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

Meryem 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 analyse 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, such as the attackers' plan or emerging threats. Meanwhile, recent advances in large language models (LLM) – such as ChatGPT – have opened up the possibility of using artificial intelligence (AI) to comprehend honeypot data better, for instance, to perform an automated and intelligent log analysis that can explain consequences, provide labels, and deal with obfuscation. 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. However, 30.46% of the request body and 7.5% of the targeted URI were falsely identified as obfuscated, leading to a very high score of false positive for obfuscation. With the SSH honeypot 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 analysing honeypot data. Its proficiency in explaining attack consequences and in managing obfuscation through implementing MITRE ATT&CK 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.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; • **Computing methodologies → Artificial intelligence**.

KEYWORDS

Honeypot, ChatGPT, Artificial Intelligence, Log Analysis

ACM Reference Format:
Meryem Berfin Ozkok, Baturay Birinci, Orcun Cetin, Budi Arief, and Julio Hernandez-Castro. 2024. Honeypot's Best Friend? Investigating ChatGPT's Ability to Evaluate Honeypot Logs. In *European Interdisciplinary Cybersecurity Conference (EICC 2024), June 05–06, 2024, Xanthi, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3655693.3655716

1 INTRODUCTION

The evolution of cyber threats has led to an increasing demand for advanced cybersecurity measures and techniques to protect computer systems. One of these measures involves the utilisation of honeypots. Honeypots are network cybersecurity mechanisms that are set up as decoys to lure and monitor attackers trying to compromise them [4]. These systems are intentionally left as vulnerable targets, and they are strategically placed throughout the network to attract, observe, measure, and analyse 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' behaviour, and such information can be used for creating better defensive mechanisms and countermeasures. However, reviewing and analysing 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

EICC 2024, June 05–06, 2024, Xanthi, Greece

ACM ISBN 979-8-4007-1651-5/24/06
https://doi.org/10.1145/3655693.3655716

or important insights to be 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, 3, 5, 11, 13, 14], but we are only scratching the surface.

In this study, we explored ChatGPT's capability in evaluating Elasticsearch and SSH honeypot logs, emphasising 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 analysing honeypot data/logs automatically.
- In particular, our results have shown ChatGPT's ability to explain the 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 and potential deobfuscation techniques.

The rest of the paper is organised 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

We first surveyed papers related to our research. The work on honeypots typically involves designing and developing better honeypots to collect intelligence from attackers. A few studies have combined honeypots and LLMs and examined the gap between these two topics. Another relevant area is the real-life usage of ChatGPT 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 analysing honeypot log analysis is reported by Setianto *et al.* [18]. This was before the release of GPT-3.5, so instead, the authors conducted the honeypot log analysis using GPT-2. 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[2]. 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 analysing honeypot logs.

[1]https://openai.com/blog/chatgpt
[2]https://github.com/cowrie/cowrie

Similarly, Petrović explains DevSecOps, which solves security concerns in implementation and run-time steps using ChatGPT [12]. 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 analyse the log files to explain consequences or mapping attacks to the MITRE ATT&CK Framework.

One of the studies about anomaly detection by using logs and ChatGPT has been conducted by Egersdoerfer *et al.* [2], who investigated how to find complex run-time anomalies in production systems using log-based anomaly detection. They heavily relied on expert-labeled logs to identify behavioural patterns. However, manually categorising 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 summarise them with the next log windows. In the second step, GPT was fed by a window of logs and all summarised logs for anomaly detection. They compared the ChatGPT results to those from using NeuralLog, DeepLog, and SentiLog. The results demonstrated that GPT-3.5-turbo achieved the highest performance [2].

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% [8].

In addition, another research [7] 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 minimises the manual validation attempt. Similarly, we aimed to get insightful explanations from ChatGPT and reduce spending time and required effort.

Gupta *et al.* [3] explains how attackers can use ChatGPT to exploit the vulnerability, along with defense techniques such as cyber defense automation, threat intelligence, attack explanation, and malware detection. By evaluating the data set, ChatGPT can give potential threats and attack explanations that organisations can use to make informed decisions about security-related activities.

In this paper, we explore the use of ChatGPT in the context of honeypots, its potential benefits, and impacts on cybersecurity. Related to this, Ahmad *et al.* [1] discusses the possibility of using honeypots as a trap to deceive attackers and collect information about their behaviours. The paper also mentions the different honeypot systems and how they can be used to gather valuable data on attacks and take necessary prevention. Similarly, our research aims to gather information, understand attack strategies, and analyse attack behaviours using honeypot logs to enhance security tools.

Although honeypot logs can provide insightful information, the evaluation of the honeypots can be time-consuming. Mokube *et al.* [10] 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 analysing 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.* [4] 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 behaviour. 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 was conducted by Mckee *et al.* [9], which explored 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. Lee *et al.* [5] 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 [5]. Another study shows ChatGPT is a promising way to analyse logs [6]. Also, Qi *et al.* states ChatGPT could be a beneficial tool for analysing logs, and adds, as it increases the interpretability of analysis [14]. 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.

One of the recent studies conducted by Sladić *et al.* [20] describes the novel approach, applying the dynamic and adaptable nature of LLMs to develop convincing honeypots named shelLM. They have experimented with attackers to evaluate the realism of each command that comes from the system by labeling it fake or not, the accuracy of the experiment is 0.92. This study shares a similar context. However, we aimed to evaluate the attacker's behaviour using attack sequences and we used existing honeypot logs to analyse the performance of the ChatGPT on attacker behaviour, these two studies use different types of inputs.

2.2 LLMs in other aspects of Digital Forensics

There are several potential benefits and risks of using LLMs in digital forensics. Scanlon *et al.* [17] mentions these benefits: LLMs can be used for question answering, multilingual analysis, automated sentiment analysis, and automatic script generation. However, there are risks of 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.

In a recent research conducted by Ozturk *et al.* [11], which is about a comparison between the efficacy of AI-powered tools 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 analyser, 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 analysers. The results indicate a novel approach for combining ChatGPT and other AI technologies with traditional static code analysers to improve the efficiency of web application vulnerability detection.

Additionally, the study conducted by Scanlon *et al.* [16] examines the utilisation 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: improving digital forensics 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 cybersecurity threats and exploring the utilisation of AI and Big Data Analytics. The research suggests the usage of the ChatGPT can be effective while evaluating cyber threats, and digital forensics can be used for investigating and analysing cyber events before they occur. Similarly, one of the studies conducted by Sarker *et al.* [15] 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.

We have mentioned mostly the benefits of using ChatGPT. In contrast, the research conducted by Qammar *et al.* [13] 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 emphasised by a debugging example showcased on the tool's official webpage[3]. 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]https://platform.openai.com/examples/default-fix-python-bugs

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 data from a scientific paper comparing low-interaction honeypots against high-interaction honeypots [4]. In that paper 7,284 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. 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 analyse 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[4]. 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 analyse 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. We have implemented an iterative process of prompt development which involves evaluating the effectiveness of prompts by using our observations as feedback. These prompts were formulated to get information directly related to our study goals. Once the prompts were finalised, we utilised the API interface to send them to ChatGPT. Subsequently, we collected and analysed the responses generated by ChatGPT to further our understanding of its capabilities in the areas we were investigating.

3.3.1 System prompts for analysis. Our system prompt has three distinct parts. The first part highlights the expected output format. The second part introduces the user input, providing the context for the data that will be analysed, and finally, the third part contains the expected analysis, which is described below:

- For the Elasticsearch honeypot dataset, we asked:
  - Identification of the attackers' aim, motivation, and possible responses from the victimised system by analysing the 'URI,' 'request body,' and 'method' fields.
  - Mapping of MITRE ATT&CK Framework instance matches with the observed attack pattern.
  - Identification and reversing any obfuscation techniques employed within the 'data' and 'URI' fields, respectively.
- For the SSH honeypot dataset, our prompts were asking:
  - List of each command was executed with the provided command sequence and identification of their parameters, options, and inputs.
  - Assessment of each command and explain the potential impact of this attack sequence on the victimised system.
  - Examines the attack sequences and matches the corresponding MITRE ATT&CK Framework.
  - Identification and reversing any obfuscation techniques employed within the data.

3.4 Evaluation

During the evaluation, ChatGPT responses are assessed according to the accuracy of ChatGPT in answering the questions. Two researchers independently worked on the log evaluation manually to mitigate the risk of misinterpreting ChatGPT's responses. 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 (it signifies incorrect, irrelevant for the attack, or ambiguously explained). Three denotes a high level of accuracy, capturing essential facts. This three-point scale serves to quantify the system's proficiency in providing 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*
  Obfuscation is a method that is applied to make information more difficult to interpret. It can be implemented either to protect sensitive information or to conceal its actual intent. For obfuscation evaluation, a binary scoring system is used. 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 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]https://threat.gg/

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 analysed 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 categorised ChatGPT's responses into three unique labels: "correct," "partial," and "incorrect," each reflecting the quality of the prediction.

Interestingly, 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.

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.

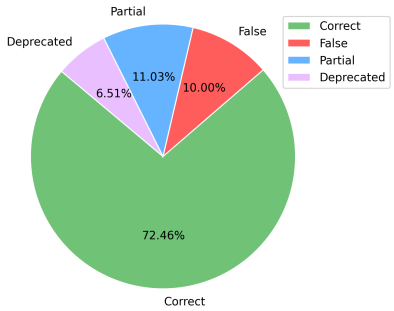

Figure 1: A Pie Chart of Elasticsearch Distribution of MITRE ATT&CK Framework

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 codes were mismatched with names, such as ChatGPT said *"... T1615 Server Software Component* because ...". However, in response, the code of **Server Software Component** was mismatched with Group Policy Discovery's code, which are T1505 and T1615, respectively.

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

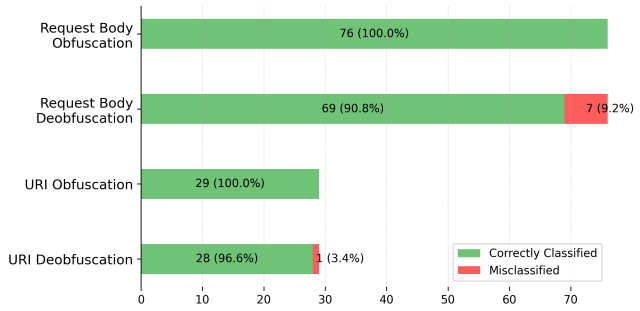

Figure 2: The Correctness of the ChatGPT Responses in Obfuscated Elasticsearch Requests

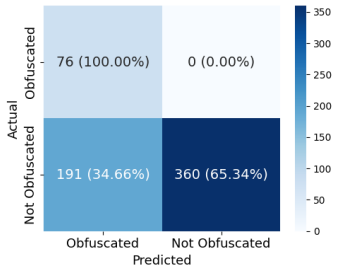

Figure 3: Confusion Matrix of Request Body Obfuscation Detection in Elasticsearch Requests

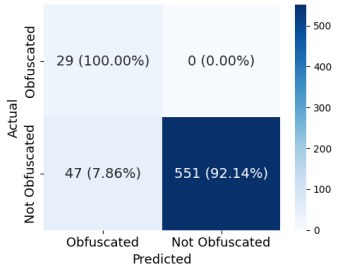

Figure 4: Confusion Matrix of URI Obfuscation Detection in Elasticsearch Requests

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 analysing 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 responses for obfuscation detection and deobfuscation performance for both Elasticsearch requests and SSH attacks. 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. We evaluated the ability of ChatGPT to find Elasticsearch requests that contain obfuscation in URI and request body and deobfuscate them. When evaluating the response of 627 Elasticsearch requests, we found that 76 instances contained obfuscation within their request bodies.

Figure 2 shows the number of obfuscated instances in the Elasticsearch request body. Remarkably, ChatGPT managed to detect all obfuscated request bodies. However, ChatGPT managed to deobfuscate nearly 91% (69 instances) of them. This shows that ChatGPT is very effective at detecting and dealing with obfuscation.

Figure 2 also displays the URI endpoints; we found 29 obfuscation instances in the request URL. ChatGPT demonstrated its classification capacity by correctly recognising 29 instances while successfully deobfuscating 28 requests.

However, while ChatGPT excelled at identifying obfuscated content, there were notable instances of false positives. In the context of the request body, ChatGPT incorrectly classified 191 instances, which constituted 30.46% of analysed responses, as obfuscated when they were not. The false positive rate for the request body obfuscation detection is 34.66%.

The confusion matrix for the request body obfuscation detection, as presented in Figure 3, demonstrates that ChatGPT correctly identified all 76 obfuscated request bodies. This achievement reflects a 100% sensitivity for obfuscation detection in the request body, highlighting ChatGPT's ability to identify obfuscated content with no instances of false negatives reported.

Figure 4 shows the confusion matrix of obfuscation identification in URI, in which 47 instances were incorrectly identified as obfuscated and 551 instances were correctly identified as not 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 Figure 3. We have mentioned 191 false positives in request-body obfuscation. Out of these 191 cases, in 136 instances, ChatGPT tried to deobfuscate the mistakenly

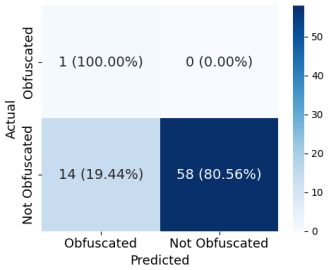

Figure 5: Confusion Matrix of Obfuscation Detection in SSH Attack Sequences

identified request bodies. In the remaining 55 false positive cases, ChatGPT misidentified the obfuscation part but did not attempt to deobfuscate the results.

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. 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 standardisation 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.

The confusion matrix presented in Figure 5 illustrates ChatGPT's performance regarding obfuscation detection, which shows that ChatGPT's precision was reflected in its accuracy, correctly identifying 59 instances. This total includes the one true positive instance, where obfuscation was accurately detected and deobfuscated, and 58 true negative instances, where the absence of obfuscation was correctly identified and did not attempt to deobfuscate.

However, the analysis also revealed 14 false positive instances, constituting 19.18% of the evaluated data. These instances represent scenarios where ChatGPT mistakenly identified sequences as obfuscated when they were not, leading to incorrect attempts at 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, the reason can be the complexity of SSH patterns. 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 analysed 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. One of the important aspects of this study is leveraging ChatGPT in a zero-shot inference, where the model is asked to interpret recent and previously unseen data. It highlights that ChatGPT can broaden its pre-trained knowledge to new contexts without prior experience. 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 ChatGPT to make 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 utilisation 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 analyse these logs. We recommend employing fine-tuning to improve its effectiveness and minimise 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 analysing 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 analyse 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 analysing 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 analysing logs, pinpointing attack consequences, mapping to the MITRE ATT&CK Framework, and recognising 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] Waqas Ahmad, Muhammad Arsalan Raza, Sabreena Nawaz, and Farhana Waqas. 2023. Detection and Analysis of Active Attacks using Honeypot. *International Journal of Computer Applications (0975 – 8887)* 184, 50 (2023), 27–31.

[2] Chris Egersdoerfer, Di Zhang, and Dong Dai. 2023. Early Exploration of Using ChatGPT for Log-based Anomaly Detection on Parallel File Systems Logs. In *Procs 32nd Int'l Symp. on High-Perf. Parallel and Distributed Computing*. 315–316.

[3] Maanak Gupta, CharanKumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. 2023. From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access* (2023).

[4] Yekta Kocaogullar, Orcun Cetin, Budi Arief, Calvin Brierley, Jamie Pont, and Julio C Hernandez-Castro. 2022. Hunting High or Low: Evaluating the Effectiveness of High-Interaction and Low-Interaction Honeypots. In *12th Int'l Workshop on Socio-Technical Aspects in Security (STAST 2022)*.

[5] Van-Hoang Le and Hongyu Zhang. 2023. An Evaluation of Log Parsing with ChatGPT. *arXiv preprint arXiv:2306.01590* (2023).

[6] Van-Hoang Le and Hongyu Zhang. 2023. Log Parsing: How Far Can ChatGPT Go?. In *2023 38th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE)*. IEEE, 1699–1704.

[7] Jinyang Liu, Junjie Huang, Yintong Huo, Zhihan Jiang, Jiazhen Gu, Zhuangbin Chen, Cong Feng, Minzhi Yan, and Michael R Lyu. 2023. Scalable and Adaptive Log-based Anomaly Detection with Expert in the Loop. *arXiv preprint arXiv:2306.05032* (2023).

[8] Yilun Liu, Shimin Tao, Weibin Meng, Jingyu Wang, Wenbing Ma, Yanqing Zhao, Yuhang Chen, Hao Yang, Yanfei Jiang, and Xun Chen. 2023. LogPrompt: Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis. *arXiv preprint arXiv:2308.07610* (2023).

[9] Forrest McKee and David Noever. 2023. Chatbots in a Honeypot World. *arXiv preprint arXiv:2301.03771* (2023).

[10] Iyatiti Mokube and Michele Adams. 2007. Honeypots: Concepts, Approaches, and Challenges. In *Procs 45th Annual Southeast Regional Conference*. 321–326.

[11] Omer Said Ozturk, Emre Ekmekcioglu, Orcun Cetin, Budi Arief, and Julio Hernandez-Castro. 2023. New Tricks to Old Codes: Can AI Chatbots Replace Static Code Analysis Tools?. In *Procs 2023 European Interdisciplinary Cybersecurity Conference*. 13–18.

[12] Nenad Petrović. 2023. Machine Learning-Based Run-Time DevSecOps: ChatGPT Against Traditional Approach. In *10th Int'l Conf. on Electrical, Electronic and Computing Engineering (IcETRAN)*. IEEE, 1–5.

[13] Attia Qammar, Hongmei Wang, Jianguo Ding, Abdenacer Naouri, Mahmoud Daneshmand, and Huansheng Ning. 2023. Chatbots to ChatGPT in a Cybersecurity Space: Evolution, Vulnerabilities, Attacks, Challenges, and Future Recommendations. *arXiv preprint arXiv:2306.09255* (2023).

[14] Jiaxing Qi, Shaohan Huang, Zhongzhi Luan, Carol Fung, Hailong Yang, and Depei Qian. 2023. LogGPT: Exploring ChatGPT for Log-Based Anomaly Detection. *arXiv preprint arXiv:2309.01189* (2023).

[15] Iqbal H Sarker, Md Hasan Furhad, and Raza Nowrozy. 2021. AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. *SN Computer Science* 2 (2021), 1–18.

[16] Mark Scanlon, Frank Breitinger, Christopher Hargreaves, Jan-Niclas Hilgert, and John Sheppard. 2023. ChatGPT for Digital Forensic Investigation: The Good, The Bad, and The Unknown. *arXiv preprint arXiv:2307.10195* (2023).

[17] Mark Scanlon, Bruce Nikkel, and Zeno Geradts. 2023. Digital forensic investigation in the age of ChatGPT. *Forensic Sci. Int'l: Digital Investigation* 44 (2023).

[18] Febrian Setianto, Erion Tsani, Fatima Sadiq, Georgios Domalis, Dimitris Tsakalidis, and Panos Kostakos. 2021. GPT-2C: A Parser for Honeypot Logs Using Large Pre-trained Language Models. In *Procs 2021 IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining*. 649–653.

[19] Pawankumar Sharma and Bibhu Dash. 2023. Impact of Big Data Analytics and ChatGPT on Cybersecurity. In *2023 4th Int'l Conf. on Computing and Communication Systems (I3CS)*. IEEE, 1–6.

[20] Muris Sladić, Veronica Valeros, Carlos Catania, and Sebastian Garcia. 2023. LLM in the Shell: Generative Honeypots. *arXiv preprint arXiv:2309.00155* (2023).