Honeypot's Best Friend? Investigating ChatGPT's Ability to Evaluate Honeypot Logs

Berfin Ozkok
Sabanci University
Istanbul, TR
bozkok@sabanciuniv.edu

Baturay Birinci
Sabanci University
Istanbul, TR
baturaybirinci@sabanciuniv.edu

Orcun Cetin
Sabanci University
Istanbul, TR
orcun.cetin@sabanciuniv.edu

Budi Arief
University of Kent
Canterbury, UK
b.arief@kent.ac.uk

Julio Hernandez-Castro
Universidad Politécnica de Madrid
Madrid, Spain
jc.hernandez.castro@upm.es

ABSTRACT

Honeypots can gather substantial data from intruders, but many honeypots lack the necessary features to analyze and explain the nature of these potential attacks. Typically, honeypot analysis reports only highlight the attacking IP addresses and the malicious requests. As such, analysts might miss out on the more useful insights that can be derived from the honeypot data, for instance they might fail to fully examine the attackers' plan, or anticipate new threats from attackers. Meanwhile, recent advances in large language models (LLM) – such as the emergence of ChatGPT – have opened up the possibility of using artificial intelligence (AI) to comprehend honeypot data better. These recent advances suggest the possibility of an automated and intelligent log analysis that can explain consequences, provide labels, and deal with obfuscation using LLMs. In this study, we probed ChatGPT's proficiency in understanding and explaining honeypot logs from actual recorded attacks on our honeypots. Our data encompassed 627 requests to Elasticsearch honeypots and 73 attacks detected by SSH honeypots, collected over a two-week period. Our analysis was focused on evaluating ChatGPT's explanation ability regarding the potential consequences of each attack, in alignment with the MITRE ATT&CK Framework, and whether ChatGPT can identify any obfuscation techniques that might be used by attackers. We found that ChatGPT achieved a 96.65% accuracy in correctly explaining the consequences of the attack targeting Elasticsearch servers. Furthermore, ChatGPT achieved a 72.46% accuracy in matching a given attack to one or more techniques listed by the MITRE ATT&CK Framework. Similarly, ChatGPT was excellent in identifying obfuscation techniques employed by attackers and offering deobfuscation solutions. Specifically, we prompted ChatGPT with obfuscated data, and it successfully provided deobfuscated versions. However, 34.66% of the request body and 8% of the targeted URI were falsely identified as obfuscated, leading to a very high score of false positive for obfuscation. With the SSH data, we achieved a 97.26% accuracy while explaining the consequences of the attacks and a 98.84% accuracy for correctly mapping to MITRE ATT&CK Framework techniques. Based on these results, we can say that ChatGPT has shown great potential for automating the process of analyzing honeypot data. Its proficiency in explaining attack consequences and in managing obfuscation through implementing MITRE techniques is impressive. Nevertheless, it is essential to be mindful of the possibility of high false positive rates, which can cause some issues. This needs to be addressed in future research, for example by leveraging the advanced fine-tuning techniques that were recently introduced to ChatGPT, but not available at the time of writing of this paper.

KEYWORDS

Honeypot, ChatGPT, Artificial Intelligence, Digital Forensic, Log Analysis

1 INTRODUCTION

The evolution of cyber threats has led to an increasing demand for advanced cyber security measures and techniques to protect computer systems. One of these measures involve the utilisation of honeypots. Honeypots are network cybersecurity mechanisms that are set up as decoys to lure and monitor attackers trying to compromise them [1]. These systems are intentionally left as vulnerable targets, and they are strategically placed throughout the network to attract, observe, measure, and analyze malicious activities. As a decoy system, a honeypot draws potential attackers, while capturing valuable information about the attackers' exploits, attack methods, and targets. This information – stored in honeypot log files – allows security researchers to gain valuable insights about attackers' behavior, and such information can be used for creating better defensive mechanisms and countermeasures. However, reviewing and analyzing honeypot log files manually can be overwhelming and time-consuming, due to the high amount of data involved. Furthermore, hiring and training personnel to perform manual log analysis can be very expensive. Finally, manual analysis is prone to errors and omissions, which can cause incorrect results or important insights being overlooked. To address these issues, we explored alternative approaches for improving the efficiency and accuracy of honeypot log analysis, and this is the main aim of the study presented in this paper.

ChatGPT[1], an AI language model developed by OpenAI, has become one of the most popular online large language models (LLMs), and it could be used in almost every field. These types of tools have shown great potential to be used in tackling cybersecurity-related issues. However, its potential has only recently been explored and discovered. Several application areas have been discussed[2–7], but we are only scratching the surface.

[1]https://openai.com/blog/chatgpt

In this study, we explored ChatGPT's capability in evaluating Elasticsearch and SSH honeypot logs, emphasizing its ability to understand the consequences of the recorded attacks (and in aligning these attacks with the MITRE ATT&CK Framework), as well as in identifying obfuscation methods that attackers might use.

Contributions. The key contributions of our paper are:

- We have demonstrated that ChatGPT can greatly assist in the process of analyzing honeypot data/logs automatically.
- In particular, our results have shown ChatGPT's ability to explain potential consequences of recorded attacks captured by our honeypot logs.
- Finally, we have highlighted the possibility of using ChatGPT to detect any obfuscation techniques employed by the attackers, along with potential deobfuscation counter measure.

The rest of the paper is organized as follows. Section 2 discusses related work in this field. Section 3 explains the methodology we followed, especially our data set, the prompt structure, and the evaluation criteria. Section 4 presents our evaluation results, while Section 5 discusses the key implications of our findings. Finally, Section 6 concludes our paper and provides several suggestions for future work.

2 LITERATURE REVIEW

In this section, we briefly survey related studies of our research. The work on honeypots typically involves designing and developing better honeypots to collect intelligence from attackers. A few studies combine honeypots and LLMs, which have examined the gap between these two areas. Another relevant area to our research is ChatGPT usage in real life in digital forensics. In this area, researchers typically discover different uses of ChatGPT in cybersecurity.

2.1 ChatGPT Usage in Honeypots and Log Files

One of the earliest studies regarding the use of ChatGPT for analyzing honeypot log analysis is reported by Setianto *et al.* [8]. This was before the release of GPT-3.5, so instead, the authors conducted the honeypot log analysis using GPT-2. The paper mentions that honeypots produce lots of log data, consisting of Unix commands used by potential attackers. The processing of honeypot data is complicated since honeypots do not work well with specific tools. To address this issue, Setianto *et al.* made a tool that uses GPT-2 to understand logs from Cowrie SSH honeypot (https://github.com/cowrie/cowrie). This tool achieved an accuracy of 89% in inferring the incoming Linux commands, which indicates a clear potential of using ChatGPT as a beneficial tool for analyzing honeypot logs.

Similarly, in the context of log analysis, another paper by Petrović explains the DevSecOps, which solves security concerns in implementation and run-time steps using ChatGPT[9]. The paper focuses on improving run-time security and introduces a different approach using server log analysis and machine learning to detect suspicious activity. Unlike our paper, they did not analyze the log files to explain consequences or mapping attacks to the MITRE ATT&CK Framework.

Also, different studies using logs and ChatGPT were conducted, which is anomaly detection. One research is conducted by Egersdoerfer *et al.* [7]. The research investigated how to find complex run-time anomalies in production systems using log-based anomaly detection. They heavily rely on expert-labeled logs to identify behavior patterns. However, manually categorizing enough log data could take too long to train deep neural networks adequately. So, they created a system that works in two steps. The initial step is to take logs and summarize them with the next log windows. In the second step, GPT was fed by a window of logs and all summarized logs for anomaly detection. They compared the ChatGPT results to the results from using NeuralLog, DeepLog, and SentiLog. The results demonstrated that GPT-3.5-turbo achieved the highest performance [7].

Research by Liu *et al.* focuses on log analysis in software systems by dividing it into two sections: parsing and anomaly detection. However, the restricted predictability of analysis results undermines analysts' trust and capacity to take appropriate action in considering the increasing number of system events. They proposed the LogPrompt as a log analysis method, and it uses LLMs to perform zero-shot log analysis with advanced prompt strategies. They evaluated that the advanced prompt increases the LLM performance by up to 107.5%. In addition, they mentioned that one of the advantages of log analysis is explanations; by explaining with results, engineers can assess the credibility of anomalies and reduce the spending time [10].

In addition, another research [11] focuses on providing SeaLog, an accurate and adaptable log-based anomaly detection, to assess its performance on data sets. ChatGPT functioned as the study's expert consultant and offered suggestions for the SeaLog framework. The study used ChatGPT to provide comments on logs, and its performance was evaluated by contrasting its choices with those of human experts. It also minimizes the manual validation attempt. Similarly, we aimed to get insightful explanations from ChatGPT and reduce spending time and required effort.

One of the studies that shares the same motivation with our research conducted by Gupta *et al.* [6] mentions the opportunities and risks of GenAI (Generative AI), like ChatGPT. The research highlights the vulnerabilities of ChatGPT, how attackers can use ChatGPT to exploit the vulnerability, and defense techniques such as cyber defense automation, threat intelligence, attack explanation, malware detection, etc. By evaluating the data set, ChatGPT can give potential threats and attack explanations that organizations can use to make informed decisions about security-related activities. Also, ChatGPT can explain attacks by generating attack patterns and behaviors [6].

In this part, we discussed the application of ChatGPT in the context of honeypots, its potential benefits, and its impacts on cybersecurity. Honeypot logs are indeed essential for understanding the attacker's motivation, different techniques that are used, and behavior. A study by Ahmad *et al.* [12] discusses the possibility of using honeypots as a trap to deceive attackers and collect information about their behaviors. Also, the paper mentions the different honeypot systems and how they can be used to gather valuable data on attacks and take necessary prevention. This study aligns with our objectives and shares the common purpose of improving security infrastructure. Similar to this study, our research aims

to gather information, understand attack strategies, and analyze attack behaviors using honeypot logs to enhance security tools.

Although honeypot logs can provide insightful information, the evaluation of the honeypots can be time-consuming. One of the papers conducted by Mokube *et al.* [13] highlights the value of honeypots as a proactive security strategy, allowing businesses to collect information about potential attacks and improve their overall cybersecurity posture. However, manually analyzing vast amounts of honeypot data can be time-consuming and challenging. To overcome this limitation, researchers have turned to AI-based solutions. Our motivation is also similar to this approach; we aim to reduce the time spent and evaluate larger logs using the LLM model, ChatGPT.

Furthermore, we have examined relevant research to decide which type of honeypot interaction can be used in our research. One notable study by Kocaogullar *et al.* [1] presents a comparative analysis of two types of honeypots, which are high interaction and low interaction. High-interaction honeypots provide essential opportunities for gathering attack information and complete insights into their behavior. Low-interaction honeypots focus on particular vulnerabilities, providing broader coverage but limited interaction. By examining this research, we have decided that we can use high-interaction honeypot logs to discover different attack techniques and evaluate ChatGPT's performance on these log analyses.

One of the other essential studies is conducted by Mckee *et al.* [14] is about exploring the potential of using question-and-answer agents like ChatGPT as a tool for improving cybersecurity in a honeypot environment. Also, this study explains how to create a dynamic honeypot environment that can identify malicious activity and how ChatGPT imitates Linux, Mac, and Windows terminals to provide an interface for common tools.

Some recent studies used ChatGPT for log parsing and anomaly detection. One of the research conducted by Lee *et al.* [4] focused on evaluating ChatGPT's ability to correctly parse logs into structured data and its performance variations across different prompting methods. ChatGPT demonstrates valuable results in the evaluation of log parsing [4]. Another study shows ChatGPT is a promising way to analyze logs [15]. Also, Qi *et al.* states ChatGPT could be a beneficial tool for analyzing logs, and adds, as it increases the interpretability of analysis [3]. Our research does not evaluate ChatGPT's effectiveness in log parsing, we investigate the ability of ChatGPT to understand attack sequences, consequences, and whether it can detect obfuscation. By using appropriate prompts to perform log analysis, we can improve the correctness of our research.

2.2 LLMs in other aspects of Digital Forensics

There are several potential benefits and risks of using LLMs in digital forensics. One of the research conducted by Scanlon *et al.* [16] mentions these benefits. LLMs can be used for question answering, multilingual analysis, automated sentiment analysis, and automatic script generation. Also, the risks are bias, errors, and hallucinations, which means it focuses on answering without considering the correct answer. LLMs can be crucial in early threat detection systems by identifying instances, threats, phishing, and vulnerabilities. This ability supports investigations by allowing an approach to potential threats.

Several studies have examined the effects of using ChatGPT in the field. In a recent research conducted by Ozturk *et al.* [5], which is about a comprehensive comparison between the efficacy of AI-powered tools, specifically ChatGPT, and traditional static code analysis tools in identifying vulnerabilities in PHP code is presented. The study highlights that even the best-performing traditional static code analyzer, which had a maximum success rate of 32%, is not as successful at discovering vulnerabilities as ChatGPT, which has a success rate of 62-68%. In addition, research also highlights ChatGPT's high false positive rate of 91%, which is lower than the highest rate of 82% among traditional analyzers. The results indicate a novel approach for combining ChatGPT and other AI technologies with traditional static code analyzers to improve the efficiency of web application vulnerability detection.

A notable instance is the research by Henseler *et al.* [17] focused on whether it can ChatGPT assist legal professionals in conducting investigations, especially detecting cybercrime and managing digital components. Both studies demonstrate the potential of ChatGPT in assisting different parts of investigations and analysis within the field of technology and security, even though our primary focuses are distinct. ChatGPT's adaptability remains essential in determining the future of digital forensics and cybersecurity.

Additionally, the study conducted by Scanlon *et al.* [18] examines the utilization of ChatGPT in different digital forensic subjects and identifies its strengths, risks, and benefits. It is mentioned that ChatGPT could be used for identification and classification tasks such as network forensics, and malware investigations. Our study shares a common approach with them, which is improving digital forensics by using AI-driven solutions. Both studies highlight the effectiveness of ChatGPT in the context of digital forensics.

The research conducted by Sharma *et al.* [19] focused on examining current forms of cybersecurity threats and explores the utilization of AI and Big Data Analytics. ChatGPT is discussed as a beneficial tool for preventing cyber threats, as it can be used to identify security vulnerabilities. The research suggests the usage of the ChatGPT can be effective while evaluating cyber threats, and digital forensics can be used for investigating and analyzing cyber events before they occur.

One of the studies conducted by Sarker *et al.* [20] indicates the significance of AI-based techniques in solving current diverse security problems and provides a detailed overview. The study highlights several research directions within the scope of the study, which can aid researchers in future studies. The advantage of AI-driven cybersecurity is that it can potentially make computing more advanced and automated than current security solutions.

We have mentioned mostly the benefits of using ChatGPT. In contrast, the research conducted by Qammar *et al.* [2] evaluates ChatGPT to test against cybersecurity attacks, including its capability to generate malicious code and phishing emails. The study discusses the importance of digital forensics in investigating cyber crime related to chatbots. It suggests that addressing the vulnerabilities in ChatGPT requires specific strategies to prevent harmful actions and digital forensics can investigate cyber attacks and malicious actions.

3 METHODOLOGY

In this section, we will explain the rationale for choosing the Large Language Model (LLM) chatbot, the data sets to be used, the study procedure, and, lastly, the evaluation criteria.

3.1 ChatGPT

This study employed the most recent version of OpenAI's chatbot model, GPT-4. The chatbot has gained increasing interest for coding and debugging activities, a use-case emphasized by a debugging example showcased on the tool's official webpage[2]. Moreover, OpenAI provides API support for automation and tool development. Throughout our research, we employed OpenAI's API using the default settings of the gpt-4-0613 model.

3.2 Data Set

Our data sets consist of 2 weeks of private honeypots that emulate unsecured Elasticsearch and SSH services.

3.2.1 Elasticsearch Honeypot Data. We use the data from a scientific paper comparing low-interaction honeypots against high-interaction honeypots [1]. In that paper, 7284 unique Elasticsearch requests were captured by private high-interaction honeypots. In our study, we randomly selected a sample of 627 requests for this data set. Some of these requests were from Internet researchers like Census [3] and Shodan [4], and the rest were from attackers. Data in our honeypot log contains the following fields: timestamp, source IP and port of the request, body, content type, content length, header user agent, host and length, URI, request method, HTTP version, attacker location, honeypot type, cloud provider, and region. We used the request body, URI, and method fields of this data to analyze logs. They are the only fields that contain information about the conducted attacks' aim and purpose.

3.2.2 SSH Honeypot Data. SSH attack data was collected by a threat intelligence sharing website called a threat.gg[5]. This threat intelligence website deploys honeypots and shares incoming requests on its website. We aggregated two weeks of SSH attack data from August 2023, where 17,480 attack sequences were gathered. Data includes the attacker's IP address, country, date, SSH client version, command list that executed after the attacker compromised the system, and username and password tuple. The system allows user to enter whatever their username and password tuple. From the collected data, we extracted 73 unique command lists, and each of them was used in research.

3.3 Study Procedure

We investigated ChatGPT's capability to analyze log files in 3 main fields: (i) consequence explanation of attacks, (ii) associated MITRE ATT&CK Framework techniques, and (iii) dealing with obfuscation.

To carry out our study effectively, we initially focused on designing prompts that would be used as prompts in the system role while sending data to ChatGPT. These prompts were formulated to get information directly related to our study goals. Once the prompts were finalized, we utilized the API interface to send them to the ChatGPT. Subsequently, we collected and analyzed the responses generated by ChatGPT to further our understanding of its capabilities in the areas we were investigating.

First, we prompted ChatGPT to find the attacks' possible consequences. In order to do this in Elasticsearch honeypot logs, we ask ChatGPT to identify the attacks' impact using provided information such as attack URI endpoint destination, HTTP method used by the attacker, and request body. Also, we asked ChatGPT to consider possible responses toward that attack and consequences for the victimized system. Furthermore, for SSH honeypot logs, we requested ChatGPT to assess each command and explain the potential impact of this attack sequence on the victimized system.

Secondly, we asked ChatGPT to map the given attack with the relevant techniques from the MITRE ATT&CK Framework.

Lastly, we tasked ChatGPT with identifying obfuscations and the methods used in the provided data and deobfuscating them. To achieve this, we explicitly mentioned in the prompt: using the obfuscation method, find the deobfuscation technique and deobfuscate the provided data.

3.4 Evaluation

During the evaluation, ChatGPT responses are assessed according to the accuracy of ChatGPT in answering the questions. We evaluated our responses according to the following ruleset:

- *Consequence Explanation* We evaluated consequences on a three-point scale. One indicates that the explanations are inaccurate and fail to explain the attack's consequences correctly. Two means partial accuracy also signifies incorrect, irrelevant for the attack, or ambiguously explained. Three denotes a high level of accuracy, capturing essential facts. This evaluative framework serves to quantify the system's proficiency in delivering precise and informative explanations.
- *MITRE ATT&CK Framework Technique Mapping*
  We assessed ChatGPT's ability to accurately identify MITRE ATT&CK Framework v13 techniques using a four-category evaluation system. These are: Correct, denoting an exact mapping; Partial, indicating that the technique is accurate; however, sub-technique is not accurate; or the given technique may apply to the attack and usage of the attack, e.g., if T1105 Ingress Tool Transfer is correct for the attack, T1068 Exploitation for Privilege Escalation is partial due to transferred tool is not necessary for privilege escalation however could be used for; "Incorrect," signifying a complete irrelevance between the response and actual techniques; and "Deprecated," referring to techniques that are no longer placed under the current name or ID.
- *Dealing with Obfuscation*
  For the evaluation, a binary scoring system is used for evaluation. In the case of Elasticsearch responses, multiple criteria were employed to assess ChatGPT in handling obfuscation. Specifically, we investigated whether the request body and URI endpoint contained any obfuscated content, whether ChatGPT could identify such obfuscations, and whether it could perform deobfuscation on both the URI

[2]https://platform.openai.com/examples/default-fix-python-bugs
[3]https://search.censys.io/
[4]https://www.shodan.io/
[5]https://threat.gg/

Table 1: Distribution of Results

Category	Percentage
Correct	96.65%
Partial	0.32%
Incorrect	3.03%

endpoint and request body. For SSH responses, the evaluation was based on three specific criteria: obfuscation in commands, ChatGPT's ability to identify any such obfuscation, and its capacity to provide deobfuscation.

4 RESULTS

In the previous section, we explained an overview of our methodology and outlined our study procedure. As we clarified, our dataset consists of 2 weeks of Elasticsearch and SSH honeypot data. For the Elasticsearch requests, we analyzed a sample of 627 unique requests. Similarly, we evaluated 73 SSH attack sequences. We explored ChatGPT's log analysis performance in providing clear consequence explanations, mapping attacks to the MITRE ATT&CK Framework, and deobfuscating obfuscated attacks, with the percentage of correct, partial, and incorrect identification serving as our primary evaluation metric.

4.1 Evaluation of Elasticsearch Request Explanations

To evaluate the ChatGPT's efficacy of consequence explanation of corresponding request, we used a method that included sending task-specific prompts to ChatGPT, consisting of three essential parameters: (i) request body, (ii) URI endpoint, and (iii) HTTP method. By using these three fields, we queried ChatGPT to explain the potential consequences of such a request or attack activity. Once queried, we evaluated the results and categorized ChatGPT's responses into three unique labels: "correct," "partial," and "incorrect," each reflecting the quality of the prediction. Table 1 demonstrates that the distribution of these percentages reflects ChatGPT's different performance levels in providing attack consequence explanations in 627 Elasticsearch requests.

In Table 1, we can see that ChatGPT achieved 96.65% accuracy in correctly explaining the attack's consequences. Nearly 3.03% of the time, the attacks' consequences were partially described, missing essential details. Lastly, 0.32% of the ChatGPT consequences explanations were incorrect or empty.

As these results suggested, ChatGPT correctly understands and explains the consequences of complicated attacks. Since other security mechanisms observe similar attacks, they can use the same approach to explain the consequences of attacks in their logs.

4.2 Evaluation of SSH Attack Sequence Explanations

To explain the primary purpose of the SSH attack sequences, we prompted ChatGPT to explain each Linux command found in the sequence and, based on these explanations, provide a prediction for the consequence of the attack. In this evaluation, 73 SSH attack sequences were used as input.

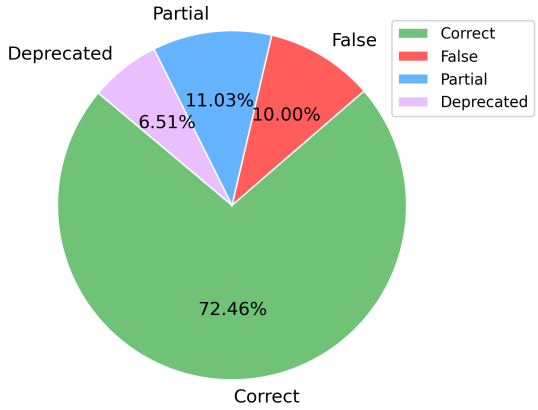

Figure 1: A Pie Chart of Elasticsearch Distribution of MITRE ATT&CK Framework

In particular, our results demonstrated that 97.26% (71 instances) of the ChatGPT output was correctly explained. This result highlights the model's ability to interpret the attackers' motivations within the scope of SSH attack sequences. In contrast, a mere 2.74% (2 instances) were partial, suggesting cases where the model's predictions were not aligned with the attacker's objectives, and there were no incorrect instances.

Surprisingly, the accuracy of ChatGPT in correctly explaining SSH attack sequences is high. The reason behind that can be many factors. SSH data have structured patterns with specific commands, parameters, and options. ChatGPT tends to identify attacks correctly in structured patterns. The methods used in SSH attacks are well-known and straightforward compared to Elasticsearch data. The standard terminology used in SSH data can contribute to ChatGPT's accuracy performance.

4.3 Efficacy of MITRE ATT&CK Framework Mapping

In this section, our primary goal is to assess ChatGPT's proficiency in classifying attacks and aligning them with the MITRE ATT&CK Framework. We have examined ChatGPT's output in 4 different categories: (i) correct; (ii) partial; (iii) deprecated, and; (iv) false. Correct is used when the request is directly related to the given mapping. Partial is used for requests that are not directly related to the given mapping but are indirectly related, or the technique is correct, but the sub-technique is incorrect. Deprecated is used for the given mapping item that may have been removed from the MITRE ATT&CK Framework. Alternatively, they could currently be represented by different, more relevant methods or code. Finally, false is used to classify requests where the responses generated by the ChatGPT do not match the actual or expected MITRE ATT&CK Framework item.

4.3.1 Evaluation of MITRE ATT&CK Framework Mapping in Elasticsearch. Figure 1 demonstrates the results of the distribution percentages of correct identification of MITRE ATT&CK Framework

mapping. Our findings concluded that MITRE ATT&CK Framework mapping was mainly successful. The percentage of labels identified as correct was 72.46% and partial was 11.03%; the total rate of these labels indicates that the LLM model predicted the requests mainly were correct. In addition, we have evaluated deprecated and false results; these are 6.51% and 10%, respectively. Deprecated results can be related to the ChatGPT's out-of-date data or upgrades to the MITRE ATT&CK Framework. In addition, we found instances of code mistakes and inconsistencies between the released attack descriptions and associated MITRE ATT&CK Framework code. In some cases, ChatGPT did not correctly identify the title or nature of the attack. Similarly, in some other false cases, MITRE ATT&CK Framework code assigns may not be correctly matched with the ones provided by ChatGPT.

4.3.2 Evaluation of MITRE ATT&CK Framework Mapping in SSH. We have also studied MITRE ATT&CK Framework mapping by using SSH attack sequences using identified Linux command sequence explanations. They have a clear pattern compared to Elasticsearch requests. Within the MITRE ATT&CK Framework, ChatGPT's performance was successful across all 73 attack sequence instances. According to the results we focused on, only one instance of a partial label was noted, and no examples of false and deprecated mappings were found. The accuracy of correct label mapping is 98.84%, and partial mapping is 1.16%.

The performance difference between analyzing Elasticsearch requests and SSH data is due to SSH data's relatively simple structure, which ChatGPT appears to understand more effectively. ChatGPT's attack sequence analysis accuracy may not match its attack explanation proficiency, but it still demonstrates above-average performance. These results demonstrate the strengths and weaknesses of ChatGPT when mapping attacks to the MITRE ATT&CK Framework. The complexity of its answers highlights its usefulness as a tool for comprehending challenging attack scenarios, even when considered in the context of its fundamental advantages and disadvantages.

In addition, the explanation performance in the SSH data is quite successful. In this case, the model demonstrates improved accuracy by using an attack explanation. The structure of SSH attack sequences makes it easier for the model to handle attack identifications. High-quality explanations enable more effective MITRE ATT&CK Framework mapping.

4.4 Efficacy of Obfuscation Identification and Deobfuscation

In this section, we evaluate ChatGPT's output for obfuscation detection and perform deobfuscation. We investigated the request body and URI fields and deobfuscated them if obfuscation was found. We evaluated the ability of ChatGPT to find Elasticsearch requests that contain obfuscation and deobfuscate them. The responses were evaluated manually to minimize mistakes. In our research, we have examined the false positive, which refers to instances where ChatGPT incorrectly identified the presence of obfuscation, even though obfuscation does not exist.

4.4.1 Evaluation of Obfuscation Identification and Deobfuscation for Elasticsearch Attacks. When we evaluate the response of 627 Elasticsearch requests, we have found that 76 instances contain obfuscation in their request body. Figure 2 demonstrates that the obfuscation counts in the Elasticsearch request body. Remarkably, ChatGPT managed to detect all obfuscated request bodies. However, ChatGPT only managed to deobfuscate nearly 91% (7 instances) of them. This shows that ChatGPT is very effective at detecting and dealing with obfuscation.

Reviewing the URI endpoints, we found 29 obfuscation instances in the request URL. ChatGPT demonstrated its classification capacity by correctly recognizing 29 instances while successfully deobfuscating 28 requests.

We also investigated false positives made by ChatGPT while looking for obfuscation. To perform this investigation, we used the data that excluded instances with obfuscation in both the request body and the URI.

Table 2: True Positive and False Positive Rates of Obfuscation Detection in Non-Obfuscated Requests

Obfuscation Type	True Positives	False Positives
Request Body	360 (65.34%)	191 (34.66%)
URI	551 (92.14%)	47 (7.86%)

Table 2 displays the false positive rates of obfuscation identification made by ChatGPT. We used 551 Elasticsearch instances in the request body and investigated whether ChatGPT correctly identified obfuscation status. ChatGPT misidentified nearly 34.66% of the Elasticsearch instances as obfuscated, while there is no obfuscation in the request body. Similarly, we have used 598 Elasticsearch requests in URI to evaluate false positive percentages. Around 7.86% of the plain URL requests are misidentified as obfuscated. These findings indicate that, although ChatGPT excels at accurately detecting obfuscated results, it can also produce a significant number of false positives.

Additionally, we explore deobfuscation methods for addressing false positives related to obfuscation in 2. We have mentioned 191 false positives in request-body obfuscation. Out of these 191 cases, in 136 instances, ChatGPT tried to deobfuscate the mistakenly identified request bodies. In the remaining 55 false positive cases, ChatGPT misidentified the obfuscation part but did not attempt to obfuscate the results. Furthermore, we investigated ChatGPT's false positive rate for the 598 URLs deobfuscation that were not obfuscated. ChatGPT misclassified 47 (7.86%) of the URLs as obfuscated. In addition to misclassified 47 instances, ChatGPT tried to deobfuscate 28 instances, without misclassification or obfuscation.

The number of false positives is interestingly high in both request body and uri obfuscation; we have investigated the reason behind these false positives. Our research revealed several issues that led to high false positive rates. One of them is the overuse of double quotation marks. ChatGPT is designed to improve content readability; sometimes, the overuse of double quotations is labeled as obfuscation. Encoding is one of the issues; the model could incorrectly predict encoding patterns, and incorrect identification can result from the sensitivity of the encoding patterns. Also, although URI parameters do not involve obfuscation, they are predicted as obfuscation in some cases. In addition to URI parameters, path traversal and JSON usage can be one of the challenging issues.

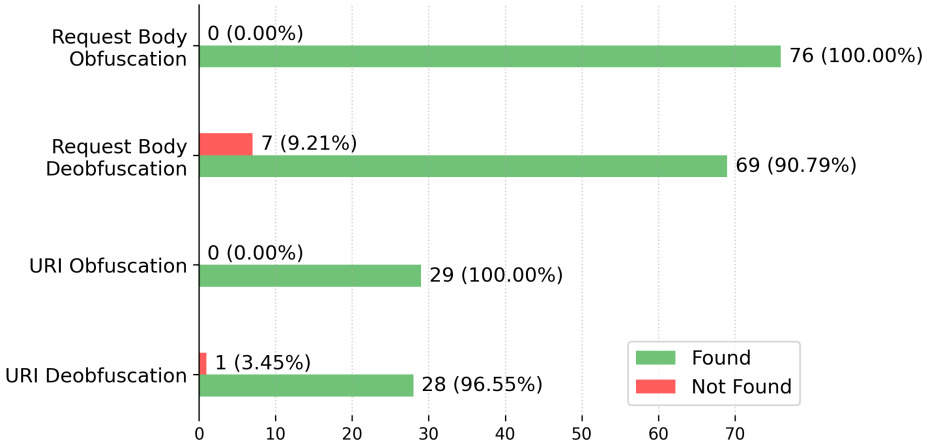

Figure 2: The Correctness of the ChatGPT Responses in Obfuscated Elasticsearch Requests

Table 3: True Positive and False Positive Rates of Obfuscation and Deobfuscation Detection in SSH Attack Sequences

Detection Type	True Positives	False Positives
SSH Obfuscation	59 (80.82%)	14 (19.18%)
SSH Deobfuscation	59 (80.82%)	14 (19.18%)

The existence of path traversal or JSON usage patterns can lead to misclassification; the model may not handle complex structures. In some cases, we have observed that script tags, such as XML, Java, and HTML, are incorrectly identified. This mistake can result from the model's emphasis on standardization and simplification of content. Due to the model's incorrect prediction of several factors, the number of false positives increased.

This detailed review highlights ChatGPT's ability to handle obfuscation issues in both the request body and URI fields. Although the deobfuscation has some problems, the model's overall performance demonstrates its potential for finding out sophisticated obfuscation situations, adding to the field of cybersecurity research and real-world applications.

4.4.2 Evaluation of SSH. We have evaluated the SSH attack sequence data for obfuscation. In only one case, we found an obfuscated sequence. This case was also successfully identified by ChatGPT. In Table 3, ChatGPT's precision was reflected in its accuracy, correctly identifying 59 instances while generating only 14 false positives in obfuscation and deobfuscation.

The number of false positives is high. There may be several reasons behind that; first of all, ChatGPT may misinterpret commands, parameters, options, or inputs for evaluation of obfuscation. Complex SSH patterns can lead to misinterpretation. Although it works successfully to detect obfuscation, false positives are high. Further improvement is required to reduce the number of false positives.

5 DISCUSSION

This section briefly presents the primary findings and discusses their implications for enhancing ChatGPT's proficiency in log analysis. In this research, we have analyzed the log files and evaluated the effectiveness of employing ChatGPT in explaining attack patterns, mapping attacks to MITRE ATT&CK Framework, and identifying and dealing with obfuscation. Through extensive evaluation, we have found that ChatGPT gives remarkably accurate and comprehensive responses in many cases. However, in various cases, false positive rates were equally high.

5.1 Increasing Effectiveness of ChatGPT's Log Analysis Ability

We have mentioned the reasons that can lead to making ChatGPT a false prediction. In addition, we did not fine-tune the ChatGPT model for our specific domain. If we still want to increase the accuracy rate to improve the ChatGPT model's performance in log analysis, we can use fine-tuning. Fine-tuning includes training the model on a specific dataset to perform more task-related and contextually appropriate tasks.

At the time we conducted the research, OpenAI was beginning to introduce the fine-tuning feature for GPT-3.5. However, we did not have a chance to use this feature yet in this paper. Applying fine-tuning with GPT-3.5 (or even, with GPT-4) could enhance performance by fitting the model's responses to cybersecurity threats. The usage of fine-tuning may contribute to the model's accuracy, and it evolves with different and sophisticated attack techniques. Accurate and efficient tools are required to investigate, understand, and respond to the threats. As the cybersecurity field continues to evolve, fine-tuning can play an essential role in developing ChatGPT to higher accuracy and efficiency.

5.2 Usefulness of ChatGPT in Honeypots and Regular Logs Analysis

As we come to the end of our study, we can mention that ChatGPT can be used as an effective tool to understand attacks, motivation,

and consequences. We demonstrated that ChatGPT could identify all instances of existing request body obfuscation and effectively address obfuscation challenges. Also, it can uncover Indicators of Compromise (IOCs) such as IP addresses, URLs, and malware existence, even if they are hidden with obfuscation techniques. The utilization of the MITRE ATT&CK Framework could further enhance its efficacy.

Furthermore, the scope of ChatGPT's applicability extends to low-interaction honeypots. These honeypots only capture logs of corresponding attacks; ChatGPT can be employed effectively to analyze these logs. We recommend employing fine-tuning to improve its effectiveness and minimize false positives. Beyond that, even now, without further development, ChatGPT still has much potential for log analysis, which could reduce the workload and spending time of security analysts.

6 CONCLUSION

In this study, we investigated ChatGPT's efficacy in analyzing Elasticsearch and SSH honeypot logs, focusing on clarifying the consequences of the attack, aligning with the MITRE ATT&CK Framework, and detecting obfuscation techniques. Through careful examination and manual validation, we determined that ChatGPT had an exceptional ability to produce accurate and insightful responses.

The research aimed to analyze logs of the Elasticsearch requests and SSH attacks through ChatGPT to explain the consequences of the corresponding attack using the request body, URIs, and HTTP method. The results demonstrate that ChatGPT has a remarkable investigation ability of identification, achieving a high accuracy rate of up to 96%. It highlights ChatGPT's competence in handling attack scenarios and providing reasonable explanations.

Moreover, in the results of SSH attack sequences, ChatGPT demonstrated a comprehensive understanding of the attack and the motivation while generating insightful explanations. The model's performance indicates its effectiveness as a tool for log analysis in the cybersecurity field.

Moving on the mapping MITRE ATT&CK Framework to the attacks, we found that ChatGPT managed to correctly identify all the SSH data and 73% of the Elasticsearch request. It suggests that ChatGPT has an impressive ability to identify and align SSH attacks with related MITRE ATT&CK Framework. However, this accuracy level was not succeeded from Elasticsearch requests; it could be the reason that the complexity of the Elasticsearch requests could have led to the model's low-level performance. In conclusion, the ChatGPT's performance in the mapping MITRE ATT&CK Framework is an important indicator as a beneficial resource in analyzing the log files.

Finally, we have measured the capacity of ChatGPT's obfuscation detection within request body, URIs, and SSH commands. However, we have faced some challenges; the rate of false positives was high. While ChatGPT was suitable for detecting obfuscation, the presence of false positives needs to be improved. Developing the model's understanding of cues and patterns can be beneficial for overwhelming the false positives.

In conclusion, our study reveals that ChatGPT is a potent instrument for log analysis. It demonstrates adeptness in analyzing logs, pinpointing attack consequences, mapping to the MITRE ATT&CK Framework, and recognizing obfuscation. This underscores ChatGPT's significant contributions to various facets of cybersecurity evaluation.

Moving forward, to enhance the insights provided by this article, we propose two key areas for further investigation: (i) exploring advanced fine-tuning techniques for Large Language Models (LLMs) to augment the log enrichment process; and (ii) delving into the efficacy of LLMs in the context of incident handling and response, evaluating their practical applications and impact.

When we conducted this research, OpenAI had not introduced the fine-tuning and assistant features. For future work, we could create a new assistant that knows our honeypot's capabilities and structure for getting fewer false positive results – or even, fewer false negative results as well.

Finally, fine-tuning with the MITRE ATT&CK Frameworks knowledge can lead to more actionable results, which can help security analysts in their job in dealing with the honeypot data.

REFERENCES

[1] Y. Kocaogullar, O. Cetin, B. Arief, C. Brierley, J. Pont, J. C. Hernandez-Castro, Hunting high or low: Evaluating the effectiveness of high-interaction and low-interaction honeypots (2022).

[2] A. Qammar, H. Wang, J. Ding, A. Naouri, M. Daneshmand, H. Ning, Chatbots to chatgpt in a cybersecurity space: Evolution, vulnerabilities, attacks, challenges, and future recommendations, arXiv preprint arXiv:2306.09255 (2023).

[3] J. Qi, S. Huang, Z. Luan, C. Fung, H. Yang, D. Qian, Loggpt: Exploring chatgpt for log-based anomaly detection, arXiv preprint arXiv:2309.01189 (2023).

[4] V.-H. Le, H. Zhang, An evaluation of log parsing with chatgpt, arXiv preprint arXiv:2306.01590 (2023).

[5] O. S. Ozturk, E. Ekmekcioglu, O. Cetin, B. Arief, J. Hernandez-Castro, New tricks to old codes: can ai chatbots replace static code analysis tools?, in: Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference, 2023, pp. 13–18.

[6] M. Gupta, C. Akiri, K. Aryal, E. Parker, L. Praharaj, From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy, IEEE Access (2023).

[7] C. Egersdoerfer, D. Zhang, D. Dai, Early exploration of using chatgpt for log-based anomaly detection on parallel file systems logs (2023).

[8] F. Setianto, E. Tsani, F. Sadiq, G. Domalis, D. Tsakalidis, P. Kostakos, Gpt-2c: A parser for honeypot logs using large pre-trained language models, in: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2021, pp. 649–653.

[9] N. Petrović, Machine learning-based run-time devsecops: Chatgpt against traditional approach, preprint (2023) 1–5.

[10] Y. Liu, S. Tao, W. Meng, J. Wang, W. Ma, Y. Zhao, Y. Chen, H. Yang, Y. Jiang, X. Chen, Logprompt: Prompt engineering towards zero-shot and interpretable log analysis, arXiv preprint arXiv:2308.07610 (2023).

[11] J. Liu, J. Huang, Y. Huo, Z. Jiang, J. Gu, Z. Chen, C. Feng, M. Yan, M. R. Lyu, Scalable and adaptive log-based anomaly detection with expert in the loop, arXiv preprint arXiv:2306.05032 (2023).

[12] W. Ahmad, M. Arsalan, S. Nawaz, F. Waqas, Detection and analysis of active attacks using honeypot, International Journal of Computer Applications 975 8887.

[13] I. Mokube, M. Adams, Honeypots: concepts, approaches, and challenges, in: Proceedings of the 45th annual southeast regional conference, 2007, pp. 321–326.

[14] F. McKee, D. Noever, Chatbots in a honeypot world, arXiv preprint arXiv:2301.03771 (2023).

[15] V.-H. Le, H. Zhang, Log parsing: How far can chatgpt go?, in: 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2023, pp. 1699–1704.

[16] M. Scanlon, B. Nikkel, Z. Geradts, Digital forensic investigation in the age of chatgpt, Forensic Science International: Digital Investigation 44 (2023).

[17] H. Henseler, H. van Beek, Chatgpt as a copilot for investigating digital evidence (2023).

[18] M. Scanlon, F. Breitinger, C. Hargreaves, J.-N. Hilgert, J. Sheppard, Chatgpt for digital forensic investigation: The good, the bad, and the unknown, arXiv preprint arXiv:2307.10195 (2023).

[19] P. Sharma, B. Dash, Impact of big data analytics and chatgpt on cybersecurity, in: 2023 4th International Conference on Computing and Communication Systems (I3CS), IEEE, 2023, pp. 1–6.

[20] I. H. Sarker, M. H. Furhad, R. Nowrozy, Ai-driven cybersecurity: an overview, security intelligence modeling and research directions, SN Computer Science 2 (2021) 1–18.