyzed, and new statistical features were developed for the classification of this traffic. It is shown that the use of the AdaBoost algorithm makes it possible to increase the accuracy of the classification of APT attacks even with a limited number of samples. However, issues related to the adaptability of features to new types of traffic and threats remain unresolved. An option to overcome these difficulties may be the development of dynamic features capable of adapting to new types of attacks in real time.

Work [11] gives the results of studies in which a bidirectional recurrent neural network (Bi-RNN) is used to detect multi-stage attacks. It is shown that the application of the SMOTE&CNN-SVM algorithm makes it possible to effectively identify attack chains. However, issues related to computational complexity and time spent in processing large volumes of data remain unsolved.

Summarizing the studies above, one can note that existing methods for detecting APT attacks using conventional approaches such as signature detection, correlation rules, and even some modern systems based on machine learning, have their limitations. In particular, their inability to adapt to new, previously unknown, or modified threats, as well as threats masquerading as legitimate activity, is a major problem. Thus, we have noticed the need for the development and implementation of innovative threat detection systems that are able to adapt to the complex and constantly changing landscape of cyber threats.

3. The aim and objectives of the study

The purpose of our study is to design an integrated system for detecting APT attacks based on a deep defense model using modern machine learning and artificial intelligence technologies. This could increase the effectiveness of security monitoring centers (SOCs), ensuring fast and accurate detection of threats, including advanced persistent threats, and minimizing the risks associated with cyber attacks.

To achieve this goal, the following tasks were set:

– to design a multi-level protection architecture for detecting APT attacks, which integrates machine learning and artificial intelligence methods;

– to evaluate the effectiveness of the proposed deep protection model in detecting and countering APT attacks;

– to evaluate the work of the AI assistant, which automates the analysis of threats and speeds up the response to incidents.

4. The study materials and methods

The object of our research is methods for detecting and preventing APT threats by integrating machine learning technologies, artificial intelligence, and multi-level security systems.

The research hypothesis assumes that the integration of modern technologies, such as AI assistants and machine learning, into multi-level protection systems significantly increases the effectiveness of detecting and responding to APT (persistent threat) attacks. This approach would allow security systems not only to detect known threats but also adapt to new, unknown threats. The use of AI could help automate threat analysis processes, reduce response time, and reduce the number of false positives.

The research method used for this work is the experimental research method. A series of experiments were conducted to evaluate the effectiveness of the concept of defense in depth using an example of the Swiss cheese model, which involves the combination of several layers of security systems to detect and prevent malicious activity. The goals of the experiment were as follows:

– assessment of the effectiveness of each security layer separately;

– analysis of how effectively each subsequent layer resolves the vulnerabilities of the previous one;

– assessment of the effectiveness and usefulness of the AI assistant as one of the layers in the protection model.

Windows Defender, Snort, and Splunk are used to detect and prevent malicious activity [12–14]. VMware Workstation Pro 17 [15] was used for virtualization and network configuration of laboratory environment. Windows Server 2022, Windows 11, Windows 10, Ubuntu 22.04.4 LTSm and Kali Linux virtual machines were employed in the experiment.

Malware, viruses, and simulators taken from public sources were used to reproduce the malicious activity. Additionally, some attacks were manually recreated according to MITRE and attack reports. The experiment was conducted in 2 stages. At the first stage, methods for detecting malicious files and fileless threats, that is, those that are executed from memory and not from a file on the operating system, were tested. To reproduce the activity of malicious files, the files were moved to test virtual machines and run from them. To reproduce fileless threats, the malicious code was downloaded from public sources or from Kali Linux and played in memory. In the second stage, the techniques were manually recreated according to the MITRE threat categories. Reproduction was carried out either manually or using simulators such as Atomic Red Team [16].

ChatGPT4-based AI assistant assists in incident investigation. It uses historical data about past incidents, information about the environment and its features. In addition, malicious activity context taken from MITRE and indicators of compromise taken from open sources [17].

Windows virtual machines were joined to a single Active Directory network and used as test machines. In addition to malicious activity, normal activity was also simulated on the machines. It involved downloading various files from the Internet and from other machines inside the domain, working with documents and file editors, software development environments, remotely connecting to other machines in the domain, using file exchangers. We also simulated the use of messengers, etc., to reproduce activity close to reality.

Kali Linux was employed to reproduce malicious activity from a remote machine and to upload malicious files to a test environment.

Ubuntu was applied to install Snort through which the network traffic of the test environment passed.

Windows audit policies and Sysmon configuration [18] were manually adjusted according to the experiment's goals to optimize Splunk performance and to reduce the number of false positives. All event logs, as well as all alerts from Windows Defender and Snort, are collected in Splunk, in which a risk-based detection system has been built. With each alert, a certain number of risk points were assigned to the corresponding machine, user, or IP address. According to the number of risk points, the priority of alerts changed. With this risk system, all alerts are grouped by user, endpoint, or IP address, and each new alert increases the risk score of the corresponding object, which in turn increases the priority of alert consideration. The more critical the user or endpoint, the greater the risk modifier. Antivirus was used to detect endpoint threats, machine learning and Snort were applied to detect threats in network traffic, and correlation rules were exploited to detect all types of threats.

5. Results of the APT attack detection model

5. 1. Development and implementation of a multi-level protection architecture based on the "Swiss cheese" model with the integration of an AI assistant

Fig. 1 shows the defense-in-depth model known as the Swiss cheese model. This concept is the basis of modern cyber security and implies that no defense mechanism is perfect, each has its weaknesses or "holes". However, combining several layers of protection makes it possible to compensate for these vulnerabilities. Each level of protection, like a layer of Swiss cheese, has its own shortcomings, but when the layers are combined into a single system, the chances of a successful penetration of an attacker are significantly reduced since it is necessary to pass through several barriers. This provides robust multi-layered protection where each layer acts as an additional security mechanism, reducing the likelihood of an attack being successful. Thus, the defense-in-depth model is based on the use of various tools and technologies to protect the information infrastructure, which is a key strategy for reducing risks in complex systems.

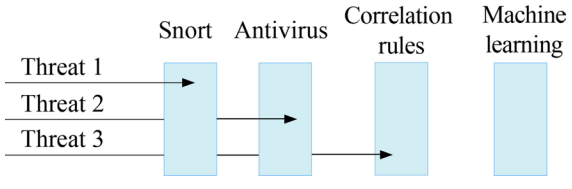


Fig. 1. Conceptual model of Swiss cheese

The general high-level architecture shown in Fig. 2 represents an architectural solution for implementing the protection model with the integration of an AI assistant into it.

Architecture begins with the collection of data from various sources. Data comes from file system events, operating system performance, network activity, and security events. These sources are the basis for monitoring the current state of the system. The data then passes through several levels of analysis. This includes signature analysis, which detects threats based on known malicious signatures. Anomaly detection, which makes it possible to identify atypical behavior, correlation rules to establish relationships between events. Also, machine learning methods that provide more complex analysis of large volumes of data. In addition, the system is integrated with SIEM, which makes it possible to centrally collect and correlate events from various sources.

Contextual information, which includes documentation, the MITRE ATT&CK taxonomy, and analysis of past incidents, complements the analysis process, allowing new threats to be identified and understood based on past incidents and known attack patterns. LLM agents work with this data to create vector representations that are used for further searches in vector databases. The vector database is another critical component of the system, providing search for similar incidents or threats based on computational embeddings. It makes it possible to quickly find relevant information, which helps analysts respond more quickly to new threats. Chatbot based on OpenAI ChatGPT API is an interface for analysts to interact with the system. It processes text queries from users, searches vector databases or SIEM systems, and provides relevant query results. If the chatbot does not find relevant information, it informs the user that the answer was not found, thereby simplifying the decision-making process for SOC analysts.

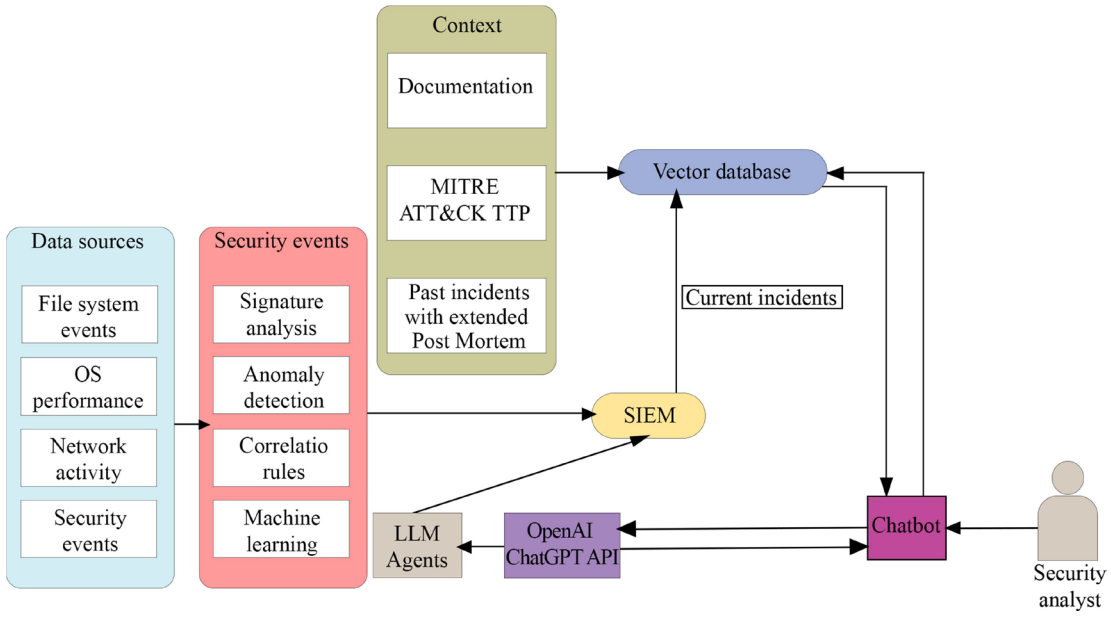


Fig. 2. Conceptual diagram of the proposed solution

5. 2. Evaluation of the effectiveness of the defense-in-depth model

For better representativeness, the threat detection results of both experiments were divided into 3 separate stages for each experiment, respectively. At the first stage, the detection results of each tool are shown separately. The second stage shows the results of sequential detection, which demonstrates how each subsequent layer detects threats that passed the previous layer. At the third stage, the number of false positives is presented.

Fig. 3 shows how different threat detection methods work with different types of threats: malicious files, modified malicious files, fileless threats, and modified fileless threats. Fig. 3 illustrates the effectiveness of each method separately, which makes it possible to understand in which scenarios each of the detection methods is the most effective.

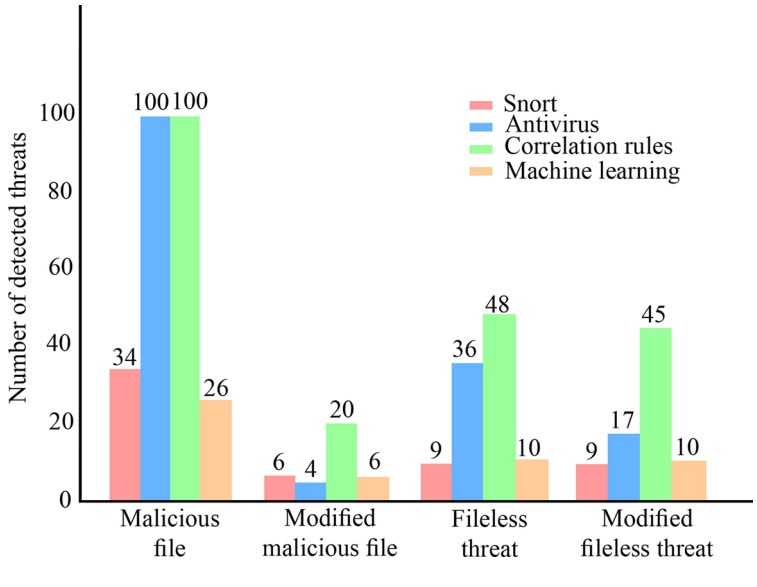


Fig. 3. Detection of malicious software by various methods

The antivirus demonstrated excellent performance against conventional malicious files, detecting 100 % of threats (100/100). However, its effectiveness significantly decreases when working with modified malicious files, where only 20 % of threats are detected (4/20). The reason is that the malicious code modifications included changing the name of variables, obfuscation, additional encryption of the code, and adding a lot of features to confuse the analysis. Such changes complicate the behavioral analysis of the antivirus, which focuses on known signatures or standard patterns of behavior. Correlation rules are highly effective in detecting more sophisticated attacks, including modified malicious files, where they achieved 100 % efficiency (20/20). This makes correlation rules an important tool for threat detection. They adapt their methods because the rules are based on behavioral patterns rather than signatures. Snort and machine learning demonstrated poor detection rates. These results were expected since both tools are mainly used to detect threats in the network. For example, Snort detected only 6 out of 20 modified malicious files (30 %). The main reason for these results is that not all threats had network activity. Because Snort focuses on network traffic, the absence of such activity reduces its ability to detect threats. Machine learning has proven to be effective when dealing with more complex threats, such as fileless threats and modified threats, where it detected more threats compared to other methods. For example, machine learning detected 10 out of 54 fileless threats, which, while not an absolute result, shows its potential in detecting unknown or more sophisticated attacks.

In summary, Fig. 3 makes it clear that each method has its strengths and weaknesses, depending on the type of threat. This information is useful for determining the optimal methods for detecting threats in different scenarios and reveals the need to combine different approaches to build a more reliable protection system.

Fig. 4 is central to understanding multi-layered protection, in which each layer of protection works sequentially to

detect threats that were not detected by the previous layer. Fig. 4 illustrates how the protection system detects threats at each level and how subsequent layers complement the previous ones. An important aspect is that after each level of protection, the number of undetected threats decreases.

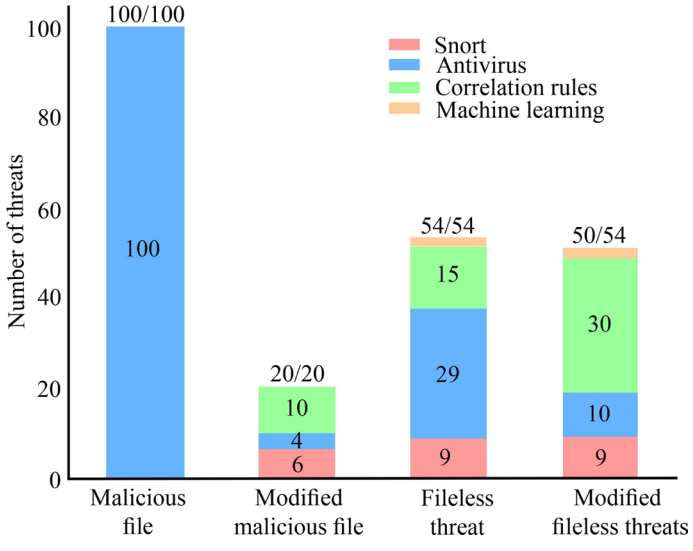


Fig. 4. Detection of malware and defense-in-depth tools

Fig. 4 demonstrates how a multi-layered approach to protection makes it possible to detect a greater number of threats through the use of different layers of protection. Each method (Snort, antivirus, correlation rules, and machine learning) works in turn, trying to detect threats that were not detected by the previous method. It is important to note that this approach makes it possible to gradually reduce the number of undetected threats by using the strengths of each detection method. At the beginning of the first level of protection, Snort detects only a part of the threats. For example, in the case of fileless threats, Snort detects 9 out of 54 threats, leaving 45 threats to be handled by other layers of protection. This is because Snort, which targets network traffic, cannot detect threats that do not generate active traffic. The next layer, namely the antivirus, is able to detect additional threats that Snort did not detect. In the case of fileless threats, the antivirus detects 29 out of 45 remaining threats, indicating its ability to detect threats that may have some activity in the file system or other system components. Correlation rules continue this process by focusing on behavioral and pattern analysis. For example, after the antivirus, 16 undetected threats remained, of which the correlation rules detect 15. This shows that the correlation rules work effectively with the behavioral characteristics of the threats, allowing the system to recognize patterns that may be invisible to conventional detection methods. The last layer – machine learning – detects those threats that remained after the work of all previous methods. In the case of fileless threats, machine learning detects the last threat that remained undetected after all previous methods worked.

The key advantage of this model is that each subsequent layer compensates for the shortcomings of the previous ones, gradually reducing the number of undetected threats. This makes it possible to achieve high efficiency of the protection system, even if each method has its limitations. Another important aspect is that some threats have been specially designed to leave minimal traces on the system and avoid conventional detection methods. For example, using minimal interaction with the Windows API and writing custom functions allowed some threats to escape early detection. However, thanks to a multi-layered approach that includes behavioral analysis and machine learning, most of these threats were still detected. Out of 224 threats, 220 were detected, which is 98 % of detected threats.

Taking into account Fig. 5, it is worth noting that the number of false positives of the antivirus is low. All of them were caused by legitimate software that has features similar to those of malicious software.

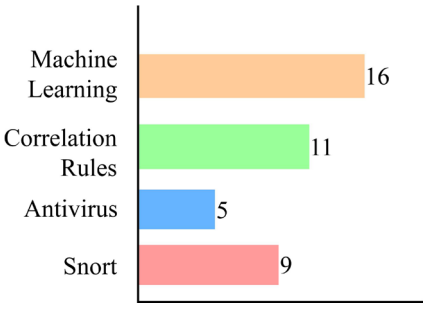


Fig. 5. Number of false positives for each detection method

The number of false positives of correlation rules, as in the case of antiviruses, remains low, although slightly higher. These triggers were associated with malicious-like activity performed by legitimate software. The number of Snort and machine learning false positives is mostly due to legitimate network traffic being identified as malicious or anomalous. This experiment replicated the techniques and tactics used by APT groups, so both the tools and teams mentioned in the reports on the activities of the malicious groups were used. The results of the experiment are represented in the form of tables because of the large number of types of threats for visualization.

Table 1 demonstrates how different threat detection methods work with attack techniques according to the MITRE ATT&CK matrix. Table 1 gives the number of techniques identified at different stages of attacks, such as reconnaissance, execution of malicious commands, privilege escalation, and command and control. Snort is focused on detecting network threats, so it detects techniques related to network activity, but its effectiveness drops for attacks that do not generate active traffic. Antivirus performs well at malware execution techniques but is not always effective at later stages of an attack, such as lateral movement or command and control. Correlation rules detect behavioral anomalies and are most effective in cases of privilege escalation and lateral movement. Machine learning detects more complex attacks, particularly at the command and control stages, but its effectiveness depends on how the model is set up.

The reproduced malicious activity in the table is displayed according to the MITRE threat matrix. Table 1 illustrates the threat detection results of various defense techniques such as Snort, antivirus, correlation rules, and machine learning in terms of their ability to detect activity associated with different stages of cyberattacks. Table 1 shows how each method handles detecting network activity, executing malicious commands, stealthy actions to gain privileges or movement within the system, and attempts to gain control over infected devices. Snort

detects a limited number of threats associated with the initial stages of attacks where there is network traffic, but its effectiveness drops at later stages where network activity is less or hidden. The antivirus does a good job of detecting the execution of malicious code but shows weaker results in detecting complex actions that leave no obvious traces in the file system. Correlation rules are successful in analyzing behavioral anomalies that occur when attempting to gain privileges or move around a network, allowing for the detection of hidden threats. Machine learning does the best job at detecting attempts to control the system and sophisticated attacks in the late stages but has an increased false positive rate.

Table 1

The number of detected tactics and techniques

Type of threat / Detection method	Snort	Antivirus	Correlation rules	Machine learning
Reconnaissance	2/6	0/6	6/6	4/6
Initial Access	X	38/44	43/44	X
Execution	X	21/54	49/54	X
Persistence	X	26/40	38/40	X
Privilege Escalation	X	34/42	41/42	X
Defense Evasion	X	21/41	40/41	X
Credential Access	X	14/41	38/41	X
Discovery	9/29	0/29	20/21	19/21
Lateral Movement	14/24	0/24	16/24	20/24
Collection	X	9/16	12/16	X
Command and Control	4/29	0/29	9/29	24/29
Exfiltration	9/13	1/13	11/13	12/13
Impact	X	0/9	9/9	X

Table 2 demonstrates the overall effectiveness of a multi-layered protection system, where each subsequent layer stops threats that have been able to bypass previous ones. The results show that even if individual methods failed to detect all threats in the initial stages, such as network attacks or malicious code execution, subsequent layers of protection compensated for these gaps. For example, correlation rules and machine learning successfully detected threats that were missed by Snort and antivirus. The final results indicate that, thanks to the combination of different approaches, the number of undetected threats decreased at each successive level of protection, which ensured a high overall system efficiency. Thus, it can be seen that the system is quite effective, as each subsequent layer of protection compensates for the vulnerabilities and shortcomings of the previous one, which made it possible to detect 385 out of 388 attacks, which is 99 %.

Analysis of false positives is given in Table 3 revealed that the antivirus is the most accurate while the other layers had a significant number of false positives: Snort – 25, correlation rules – 44, machine learning – 45.

False positives are a significant problem. Their large number can lead to overloading of people. And also to the fact that critical notifications can be missed or processed late. Therefore, two tools were used to overcome this problem: a risk assessment system and an AI assistant.

Owing to the risk assessment system, it was possible to solve the problem of prioritization, so alerts related to more critical risk objects or those that generated more alerts were considered first. That made it possible to respond to incidents faster and more efficiently.

Table 2

The number of detected defense-in-depth tactics and techniques

Type of threat / Detection method	Snort	Antivirus	Correlation rules	Machine learning	Number of undetected threats
Reconnaissance	2/6	0/4	4/4	0/0	0
Initial Access	X	38/44	6/6	X	0
Execution	X	21/54	32/33	X	1
Persistence	X	26/40	16/16	X	0
Privilege Escalation	X	34/42	5/6	X	1
Defense Evasion	X	21/41	20/20	X	0
Credential Access	X	14/41	27/27	X	0
Discovery	9/29	0/20	10/20	10/10	0
Lateral Movement	14/24	0/10	8/10	2/2	0
Collection	X	9/16	6/7	X	1
Command and Control	4/29	0/25	9/25	16/16	0
Exfiltration	9/13	1/4	3/3	0/0	0
Impact	X	0/9	9/9	X	0

Table 3

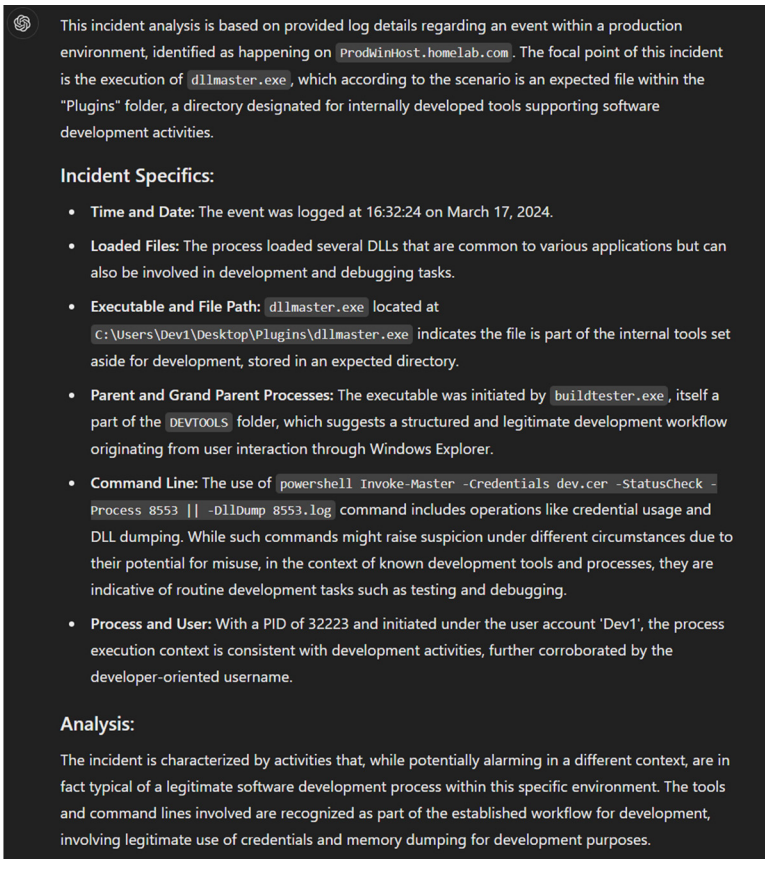
The number of false positives during the detection of tactics and techniques

Type of threat / Detection method	Snort	Antivirus	Correlation rules	Machine learning
Reconnaissance	3	0	4	8
Initial Access	X	1	2	X
Execution	X	3	2	X
Persistence	X	0	5	X
Privilege Escalation	X	0	6	X
Defense Evasion	X	0	1	X
Credential Access	X	0	2	X
Discovery	6	0	1	10
Lateral Movement	5	0	2	9
Collection	X	0	9	X
Command and Control	9	0	9	11
Exfiltration	2	0	1	7
Impact	X	0	0	X

5. 3. Evaluation of the work of the AI assistant

The AI assistant, in turn, provided context to incident alerts using data from an internal database and open sources. On the example from Fig. 6, after analyzing the alert generated in SIEM, the AI-assistant provided the correct conclusion that this activity is legitimate, confirming it with data from the internal database, in particular, the features of the internal software.

Also, the AI assistant proved to be effective in identifying malicious activity. On the example from Fig. 6 one can see how it identified suspicious activity based on DLLs used, process path, commands executed, and user privileges. After analyzing the data, the AI assistant determined that the activity was related to credential theft and supported this with information from the MITRE ATT&CK Enterprise Matrix, providing data on the attack and the APT groups it was associated with. Fig. 7 shows the detection of malicious activity by the AI assistant. The AI assistant is also useful in the context of historical data analysis.



This incident analysis is based on provided log details regarding an event within a production environment, identified as happening on `ProdWinHost.homelab.com`. The focal point of this incident is the execution of `dllmaster.exe`, which according to the scenario is an expected file within the "Plugins" folder, a directory designated for internally developed tools supporting software development activities.

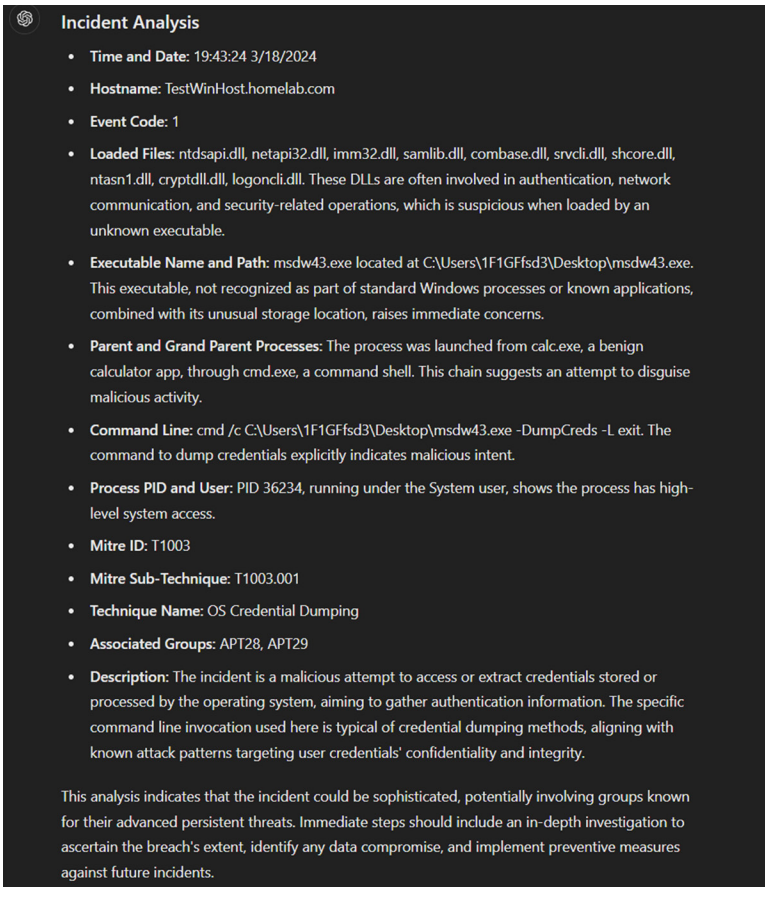
Incident Specifics:

- **Time and Date:** The event was logged at 16:32:24 on March 17, 2024.
- **Loaded Files:** The process loaded several DLLs that are common to various applications but can also be involved in development and debugging tasks.
- **Executable and File Path:** `dllmaster.exe` located at `C:\Users\Dev1\Desktop\Plugins\dllmaster.exe` indicates the file is part of the internal tools set aside for development, stored in an expected directory.
- **Parent and Grand Parent Processes:** The executable was initiated by `buildtester.exe`, itself a part of the `DEVTOOLS` folder, which suggests a structured and legitimate development workflow originating from user interaction through Windows Explorer.
- **Command Line:** The use of `powershell Invoke-Master -Credentials dev.cer -StatusCheck -Process 8553 || -DllDump 8553.log` command includes operations like credential usage and DLL dumping. While such commands might raise suspicion under different circumstances due to their potential for misuse, in the context of known development tools and processes, they are indicative of routine development tasks such as testing and debugging.
- **Process and User:** With a PID of 32223 and initiated under the user account 'Dev1', the process execution context is consistent with development activities, further corroborated by the developer-oriented username.

Analysis:

The incident is characterized by activities that, while potentially alarming in a different context, are in fact typical of a legitimate software development process within this specific environment. The tools and command lines involved are recognized as part of the established workflow for development, involving legitimate use of credentials and memory dumping for development purposes.

Fig. 6. Analysis of the incident by an AI assistant



Incident Analysis

- **Time and Date:** 19:43:24 3/18/2024
- **Hostname:** TestWinHost.homelab.com
- **Event Code:** 1
- **Loaded Files:** ntdsapi.dll, netapi32.dll, imm32.dll, samlib.dll, combase.dll, srvcli.dll, shcore.dll, ntasn1.dll, cryptdll.dll, logoncli.dll. These DLLs are often involved in authentication, network communication, and security-related operations, which is suspicious when loaded by an unknown executable.
- **Executable Name and Path:** msdw43.exe located at C:\Users\1F1GFfsd3\Desktop\msdw43.exe. This executable, not recognized as part of standard Windows processes or known applications, combined with its unusual storage location, raises immediate concerns.
- **Parent and Grand Parent Processes:** The process was launched from calc.exe, a benign calculator app, through cmd.exe, a command shell. This chain suggests an attempt to disguise malicious activity.
- **Command Line:** cmd /c C:\Users\1F1GFfsd3\Desktop\msdw43.exe -DumpCreds -L exit. The command to dump credentials explicitly indicates malicious intent.
- **Process PID and User:** PID 36234, running under the System user, shows the process has high-level system access.
- **Mitre ID:** T1003
- **Mitre Sub-Technique:** T1003.001
- **Technique Name:** OS Credential Dumping
- **Associated Groups:** APT28, APT29
- **Description:** The incident is a malicious attempt to access or extract credentials stored or processed by the operating system, aiming to gather authentication information. The specific command line invocation used here is typical of credential dumping methods, aligning with known attack patterns targeting user credentials' confidentiality and integrity.

This analysis indicates that the incident could be sophisticated, potentially involving groups known for their advanced persistent threats. Immediate steps should include an in-depth investigation to ascertain the breach's extent, identify any data compromise, and implement preventive measures against future incidents.

Fig. 7. Detection of malicious activity by an AI assistant

For example, after receiving an alert, it is possible to learn about previous investigations related to the risk object, process, etc. In the example shown in Fig. 8, it is possible to obtain the context of past alerts related to "buildtester.exe" and the conclusion drawn in the previous incident.

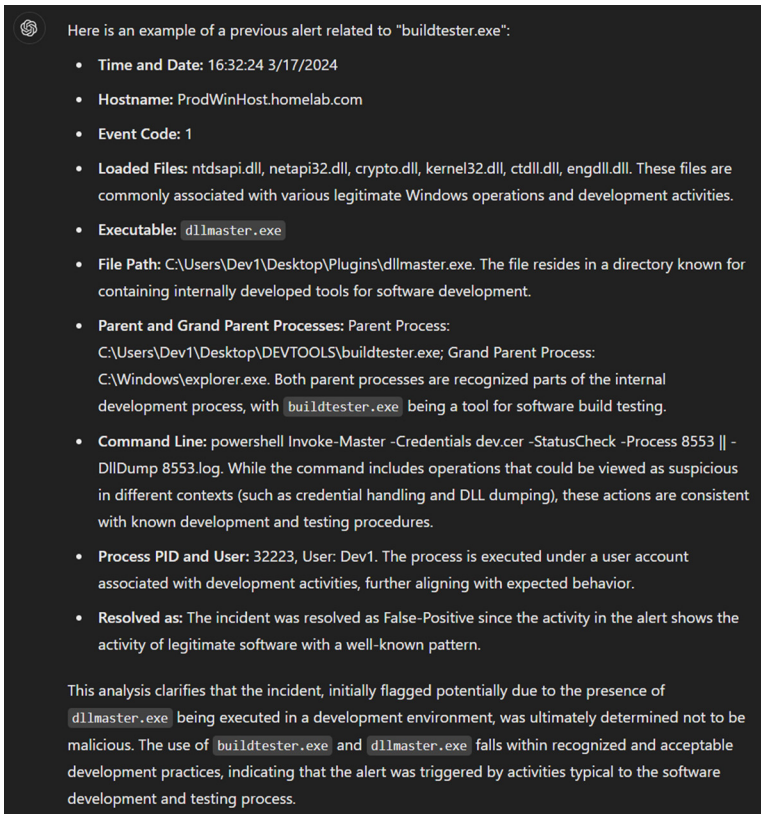


Fig. 8. Provision of historical data by an AI assistant

This incident context significantly speeds up incident analysis and makes it possible to quickly determine whether it is a real incident or a false positive. This makes it easier and faster for analysts, which would improve the efficiency of their work.

6. Discussion of results of investigating the integrated system for detecting APT attacks

Each element of the protection system demonstrated its strengths and weaknesses in detecting malicious files and malicious activity, which is confirmed by the data in Fig. 3 and in Table 1. Antivirus solutions do a good job of detecting known threats, but their effectiveness drops significantly when it comes to modified or unknown threats. Correlation rules have proven to be particularly effective in detecting such threats, particularly fileless attacks, which are not always detectable by other means. Machine learning did a good job of detecting anomalies in network traffic, which allowed new threats to be identified. However, each of these detection methods has its limitations. Combining all of these tools into a single system provided a comprehensive approach to protection that allowed detection of 98 % of malicious files and tools, and 99 % of tactics, techniques, and procedures. This is especially clear in Fig. 4 and in Table 2, which show a significant reduction in the number of undetected threats compared to the use of individual methods.

The main advantage of the proposed system is its multi-level approach, which makes it possible to adapt to new threats and ensures the protection of the entire infrastructure. The combination of several detection methods allows each of them to compensate for the weaknesses of the others.

It is also important to emphasize the functionality of the AI assistant, which helped significantly speed up the incident investigation process. The AI assistant analyzes security alerts and provides contextual information that makes it possible to more effectively categorize activity and identify its affiliation with specific malicious groups. Owing to access to historical data, the AI assistant provides additional information about suspicious processes, teams, users, and IP addresses that appeared in previous incidents, which significantly speeds up the investigation process and increases the overall efficiency of the system.

In contrast to [5], in which the detection of APT attacks was based on the detection of malicious behavior patterns in network traffic, our work uses a multi-level detection system that makes it possible to effectively detect threats at all levels of the information system perimeter.

Unlike [6, 7], in which systems were designed that build compressed origin graphs that link events related to attacks, our work uses an AI assistant that, in addition to linking malicious activity between different alerts security, can provide context to expedite the investigation. Also, providing context about past events and infrastructure specifications helps quickly identify false positives.

In [9], in which a system for predicting APT attacks in network traffic based on machine learning was built, it showed similar accuracy results. But in contrast, the results of our work were achieved owing to the optimization of the use of machine learning for specific limited tasks.

Unlike [11], in which fuzzy clustering and the Bi-RNN algorithm are used to detect multi-level attacks, the current work uses a risk system and alert analysis with the help of an AI assistant, which in turn showed better results.

The proposed system solves the problem of detecting APT attacks, which often use significantly modified attack methods and threats masquerading as legitimate activity. This is achieved by integrating various detection methods. Antivirus solutions are complemented by correlation rules that allow detection of more complex threats, and machine learning makes it possible to effectively detect new types of anomalous activity in network traffic. The AI assistant adds an extra level of speed and accuracy to the incident investigation process, significantly improving the system's ability to respond to new and changing threats. With access to historical data about past incidents, one can quickly get information about the processes, teams, users, IP addresses, etc. that appeared in past incidents, as well as the relevant context, which significantly speeds up the investigation process.

The limitations of our study include the need for large computing power to operate the machine learning detection methods. In addition, worth noting are the limitations of publicly available datasets for training artificial intelligence. Also, although the AI assistant shows good results, it needs extremely detailed configuration.

Considering the shortcomings of the studied solution, it should be noted that for effective operation, the system requires constant support and updating. Modifying correlation rules and adding new ones could increase the number of detected threats, but it may also lead to an increase in system load and the number of false positives. In addition, correlation rules may duplicate the functions of already existing solutions or not bring additional benefits. As for machine learning, it requires regular updating of models, which requires up-to-date datasets to maintain detection accuracy. It is also important to note that an AI assistant needs constant training in order to effectively work with new data and correctly identify new threats.

As part of further research, the following can be considered:

– the possibility of implementing the generation of periodic reports on incidents for a certain period of time with their analysis by an AI assistant;

– integration of infrastructure data into an AI assistant to generate reports on potential vulnerabilities based on attack data.

7. Conclusions

1. Our research resulted in an architecture that combines conventional detection methods, such as signature analysis and correlation rules, with machine learning and artificial intelligence technologies. An important component of this architecture is the AI assistant, which has significantly improved the threat analysis process. Owing to its ability to analyze alerts and use historical data, the AI assistant has accelerated the incident investigation process by providing contextual information that makes it possible to faster and more accurate determine the nature of the threat. The combination of detection methods and an AI assistant made it possible to build a system adapted to new and modified threats, with high efficiency in detecting complex attacks. The difference from known solutions is the flexibility and ability to integrate different methods, which makes it possible to balance their strengths and weaknesses, providing more reliable protection.

2. The results showed that the defense-in-depth model based on multi-level architecture proved to be effective. The system built on it was able to detect 98 % of malicious files and 99 % of the tactics, techniques, and procedures used in APT attacks. This result is explained by the synergy of several levels of protection, where each level compensates for the shortcomings of the previous one. Owing to signature analysis, known threats were detected. Correlation rules handled most threats. Owing to machine learning, most of the threats that were missed by previous layers were detected. Efficiency in comparison with similar solutions is provided precisely by a comprehensive approach to protection.

3. The implementation of an AI assistant improved the process of threat analysis, significantly reducing the time of incident investigation. The AI assistant provides contextual information that makes it possible to determine the nature of the threat faster and more accurately. The difference from similar solutions is the ability to use historical data and context to speed up investigations and reduce the burden on analysts. Given this, the system provides more accurate detection of new threats, which increases its overall effectiveness.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

References

1. The swiss cheese model of security and why its important to have multiple layers of security. Firm Guardian. Available at: https://www.firmguardian.com/blog/swiss-cheese-model
2. McKee, F., Noever, D. (2023). Chatbots in a Botnet World. International Journal on Cybernetics & Informatics, 12 (2), 77–95. https://doi.org/10.5121/ijci.2023.120207
3. Ruby, A. R., Banu, A., Priya, S., Chandran, S. (2023). Taxonomy of AISecOps Threat Modeling for Cloud Based Medical Chatbots. arXiv. https://doi.org/10.48550/arXiv.2305.11189
4. Third-Party Cybersecurity Risk Management: A Short Guide for 2024. Available at: https://flare.io/learn/resources/blog/third-party-cybersecurity-risk-management/
5. Hassannataj Joloudari, J., Haderbadi, M., Mashmool, A., Ghasemigol, M., Band, S. S., Mosavi, A. (2020). Early Detection of the Advanced Persistent Threat Attack Using Performance Analysis of Deep Learning. IEEE Access, 8, 186125–186137. https://doi.org/10.1109/access.2020.3029202
6. Li, S., Dong, F., Xiao, X., Wang, H., Shao, F., Chen, J. et al. (2024). NODLINK: An Online System for Fine-Grained APT Attack Detection and Investigation. Proceedings 2024 Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2024.23204
7. Wang, N., Wen, X., Zhang, D., Zhao, X., Ma, J., Luo, M. et al. (2023). TBDetector:Transformer-Based Detector for Advanced Persistent Threats with Provenance Graph. arXiv. https://doi.org/10.48550/arXiv.2304.02838

8. Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., Djukic, P. (2022). Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats. ACM Computing Surveys, 55 (5), 1–37. https://doi.org/10.1145/3530812
9. Pham, V.-H., Nghi Hoang, K., Duy, P. T., Ngo Duc Hoang, S., Huynh Thai, T. (2024). Xfedhunter: An Explainable Federated Learning Framework for Advanced Persistent Threat Detection in Sdn. https://doi.org/10.2139/ssrn.4883207
10. Zhang, R., Sun, W., Liu, J.-Y. (2020). Construction of two statistical anomaly features for small-sample APT attack traffic classification. arXiv. http://dx.doi.org/10.48550/arXiv.2010.13978
11. Jia, B., Tian, Y., Zhao, D., Wang, X., Li, C., Niu, W. et al. (2021). Bidirectional RNN-Based Few-Shot Training for Detecting Multi-stage Attack. Information Security and Cryptology, 37–52. https://doi.org/10.1007/978-3-030-71852-7_3
12. Getting Started with Windows Security and Windows Defender. Institute for Advanced Study. Available at: https://www.ias.edu/security/getting-started-with-windows-security-windows-defender
13. Downloads. Available at: https://www.snort.org/downloads
14. About data models. Splunk. Available at: https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Aboutdatamodels
15. VMware Workstation Pro: Now Available Free for Personal Use. VMware Workstation Zealot. Available at: https://blogs.vmware.com/workstation/2024/05/vmware-workstation-pro-now-available-free-for-personal-use.html
16. Redcanaryco/atomic-red-team. Small and highly portable detection tests based on MITRE's ATT&CK. GitHub. Available at: https://github.com/redcanaryco/atomic-red-team
17. Piskozub, A., Zhuravchak, D., Tolkachova, A. (2023). Researching vulnerabilities in chatbots with LLM (large language model). Ukrainian Scientific Journal of Information Security, 29 (3), 111–117. https://doi.org/10.18372/2225-5036.29.18069
18. Sysmon v15.15. Available at: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon