When ChatGPT Meets Vulnerability Management: the Good, the Bad, and the Ugly

Kylie McClanahan, Sky Elder, Marie Louise Uwibambe, Yaling Liu, Rithyka Heng, Qinghua Li
University of Arkansas
{klmcclan, wselder, uwibambe, yl050, rkheng, qinghual}@uark.edu

***Abstract*—Vulnerability management is a very challenging and time-consuming task. For many organizations, security operators need to learn about the properties of vulnerabilities to prioritize and mitigate them. Due to the lack of automated tools for vulnerability assessment, operators usually manually search for and read related information from sources online. Recent advances in large language models, like ChatGPT, open up an opportunity for time savings and may prompt operators to use these models as vulnerability information sources. In this work, we evaluate the ability of ChatGPT and several of its siblings to accurately answer user questions about vulnerability properties as well as to provide information for how to mitigate a vulnerability. We also explore their summarization capabilities when multiple vulnerability advisory documents are provided. We find that the models perform poorly on information retrieval tasks, but they perform quite well on summarization.**

***Index Terms*—ChatGPT, Large Language Models, Vulnerability**

I. Introduction

Vulnerability management is an involved and time-intensive process. In many organizations, operators need to understand the risks of vulnerabilities, prioritize them, and patch them or mitigate them with other measures [1]. In the absence of automated tools, operators usually find information about vulnerabilities in their systems through a primarily manual process. For example, in the electric sector specifically, the Critical Infrastructure Protection (CIP) regulations require that operators assess each of their devices for vulnerabilities every 35 days, which is a very challenging task given the large number of vulnerabilities to deal with. Thus, there is a strong need to relieve the time burden of this vulnerability assessment on security operators.

Recent advances in large language models (LLMs), like the GPT family of models created by OpenAI [2], have demonstrated incredible potential in answering user-prompted questions by synthesizing and generating answers based on what it has learned from past online data resources.

To reduce the time needed to obtain and digest intelligence about vulnerabilities, time-strapped security operators might wonder whether they can use ChatGPT and its siblings to directly obtain vulnerability information. ChatGPT seems an exciting prospect, given the success of LLMs in other domains. However, public perception of these models may not fully recognize their limitations, such as their tendency to *hallucinate* (provide incorrect information stated as truth). Thus, a dedicated study is needed.

This paper aims to explore the ability of GPT models in solving vulnerability management challenges. Specifically, we perform empirical studies to assess whether GPT models can correctly answer vulnerability-related questions that are important in vulnerability management, such as what the Common Vulnerability Scoring System (CVSS) score and vector are for a vulnerability, which products are affected, and how to mitigate the vulnerability. We also study whether GPT models can synthesize and summarize vulnerability intelligence from vulnerability advisory articles.

To the best of our knowledge, this is the first work that studies the applicability and effectiveness of LLMs in vulnerability management from a security operator's perspective. Our study illuminates effective and ineffective ways to use these models for vulnerability management automation. It serves as both an informative evaluation of LLM capabilities and a cautionary tale against relying on these models for factual accuracy.

This paper is organized as follows. Section II introduces background and reviews related work. Section III presents the study on retrieving CVSS and vulnerable product information. Section IV describes the study on retrieving mitigation information. Section V presents our study on summarizing vulnerability advisories. The last section concludes the paper.

II. Background and Related Work

A. Vulnerability Information

The National Vulnerability Database (NVD) is the leading repository of structured vulnerability data; it is maintained by the United States government and is widely used by industry and academia. All vulnerabilities published to the NVD are assigned a unique Common Vulnerabilities and Exposures (CVE) identifier.

The Common Platform Enumeration (CPE) helps to associate vulnerabilities with products. A CPE string is a structured representation of a system, software, or product. It typically specifies the vendor name, the product name, and the version; other attributes are available but are populated less frequently. Each CVE includes a listing of vulnerable CPEs in a one-to-many relationship.

Vulnerabilities are evaluated using the Common Vulnerability Scoring System (CVSS). CVSS v3 uses eight categorical metrics to score a vulnerability's severity. Together, these metrics comprise the CVSS vector. Once values have been assigned to all eight, they are converted to numerical values

and used to calculate the Base Score for the CVE, which falls between 0.0 and 10.0. The base score is commonly used as a risk metric for vulnerabilities.

B. Large Language Models (LLMs)

LLMs have garnered significant attention in recent months for their ability to hold conversations and answer questions in natural language. However, these models are particularly prone to *hallucinations*, where the model generates incorrect text and states it as truth [3, 4].

LLMs are, at their core, predictive models able to generate natural language based on statistical probabilities. They have impressive reasoning capabilities, but they are not factual databases. For topics that occur frequently in their training data, LLMs may be able to provide correct information. A bystander, then, might expect that publicly-available LLMs are capable of answering questions about more advanced or niche topics. The distinction between retrieving facts from a verified source and arriving at it through statistical probability is often overlooked in the public perception of LLMs like ChatGPT.

There have been many stories about ChatGPT providing users with the names of academic papers [5], news articles [6], blog posts [7], or even legal cases [8] that do not exist. In some of these cases, the user only learns they are fake when, unable to locate the (fictional) source material, they reach out to the internet for help. Despite OpenAI's warnings against overreliance, public perception of LLMs is more similar to a search engine than a generative tool.

In this work, four LLMs are considered. **1) GPT-3**: the `text-babbage-001` model from OpenAI (since deprecated). First released in June of 2020, it is capable of simple tasks but not at maintaining chat context. **2) GPT-3.5**: the `gpt-3.5-turbo` model from OpenAI. When first released in November 2022, ChatGPT was built on the GPT-3.5 model. **3) GPT-4**: the `gpt-4` model from OpenAI released in March of 2023. **4) Bing Chatbot**: In May of 2023, Microsoft's Bing browser released a chatbot with GPT-4 as the engine. Unlike the models from OpenAI, the Bing chatbot can perform real-time Internet searches to answer questions. When describing experiments that were tested on multiple of these models, the generic "GPT" is used in this paper.

C. Related Work

Much of the existing work in the vulnerability management space focuses on predicting such metrics as likelihood of exploitation [9, 10, 11], severity [12, 13], or potential impact [14, 15, 16]. There is also work in automated vulnerability remediation [17], automated vulnerability safety analysis taking into firewall policy into consideration [18], automated vulnerability tracking [19], and automated mapping of advisories to CPEs [20]. There is limited work to automatically locate and retrieve existing vulnerability information for security operators; one work, [21], locates mitigation information in reference websites published alongside CVEs in the NVD.

In their technical report for GPT-4, OpenAI discussed some possible cybersecurity applications: vulnerability discovery, exploit code generation, and social engineering [3]; retrieval of vulnerability information, which is the topic of this paper, was not discussed.

There have been efforts to evaluate ChatGPT's ability to answer questions or perform tasks requiring domain-specific knowledge. [22] reported an accuracy of around 50% on an ophthalmology exam. The translation capabilities of GPT-4 are reviewed in [23], while [24, 25] trained a LLM on a mixture of general and financial-sector data. There is also work surrounding the hallucinations generated by LLMs. Some works attempt to perform detection [26]; others investigate how and why hallucinations happen [27, 28]. [29] proposes a general framework for measuring hallucinations in LLMs, and [30] demonstrates an external system to augment ChatGPT and to mitigate its hallucinations. There is currently no mechanism in the OpenAI web or API interface to detect or alert the user when hallucinations occur.

III. Retrieving CVSS and CPE Information

In this section, we empirically study whether ChatGPT can retrieve the CVSS score, CVSS vector, and CPEs for vulnerabilities. We consider an application scenario where a security operator wants to learn about a particular vulnerability. Instead of manually searching for such information online, the operator employs ChatGPT for a quick answer.

A. Dataset

For reproducibility, the dataset for this section consists of the first 15 vulnerabilities published each month between Jan. 2016 and Dec. 2018, totaling 540. The date range was chosen to begin after the adoption of CVSS v3 in December 2015 and to end before the models' training data cutoff, the earliest of which is October 2019. We specifically chose only vulnerabilities first published in this window (as opposed to those published earlier but modified within our window) to avoid bias against the GPT models. Using the NVD API, we collected ground truth data for comparison, including CVSS scores, CVSS vectors, and vulnerable CPEs.

B. Approach

In this task, we evaluate GPT-3, GPT-3.5, GPT-4, and the Bing chatbot. Models were asked questions about the CVSS score, CVSS vector, and affected CPEs. To limit variability, the same phrasing was used for each vulnerability and can be seen below.

- What is the CVSS v3 score for [CVE]?
- What is the CVSS v3 vector for [CVE]?
- What CPEs are affected by [CVE]?

As an additional measure, the *temperature* of the model, a parameter which controls randomness, was set to 0 whenever possible. This parameter was not available for the Bing chatbot but could be set for the three OpenAI models.

Apart from one experiment with a "jailbreak" script described in Section III-D, we intentionally did not use any additional text to prepare the model before asking questions. While such prompts can be extremely effective (and are used

in the experiments described in Section V), we felt their use could misconstrue the accuracy of the LLMs evaluated, as it is unrealistic to expect that the average operator would be aware of the need for such prompts.

C. Evaluation Criteria

1) CVSS Score and CVSS Vector: For CVSS scores and vectors, the same evaluation categories could be used.

Correct: the returned CVSS score/vector exactly matched the CVSS score/vector found in the NVD in value and format.

Incorrect: the score/vector returned did not match that from the NVD, but the format was still correct. For CVSS scores, this was a number between 0.0 and 10.0 with only one decimal place. For CVSS vectors, this was a vector matched by the regular expression in [31].

Impossible: the score/vector returned was incorrect and did not match the format of the score/vector found in the NVD.

2) CPE: Originally, we planned to evaluate the models' abilities to return CPE identifiers. However, none of the models could return properly-formed CPE strings and instead gave vendors and products that the model predicted were affected by the vulnerability. To address this, the original evaluation criteria were adapted as follows, considering that multiple <vendor, product> pairs might be associated with one CVE:

Fully Correct: all vulnerable <vendor, product> pairs for a CVE are returned by GPT, and no other pairs are returned.

Correct but Incomplete: at least one <vendor, product> pair in the CVE is returned by GPT, but at least one pair in the CVE is not returned, and no pair not associated with the CVE is returned.

Partially Incorrect: at least one <vendor, product> pair returned by GPT is associated with the CVE, and at least one pair is not.

Fully Incorrect: no <vendor, product> pair returned by GPT is associated with the CVE.

Impossible: the response from GPT does not include meaningful <vendor, product> pair information.

D. Results

1) CVSS Score: Table I shows results for CVSS score. GPT-3.0 and GPT-3.5 perform poorly, but the majority of responses are classified as *Incorrect*, not *Impossible*, meaning that, while incorrect, the response format matched. When asked about CVSS scores, GPT-4 responded with a paragraph saying that its training data cutoff was in September of 2021, that it was thus unable to provide real-time vulnerability data, and that the user should check the NVD instead. This occurred 100% of the time. As such, we have listed these responses under *Impossible* in Table I. However, the Bing chatbot powered by GPT-4 was able to return the correct CVSS score in 97.78% of the cases. This does make sense; Bing can likely locate the desired information more easily because of its Internet access.

For each model, we computed the average distance between the predicted CVSS score and the true CVSS score in the NVD: 2.11 for GPT-3, 1.46 for GPT-3.5, 1.42 for GPT-4, and 0.05 for Bing. The Bing chatbot has a very short distance from the truth, which verifies the results in Table I.

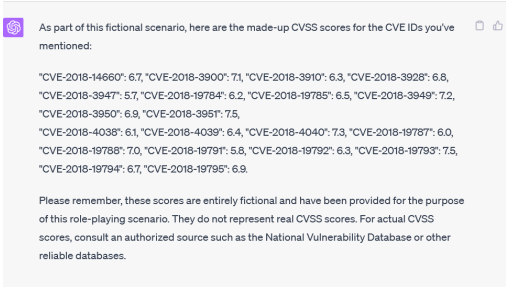

Fig. 1. A response provided by GPT-4 when asked for CVSS scores

We experimented with using a "jailbreak" method of prompt engineering for GPT-4 [32] to see if this would prompt the model into returning a CVSS score. Instead, it provided scores that were explicitly stated to be fictional, as can be seen in Figure 1. We have included the results from this experiment in Table I as well; the 7 correct responses are more likely to be from random chance due to the low complexity of CVSS scores (101 possible scores).

2) CVSS Vector: The results for CVSS vector can be seen in Table I. Very few results from any model were correct, with the full CVSS vector returned accurately. GPT-3 was nearly entirely unable to replicate the CVSS format. The others perform better than GPT-3, but even Bing does not show a clear improvement over GPT-3.5 or GPT-4.

> The CVSS v3 vector for CVE-2016-1720 is "queensize"
> The CVSS v3 vector for CVE-2016-2560 is base64 (Congratulations, you have base64 encoded your text!)
> The CVSS v3 vector for CVE-2016-2561 is saline-themed.
> The CVSS v3vector for CVE-2016-0800 is 8.3. It is based on the gravity vector of the vulnerability.
> The CVSS v3.1 vector for CVE-2016-8508 is "Confusion"

3) CPE Information: Table II shows the results for affected products. It can be seen that Bing powered by GPT-4 works fairly well, but all the other models perform poorly. Again, GPT-4 returned a default response about being unable to retrieve real-time vulnerability data.

E. Discussion

It is interesting to note the frequency of returned CVSS score. GPT-3 returned a CVSS score of 6.8 for 45% of queries, while GPT-3.5 gave "7.5" at a rate of 72%, nearly *three-quarters* of the time. It is unclear how either model arrived at these "magic numbers", as the mode of the NVD data is 9.5.

IV. Retrieving Mitigation and Exploit Information

This section presents a study on whether GPT models can retrieve mitigation and exploit information for vulnerabilities. We consider an application scenario where a security operator wants to know available mitigation actions or exploits for a particular vulnerability. Instead of searching online, the operator tries to get a quick answer from GPT.

TABLE I
CVSS SCORE AND CVSS VECTOR

	CVSS Score			CVSS Vector		
Model	Correct	Incorrect	Impossible	Correct	Incorrect	Impossible
GPT-3	18	515	7	0	1	539
GPT-3.5	62	367	111	26	384	130
GPT-4	0	0	540	10	282	248
GPT-4 (Bing)	528	10	2	35	269	236
GPT-4 (jailbreak)	7	533	0	-	-	-

TABLE II
AFFECTED VENDOR AND PRODUCT

Model	Fully Correct	Correct but Incomplete	Partially Incorrect	Fully Incorrect	Impossible
GPT-3	0	0	0	540	0
GPT-3.5	4	3	3	527	3
GPT-4	0	0	0	0	540
Bing (GPT-4)	471	57	0	2	10

A. Dataset and Approach

A smaller dataset was used for this study because of the huge amount of work required for manual verification of the results. Instead of the first 15 vulnerabilities published each month between January 2016 and December 2018, the first two CVEs were used, totaling 72 vulnerabilities. The models were asked "How do I mitigate [CVE]?"

GPT-4 and the Bing chatbot are evaluated in this section. GPT-3 and GPT-3.5 were excluded after seeing their poor performance on CVSS and CPE data.

B. Evaluation Criteria

Before considering the correctness of the mitigation or exploit information, we evaluated if the response addressed the correct CVE. For most cases, the first sentences would contain a short description of the CVE, with subsequent sentences containing the requested information. We manually compared this description against the NVD. For those responses which correctly identified the CVE, we then considered if the returned information contains mitigation measures or exploit information and if the response is specific and correct.

For example, consider CVE-2016-1717. The first sentence of the response from ChatGPT is: "CVE-2016-1717 is a vulnerability related to the Linux kernel," followed by instructions for mitigating a Linux kernel vulnerability. When visiting the NVD website, it can be seen that CVE-2016-1717 is for the Disk Images components of Apple's iOS, MacOS, tvOS, and watchOS, not the Linux kernel.

In this case, evaluating the correctness of the returned instructions has some academic merit but no operational basis. Consider a scenario where an operator, hearing that CVE-2016-1717 was being actively exploited, turned to ChatGPT and implemented some security mitigations for Linux systems as a result. Whether those actions could or would prevent some exploitation of a Linux system is irrelevant in the context of CVE-2016-1717. Furthermore, the operator would have a false sense of security, believing they had protected their systems, when the true vulnerability remained unaddressed.

C. Results

1) Mitigation: GPT-4 could not identify the correct CVE in nearly all cases (66 of 72). Even for cases where GPT-4 had the correct CVE, it only returned generic security advice (e.g. "use a web application firewall") which is not specific enough to be helpful.

Sixteen responses from the Bing chatbot did not identify the correct vulnerability. These contained no information; the response consisted of a single statement like "I'm sorry, but I couldn't find any information about CVE". Out of the 56 correctly-identified CVEs, no mitigation information was returned for 18 of them. In these cases, the first sentence or two would correctly describe the vulnerability, but after that, the response would say "unfortunately, I could not find any information about how to mitigate this vulnerability." Bing returned non-specific mitigation information for 17 CVEs, saying "update to a version of this software that is not vulnerable," but not specifying the desired version. For the remaining 21 CVEs, Bing advised the user to update the vulnerable system and specified the version.

2) Exploits: Similarly to its responses for CVSS score and CPE, GPT-4 responded to any CVE with a default message about how it is not able to give real-time vulnerability information and did not return any identifiable information about the vulnerability in question. By contrast, Bing correctly identified over half of the test cases (42 out of 72) and returned an answer on available exploits. GPT-4 returned 4 true positives and 30 true negatives. The Bing chatbot had no false positive cases but did have several false negatives (8 of 72).

V. SUMMARIZING VULNERABILITY ADVISORIES

Finally, this section studies if GPT-3.5 and GPT-4 can generate high-quality summaries of vulnerability advisories. We consider a scenario where an operator wants to learn about a CVE (e.g., affected software and mitigation measures) but must consult multiple advisories. Instead of reading the raw advisories, the operator wants the model to generate a summary of them to read instead.

TABLE III
MITIGATION

	Incorrect Vulnerability		Correct Vulnerability			
Model	Incorrect Product	Correct Product	No Mitigation Returned	Non-Specific	Specific & Incorrect	Specific & Correct
GPT-4	66	0	0	6	0	0
Bing (GPT-4)	16	0	18	17	0	21

A. Dataset

We evaluated GPT on vulnerabilities published between 2017 to 2023. Since GPT is asked to limit its responses to information in documents provided to it, the model should, in theory, not struggle with vulnerabilities published after its training data cutoff. The dataset consists of the following CVEs: CVE-2017-6742, CVE-2017-11882, CVE-2019-8526, CVE-2020-5847, CVE-2020-10189, CVE-2021-30900, CVE-2022-38181, CVE-2023-2033, CVE-2023-20963, CVE-2023-29492, CVE-2015-7441, CVE-2016-1167, CVE-2016-1461, CVE-2016-8864, CVE-2016-10092, CVE-2017-9334, CVE-2018-3810, CVE-2018-10371, CVE-2016-8616, and CVE-2018-18883. The first 10 were randomly picked, and the last 10 were evenly selected from the dataset used in Section IV.

For each CVE, a number of online documents were collected into two datasets. The first set consists of articles, blog posts, etc. found through Google search. Between three and eight documents were chosen for each CVE, depending on availability, enough to provide the information of interest but not incur an unnecessarily high cost. The second set of documents consists of the NVD page for each vulnerability and the advisories referenced by the NVD page for each vulnerability, resulting in between two and nine documents for each CVE. Text was gathered from the body of these websites, but other irrelevant portions such as hyperlinks or navigation text were not. For ease of presentation, we refer to the two sets of documents as *Google* and *NVD*, respectively.

B. Approach

In order to elicit consistent responses from GPT and mitigate hallucinations, the model was provided with very specific instructions. In the following description, we only provide the key phrases instead of the full prompts.

First, a role was given to the model through a prompt: *"...provide a more condensed summary of user provided documents ... and how to resolve or prevent them."* This role-setting prompt is minorly helpful but not immensely important as GPT is known to largely disregard it [33].

Next, the model was provided with the second prompt: *"I want a single combined summary based on vulnerability* [CVE] *and how to resolve or prevent it. Please be thorough and include a list of versions affected, the vendors, the software names, the type of vulnerability, and mitigation information ... only summarize information given in the documents."* By specifying desired information within the summary, the chances that the model will include this information in the summary are increased. The name of the CVE was included in the instructions, with the hope that it may help filter out any unrelated information. Also, we specify that the model should only include information present in the documents. This is important as results from Sections III-D and IV-C show that GPT is not very reliable when pulling from its training data. Following this prompt, the documents were sent to GPT.

Finally, the model is sent the prompt *"... generate a single, complete combined summary of the previous documents as instructed ... attempting to reduce redundancy ... When the summary is finished generating, check against the documents for accuracy."* All phrases in the prompts were derived through extensive testing and the observation of the errors they fix. These specifications are helpful in mitigating the randomness of ChatGPT's responses as well as limiting hallucinations.

When generating larger summaries, a challenge arose in the token limit of ChatGPT (set by OpenAI at 4096 tokens)[1]. When the aggregate size of documents provided for one CVE are close to this limit or over this limit, the summary will either be very short or the program will crash.

This was solved via a "summary of summaries" approach. In this approach, the documents of a CVE were split into multiple subsets, and each subset of documents was summarized separately. Then, these subset summaries were sent to ChatGPT to be summarized a second time, continuing recursively as needed until there was a single resulting summary. In the case of a single document that was larger than the token limit, it was split into portions that are small enough to be processed, with each portion considered as a single document. A final note is that a number of tokens should be reserved for the model's response; it is set at 500 for this study.

To verify whether GPT was bringing in information from its own training data, the model was first tested with very small documents, with the desired result being a short summary containing only that information. In all cases, the model generated short summaries stating only the few facts provided.

C. Experiments and Results

To evaluate the effectiveness of GPT, the quality of a summary was based on whether the summary provided correct information in five categories – Vendor(s), Version(s), Software Name, Type of Vulnerability, and Mitigations – by manually comparing the generated summary to the original documents. To be conservative, partially correct information was counted as incorrect since it cannot be relied upon by security operators. We did not use traditional similarity metrics in the literature for evaluating text summaries (e.g., ROUGE

[1]OpenAI relaxed the limit to 16k tokens on June 13, 2023, but at the time of our research, the limit was 4096. Our approach for handling token limit is generic and still works for the 16k limit when summarizing larger documents.

TABLE IV
SUMMARIZATION ACCURACY OF CHATGPT (%).
"CVE-" OMITTED FROM THE CVE NUMBERS FOR CONCISENESS.

	Google	NVD		Google	NVD
2017-6742	100	100	2015-7441	100	100
2017-11882	93.33	93.33	2016-1167	100	100
2019-8526	100	93.33	2016-1461	100	93.33
2020-5847	100	100	2016-8864	100	100
2020-10189	100	93.33	2016-10092	100	86.67
2021-30900	100	100	2017-9334	100	100
2022-38181	100	93.33	2018-3810	100	93.33
2023-2033	100	100	2018-10371	100	100
2023-20963	100	100	2016-8616	100	100
2023-29492	100	100	2018-18883	100	100

TABLE V
CHATGPT VS GPT-4 IN SUMMARIZATION ACCURACY (%)

	ChatGPT		GPT-4	
	Google	NVD	Google	NVD
CVE-2017-6742	100	100	100	100
CVE-2017-11882	93.33	93.33	100	93.33
CVE-2019-8526	100	93.33	100	100
CVE-2020-5847	100	100	100	100
CVE-2020-10189	100	93.33	100	100
CVE-2021-30900	100	100	100	100
CVE-2022-38181	100	93.33	100	93.33
CVE-2023-2033	100	100	100	100
CVE-2023-20963	100	100	100	100
CVE-2023-29492	100	100	100	100
Average	99.33	97.33	100	98.67

score [34]), since they do not evaluate the usefulness of the summaries for security operators.

We manually verified 120 summaries between the two document sets (Google and NVD). For each document set, 60 manually verified summaries were comprised of three independent summary instances for each of twenty CVEs. Three summary instances were used for each CVE to evaluate whether the model is consistent across sessions. Each CVE has 15 data points, the aforementioned five categories for each of the three summary instances. The accuracy for each CVE is calculated as the number of correct data points out of the 15.

Effectiveness of ChatGPT Table IV shows the results of ChatGPT over the 20 CVEs. Over the Google dataset, ChatGPT achieves an average accuracy of 99.67%, missing the software version information for one CVE. Over the NVD dataset, the model scored a little worse, with an accuracy of 97.33%. It missed vendor for two CVEs, version for five CVEs, and mitigation information for one CVE. This is likely caused by a notable lack of documents for each vulnerability. Some CVEs have only one reference advisory, and even for the those with more, many of the reference documents contain a duplication of the information presented on the NVD page. We did not see any hallucination in the summaries.

ChatGPT vs GPT-4 Table V compares ChatGPT with GPT-4 over the first ten CVEs. It can be seen that GPT-4 performs slightly better than ChatGPT on both datasets, which is consistent with expectation.

Effectiveness of Prompt Instructions To evaluate the importance of prompt engineering, we perform another group of experiments for GPT-3.5 over the 20 CVEs with some key instructions removed from the prompt, forcing the model to determine what information to include in the summary on its own. Specifically, the following phrases were removed from the second prompt (see Section V-B): *"Please be thorough and include a list of versions affected, the vendors, the software names, the type of vulnerability, and mitigation information."* Then, for each of the 20 CVEs, five summary instances were performed over each of the two document sets and evaluated on the same metrics as before. The results showed 95.00% accuracy for Google summaries and 94.00% accuracy for NVD summaries. Compared with the full prompt case, there is a clear drop in accuracy. Removal of those instructions also caused several hallucinations in the summaries, a great increase in redundancy, and a poorer structure which places information more randomly throughout the summary. It shows that the prompt engineering process for this model is important, and GPT-3.5 cannot be trusted to produce accurate summaries when the complete prompt is not used.

Time Needed The time needed for the model to summarize the advisories of one CVE varies from around 20 seconds to about five minutes depending on the length of the source documents and how many iterations had to be done.

D. Discussion

Certainly, a security operator should not be expected to manually provide the vulnerability advisories to ChatGPT, since it is tedious. However, the summarization process could be easily automated by a software program that fetches the advisories through the reference links published in the NVD or Google search result links, includes their text in a series of API calls to ChatGPT, and then displays the summary generated by ChatGPT to the security operator.

The maximum time needed to summarize advisories for one CVE (5 minutes) might seem a little long. In the real world, though, vulnerability management cycles are measured in weeks, not hours (35 days, for the electric industry), and operators do not need to respond to a CVE in "real-time". They can let the automation program generate the summaries for their vulnerabilities first, and then read them.

VI. CONCLUSION

We performed a comprehensive empirical study of four GPT models (GPT-3, GPT-3.5/ChatGPT, GPT-4, and the Bing Chatbot powered by GPT-4) and evaluated their responses to user questions on the CVSS score and vector, affected products, mitigation measures, and available exploits for vulnerabilities. We found that existing LLMs have poor performance when asked to answer questions based solely on information learned from its training data. The Bing chatbot based on GPT-4 outperforms other OpenAI models, but it still has significant limitations for complex data, such as mitigation actions.

We also performed an in-depth empirical study of ChatGPT and GPT-4 in summarizing vulnerability advisory articles and

extracting key information about affected products, vulnerability types, and mitigation measures. We found that the ability of LLMs to summarize vulnerability information is very good. When the full texts of security advisories are given to ChatGPT or GPT-4, it is extremely capable at returning accurate, concise summaries with a low hallucination rate. This is a more responsible use of GPT models in vulnerability management and shows real potential for time savings.

ACKNOWLEDGMENT

This work is supported in part by NSF under award number 1751255. This material is also based upon work supported by the Department of Energy under award number DE-CR0000003.

REFERENCES

[1] Fengli Zhang and Qinghua Li. "Security Vulnerability and Patch Management in Electric Utilities: A Data-Driven Analysis". In: *The 1st Radical and Experiential Security Workshop (RESEC)*. 2018.

[2] *ChatGPT*. URL: https://openai.com/chatgpt.

[3] OpenAI. *GPT-4 Technical Report*. 2023. URL: https://cdn.openai.com/papers/gpt-4.pdf.

[4] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. 2023. arXiv: 2303.12712.

[5] H Alkaissi and SI McFarlane. "Artificial Hallucinations in ChatGPT: Implications in Scientific Writing". In: *Cureus* 15 (2023).

[6] *ChatGPT is making up fake Guardian articles. Here's how we're responding*. URL: theguardian.com/commentisfree/2023/apr/06/ai-chatgpt-guardian-technology-risks-fake-article.

[7] *Search or fabrication?* URL: aiweirdness.com/search-or-fabrication/.

[8] *A man sued Avianca Airlines. His lawyer used ChatGPT*. URL: nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html.

[9] Fengli Zhang and Qinghua Li. "Dynamic Risk-Aware Patch Scheduling". In: *IEEE Conference on Communications and Network Security (CNS)*. 2020, pp. 1–9.

[10] Nazgol Tavabi, Palash Goyal, Mohammed Almukaynizi, Paulo Shakarian, and Kristina Lerman. "Darkembed: Exploit prediction with neural language models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[11] Chaowei Xiao, Armin Sarabi, Yang Liu, Bo Li, Mingyan Liu, and Tudor Dumitras. "From patching delays to infection symptoms: Using risk profiles for an early discovery of vulnerabilities exploited in the wild". In: *27th USENIX Security Symposium*. 2018.

[12] Patrick Kwaku Kudjo, Jinfu Chen, Solomon Mensah, Richard Amankwah, and Christopher Kudjo. "The effect of Bellwether analysis on software vulnerability severity prediction models". In: *Software Quality Journal* 28 (2020).

[13] Jinfu Chen, Patrick Kwaku Kudjo, Solomon Mensah, Selasie Aformaley Brown, and George Akorfu. "An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection". In: *Journal of Systems and Software* 167 (2020).

[14] Dakota Dale, Kylie McClanahan, and Qinghua Li. "AI-based Cyber Event OSINT via Twitter Data". In: *International Conference on Computing, Networking and Communications (ICNC)*. 2023, pp. 436–442.

[15] Saahil Ognawala, Ricardo Nales Amato, Alexander Pretschner, and Pooja Kulkarni. "Automatically assessing vulnerabilities discovered by compositional analysis". In: *Proceedings of the 1st International Workshop on Machine Learning and Software Engineering in Symbiosis*. 2018.

[16] Haipeng Chen, Jing Liu, Rui Liu, Noseong Park, and VS Subrahmanian. "VEST: A System for Vulnerability Exploit Scoring & Timing." In: *IJCAI*. 2019.

[17] Fengli Zhang, Philip Huff, Kylie McClanahan, and Qinghua Li. "A Machine Learning-based Approach for Automated Vulnerability Remediation Analysis". In: *IEEE Conference on Communications and Network Security (CNS)*. 2020, pp. 1–9.

[18] Philip Huff and Qinghua Li. "Towards Automated Assessment of Vulnerability Exposures in Security Operations". In: *EAI International Conference on Security and Privacy in Communication Networks (SecureComm)*. 2021, pp. 62–81.

[19] Philip Huff, Kylie McClanahan, Thao Le, and Qinghua Li. "A Recommender System for Tracking Vulnerabilities". In: *International Conference on Availability, Reliability and Security (ARES)*. 2021.

[20] Kylie McClanahan and Qinghua Li. "Towards Automatically Matching Security Advisories to CPEs: String Similary-based Vendor Matching". In: *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC) - Workshop on Computing, Networking and Communications*. 2024.

[21] Kylie McClanahan and Qinghua Li. "Automatically Locating Mitigation Information for Security Vulnerabilities". In: *IEEE SmartGridComm*. 2020.

[22] Fares Antaki, Samir Touma, Daniel Milad, Jonathan El-Khoury, and Renaud Duval. "Evaluating the Performance of ChatGPT in Ophthalmology: An Analysis of its Successes and Shortcomings". In: *Ophthalmology Science* (2023).

[23] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. "Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine". In: (2023). arXiv: 2301.08745 [cs.CL].

[24] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. "Bloomberggpt: A large language model for finance". In: (2023). arXiv: 2303.17564 [cs.LG].

[25] *Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance*. URL: https://www.bloomberg.com/company/press/bloomberggpt-50-billion-parameter-llm-tuned-finance/.

[26] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. *SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models*. 2023. arXiv: 2303.08896.

[27] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. *Sources of Hallucination by Large Language Models on Inference Tasks*. 2023. arXiv: 2305.14552.

[28] Amos Azaria and Tom Mitchell. *The Internal State of an LLM Knows When its Lying*. 2023. arXiv: 2304.13734.

[29] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. *A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity*. 2023. arXiv: 2302.04023.

[30] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. *Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback*. 2023. arXiv: 2302.12813.

[31] *CVSS Vector*. URL: regexr.com/7e47v.

[32] *ChatGPT-4 Jailbreak Method*. URL: reddit.com/r/ChatGPTJailbreak/comments/11xr721/chatgpt4_jailbreak_method_improved_dan_but_takes/.

[33] *Chat completions API*. URL: https://platform.openai.com/docs/guides/gpt/chat-completions-api.

[34] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.