

C04: Reportinator (did not hack)

Saturday, December 23, 2023



6:27 AM

Difficulty: Level 2

Noel Boetie used ChatNPT to write a pentest report. Go to Christmas Island and help him clean it up.

CONVERSATION w/ Elf Noel Boetie	<p>Noel Boetie (Rudolph's Rest Resort)</p> <p>Hey there, Noel Boetie speaking! I recently tried using ChatNPT to generate my penetration testing report. It's a pretty nifty tool, but there are a few issues in the output that I've noticed. I need some guidance in finding any errors in the way it generated the content, especially those odd hallucinations in the LLM output.</p> <p>I know it's not perfect, but I'd really appreciate the extra eyes on this one. Some of the issues might be subtle, so don't be afraid to dig deep and ask for further clarification if you're unsure. I've heard that you folks are experts about LLM outputs and their common issues, so I trust you can help me with this. Your input will be invaluable to me, so please feel free to share any insights or findings you may have. I'm looking forward to working with you all and improving the quality of the ChatNPT-generated penetration testing report. Thanks in advance for your help! I truly appreciate it! Let's make this report the best it can be!</p> <p>** ----- Response after completing challenge ----- **</p> <p>Great job on completing that challenge! Ever thought about how your newfound skills might come into play later on? Keep that mind sharp, and remember, today's victories are tomorrow's strategies!</p> <p>From https://2023.holidayhackchallenge.com/badge?section=conversation&id=noelboetie</p>
HINTS	<p>Reportinator</p> <p>From: Noel Boetie Terminal: Reportinator</p> <p>I know AI sometimes can get specifics wrong unless the prompts are well written. Maybe chatNPT made some mistakes here.</p> <p>From https://2023.holidayhackchallenge.com/badge?section=hint&id=hintReportinator1</p>

ChatNPT created a pentest report. However, the elves need some manual intervention to check the findings. You have to mark which findings are legit and which ones are hallucinations. You will get a message when you've selected the correct hallucinations - both in the iFrame and as a completion message. See the Pentest Report [here](#).

	This icon represents a legitimate finding. Click to toggle to a hallucination.
	This icon represents a hallucinated or false finding. Click to toggle to a legitimate finding.

MY WORK AND ANSWER










I thought this would be easy enough and a quick challenge to solve given my experience in this space. So in my initial approach, I read each reported item to evaluate its validity and used Microsoft's Bing AI to help ferret out hallucinations. Despite asking it specifics from the report, the AI proved to add to the complexity of solving this challenge (this convolution was a clear illustration of the pain point of AI, which I will elaborate later) .

I then created an Excel file where I used Bing AI to generate a listing of all possible combinations of a binary 9-digit where exactly two of the digits were zeros (0's). After spending a significant amount of time working this angle, I later learned there were three, not two hallucinations in the vulnerability report. So back to Bing AI to now generate a listing of all possible combos where exactly three of the digits are 0s. At this stage, I'm thoroughly confused about what the 0s and the 1s mean - yep, I spent that much time on this challenge that what I was counting as 0s became 1s and vice versa.

And Bing AI continued to produce lists (in both instances) of 9 digits containing no zeros, 1, 2, 3 or more zeros. I cannot tell you how many ways I tried ask the questions to nudge it to produce useable data or how frequently it apologized for the mistakes. So much in fact, that I entertained the idea of bypassing the learning and going the brute force route.

Eventually, I was able to verify that six of the findings are legitimate and three are hallucinated or false and correctly reset on my binary representation: 0 denoting legitimate findings an 1 denoting hallucinated or false findings. No longer feeling like I was spinning my wheels, I continued with the manual review of the findings. I wasn't particularly interested in brute forcing the answer because I wanted to learn more about these vulnerabilities, especially knowing that knowledge would be useful in upcoming challenges.

Heeding hints from other players, especially on how to look at the report (not as a real pen test but as a report from an LLM), I was able to more easily reason which were likely hallucinated by the AI, as noted in my table below.

Vulnerability Report	Invalid	Reasoning for identifying the item as a false-positive
1. Vulnerable Active Directory Certificate Service-Certificate Template Allows Group/User Privilege Escalation	VALID 	No functional errors found. The find command in Certipy can be used to enumerate Active Directory Certificate Services (AD CS) certificate templates, certificate authorities, and other configurations. However - and probably a nit - there is not a <u>specific</u> option in the find command that directly identifies certificate templates that allow users to supply their own Subject Alternative Name (SAN) and determine if a Client Authentication Extended Key Usage (EKU) is set1.
2. SQL Injection Vulnerability in Java Application	VALID 	No functional errors found. This may be a nit: The sqlmap tool is an automated (not manual) pentest tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. Also, Jeff Forristal, also known as Rain Forrest Puppy, is credited with discovering SQL injection vulnerabilities, which he identified in 1998. It is unknown what tools he used. The scanning tools Nessus and OWASP Zed Attack Proxy (ZAP) were founded in 2010 and 201x, respectively.
3. Remote Code Execution via Java Deserialization of Stored Database Objects	FALSE 	The report mentions an invalid TCP port number (88555). This may be a nit: The ysoserial tool is used to generate payloads that exploit Java deserialization vulnerabilities. It's not used to evaluate an application for vulnerabilities, but rather to demonstrate the impact of a known vulnerability. This may be a nit: Vulnerabilities are classified as CVEs, not CWEs (while the CWE-502: Deserialization of Untrusted Data is real). <ul style="list-style-type: none"> • CVE is a standard for identifying and naming specific vulnerabilities, CWE is a standard for classifying and describing the types of weaknesses that can lead to vulnerabilities1234. • Basically, the CVE treats symptoms (specific vulnerabilities), while CWE treats causes (types of weaknesses). Note: The vulnerability within an externally-accessible Java app can occur on an internal IP address (as identified in the write-up)
4. Azure Function Application-SSH Configuration Key Signing Vulnerable to Principal Manipulation	VALID 	No functional errors found NOTE: The AI noted it was not aware of such a vulnerability with the Azure SSH configuration in the Azure Function Application responsible for SSH key signing. Also, SSH if used within Azure Functions, is only visible/enabled with the "Azure Function app premium" and "App service hosting plan" of the Azure Function app12. Connecting with SSH is not supported in the consumption plan12. This may be a nit: The parameter is called ServicePrincipal, not service principal. If sign-principal is being used in your application, it's likely a custom parameter defined by the developers. This may be a nit: This finding is not classified as a Broken Authentication vulnerability, according to the OWASP Top 10 Application Security Risks
5. Azure Key Vault-Overly Permissive Access from Azure Virtual Machine Metadata Service/Managed Identity	VALID 	No functional errors found. NOTE: The command to list deleted keys requires the subscription id (i.e., az keyvault list-deleted --subscription {SUBSCRIPTION ID} --resource-type vault) This may be a nit: The error message should not be so verbose. It can be used in recon.
6. Stored Cross-Site Scripting Vulnerabilities	FALSE 	"HTTP SEND" is a non-standard term used by the reporting team. This may be a nit: Input data needs to be sanitized, not encoded (i.e., improper terminology)
7. Browsable Directory Structure	VALID 	No functional errors found.
8. Deprecated Version of PHP Scripting Language	VALID 	No functional errors found. This may be a nit: Nmap does not directly enumerate the PHP version of a host.
9. Internal IP Address Disclosure	FALSE 	There is no HTTP 7.4.33 version. Also, the curl command-line tool is used for transferring data using various protocols, including HTTP, HTTPS, FTP, and more and is often used to interact with web servers and APIs. There are certain scenarios where you might be able to use curl to interact with a service that could potentially reveal an internal IP address. For example, if a service is configured to respond with its internal IP address, then a curl request to that service could reveal the internal IP2. However, this would be a function of the service, not curl itself. Curl does not have the capability to expose the internal IP address of a target IP address.

☐ <attach XLS here>

F1	F2	F3	F4	F5	F6	F7	F8	F9	Results & Comments
									<-- CORRECT ANSWER
0	0	1	0	0	1	0	0	1	This is the correct representation of the real vs hallucinated findings
1	1	0	1	1	0	1	1	0	

Report Validation Complete

Great work! You've successfully navigated through the intricate maze of data, distinguishing the authentic findings from the AI hallucinations. Your diligence in validating the penetration test report is commendable.

Your contributions to ensuring the accuracy and integrity of our cybersecurity efforts are invaluable. The shadows of uncertainty have been dispelled, leaving clarity and truth in their wake. The findings you have authenticated will play a crucial role in fortifying our digital defenses.

We appreciate your expertise and keen analytical skills in this crucial task. You are a true asset to the team. Keep up the excellent work!

Comments of Interest from other Players

Test cases in this challenge may provide hints for later challenges, such as or [Missile Diversion](#).

Step back and simplify it in your mind. What is the challenge asking you to do. How would you explain it to a 5 year old? Once you do that, it should be a lot clearer. Also, if the elves used an AI to create it, AI can help you digest, understand, interpret, dig through each finding. Lastly, there are hints in the challenge itself and in the Discord chat. NOTE: Misspellings and punctuation errors are not hallucinations or errors to be considered when evaluating the findings for their accuracy.

Read through and verify accuracy. Look at the recommendations and screenshots and see if they make sense in the context of the vulnerability reported.

- I saw some 'external' findings that looked like they ran locally which tipped me off on a few of them
- Use a brute force technique of inputting different iterations and monitoring what the http responses are for each combo to work out what is correct.

If you are reading it as a real pen test report you will probably pick up on things that are less relevant to this challenge. Look at it as if it's a report from an LLM and you are looking for hallucinations! You have been told that it's from an LLM so you should probably accept that fact and accept that it has NOT actually gone off any done ANY of the things in the report (just as ChatGPT won't go exploit for you!). So your job is less to validate if the findings are valid, but to find the ones where there are errors so glaring that you know they *could not* be valid.

Ask AI to be as critical as possible about technical accuracy; follow up w/ specific questions about each technical piece will give you more to work with.

- Instead of saying: how can i write a powershell script that brute forces the answer to this website <https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/>, say: just tell it you want to automate testing the correct combination of button toggles until it gets one right, then there's nothing bad there

You don't need LLMs (Large Language Models) to spot the hallucinations - which are kinda funny - but we encourage your use of them.

- Example: <https://www.makeuseof.com/what-is-ai-hallucination-and-how-do-you-spot-it/#:~:text=Here%20are%20some%20ways%20to%20spot%20AI%20hallucinations,2.%20Computer%20Vision%20...%203%203.%20Self-Driving%20Cars>
- Find an AI to check for hallucinations for ChatGPT Bing Bard
 - most like <https://contentdetector.ai/> tell me the chances of that suggestion is 25% or under
 - Someone asked Copilot "What are the factual errors in the report on this page?" It replied: "There are no factual errors in the report on this page, as it is a fictional story set in a virtual world called Film Noir Island. The page is part of a game called Gumshoe Alley, where the user plays as a rookie detective who works with Tangle Coalbox of Kusto Detective Agency to solve a murder mystery. The page introduces the character of Tangle Coalbox and the setting of the game."
- Running the report through different LLMs to spot any factual errors --> . Kinda funny about what different ones (bard, bing, chatgpt) will tell you is and isn't factual.
 - ChatGPT hallucination is reportedly more severe and common than Bard AI hallucination, as ChatGPT is trained on a larger and more diverse dataset than Bard3.
 - ChatGPT also has a more open-ended and creative style than Bard, which makes it more prone to generating irrelevant or nonsensical answers3. However, there are ways to reduce these hallucinations.
 - <https://www.bloomberg.com/news/newsletters/2023-04-03/chatgpt-bing-and-bard-don-t-hallucinate-they-fabricate>
 - <https://g.co/bard/share/04969aa221eb>

For those struggling on this and wanting to avoid brute force I think the best advice I can give is this, I don't think it needs spoiler tags but tell me if you disagree...

- Don't focus too much on attack details.

- Use the text as your primary source.
- Parse one sentence at a time and look for things that are glaringly wrong/impossible/non-sensical.
- You can solve this with basically no knowledge of the attacks, relatively fundamental networking knowledge is all you need
- It might help to put them in order of the 'wrongest' and submit your set of answers from the top of that list

Finally, despite me saying you don't need to know about the attacks for this challenge, there sure is a lot of interesting insight into how to exploit stuff, could come in useful in the distant future maybe.

Issues found

[SPOILER] Note for Challenge Author: | | EDIT: This has now been resolved, for anyone reading -- The CVE's listed on Finding number 8 explicitly do NOT apply to PHP version 7.4.33. This threw me for quite the loop, considering the correct answer for this finding.

The recommendation of 9 is fishy : SCS recommends that NPS modify the Location header to reflect the host Windows registration key rather than the internal IP address of the host.

Tools to Brute-Force

There's technically a way to figure it out without reading through the report: brute forcing. There are 9 T/F questions, so 512 possible permutations ($2^9 = 512$ possibilities). You can brute force the 512 combinations of answers to get the right one. Count in binary (1 means hallucination, 0 means fact).

It was shared that there are only 2 reported vulnerabilities that are valid. Someone else shared the count is 3. Confirmed there are 3 which means there are 6 false-positives in the report. Given 3 ($r=3$), there are exactly ...

UPDATE: 6 are correct findings, 3 are false-positive

Burp Intruder / Burp Suite to brute force it

- When given different status codes EX: Not 400, but I had only 2 give me 500 Yet when inputting those answers, they were wrong. Am I not doing something right here?
- Trying to use | | burp, got the response captured in repeater just cant figure out to how to automate with legit-lg.png and hallucinate.png anyone free for a dm?

take ref of burp bruteforce

I don't know if this is really a hint on how to brute force it but I found <https://curlconverter.com/> helpful

- used curl and bash

Anyone try to brute force it w/ Windows? I "copy as cURL"ed the check request to get the parameters right and brute-forced every combo w/ a shell script | | I get all failures

I did it via Zap to validate the brute force way, then basically read each item and tried to work out if I thought it was valid or not just for completeness. I would suggest others at least give it one pass 'properly' before brute forcing, I feel like I half cheated myself out of some learnings!

I'm trying to get around it as well. I found the javascript file that contains the conditional "if wrong then return error, else success" and I changed that statement to always take any answer to be true. The problem is I don't know how to use that changed code in the environment.

When I noticed the most obvious networking error ever I laughed.

+1 for the fuzzing script route for this one.

Np, I found bash to be the most accessible, with python solely for the parameter value replacement bit. Either or works, took me maybe 1hr total

```
import itertools
def test_combination(combination):
    # This is a placeholder for the function that tests the combination. # You should replace this with your actual testing code.
    correct_combination = [1, 0, 1, 0, 1, 0, 1, 0, 1]
    return combination == correct_combination
# Generate all possible combinations of 9 buttons being either on or off.
for combination in itertools.product([0, 1], repeat=9):
    if test_combination(combination):
        print(f"Found the correct combination: {combination}")
        break
```

In this script, `itertools.product([0, 1], repeat=9)` generates all possible combinations of 9 buttons being either on (1) or off (0). The `test_combination` function is a placeholder for the function that tests the combination. You should replace this with your actual testing code.

From <https://www.bing.com/search?q=Certipy+command+to+enumerate+and+attack+Active+Directory+Certificate+Services+%28AD+CS%29&q=d&form=CONVAJ&showconv=1>>