ATLAS.ti Report

CodeMap-formative-study-new

Codes

○ auditing efficiencies differences within the same company

1 Quotations:

6:3 ¶ 6 – 11 in P6.docx

Codes:

○ auditing efficiencies differences within the same company

Content:

> Interviewer: Right, the efficiency has improved a lot.
> P6: Right, this is during the audit period, and it is relatively fair. It won't be the case that, for example, in a company with 5 auditors, the time given by each auditor may differ by two or three days or three or four days. No, because we can help you, for example.
> Interviewer: So it means that some auditors work quickly while others work slowly, right?
> P6: There are still issues with the given time gap.
> Interviewer: But if we talk about fairness, is it more fair for the company or for each auditor?
> P6: It is relatively fair for clients because in many cases, audit fees depend on the difficulty of your project, and what does the project difficulty determine? The length of time we conduct the audit determines the amount of money received. In the early stage, during the audit, for example, when a project party engages an audit firm for an audit, the first step is to conduct an audit assessment, and ChatGPT provides us with some assistance.

○ Challenge

28 Quotations:

1:21 ¶ 44 – 45 in P1.docx

Codes:

○ Challenge: GPT challenge: reasoning: context is limited (due to the data cannot be imported totally), especially for reasoning and validating

Content:

> Interviewer: Right, I understand, so it's equivalent to saying that ChatGPT can participate in all three stages of your planning, reasoning, and validating. When using this conversation-based approach to execute your process, what do you think is the biggest challenge you've encountered?
> P1: There are quite a few. The first issue is that ChatGPT has insufficient context, resulting in its scope being incomplete, while humans can consider things from a global perspective. For example, if using a checklist, it often produces some false positives. The cause of these false positives is actually because I am querying and detecting at the function level, so sometimes it will produce some false positives. That is, because there are already checks, corresponding judgments, and corresponding restrictions elsewhere, it still thinks there are vulnerabilities, which is definitely a false positive. In fact, it is still a context issue.

1:22 ¶ 50 – 51 in P1.docx

Codes:

○ Challenge: GPT challenge: planning: lack of customized scanning

Content:

> Interviewer: Is there any other challenge in the second stage?
> P1: Specifically, I don't remember very clearly either. There might be redundant processes in knowledge. During the planning phase, knowledge might have some redundant processes.

> Specifically, there are some protected functions that ChatGPT also scanned, which is actually the so-called "only owner" part mentioned earlier. That is, ChatGPT also scanned some functions that didn't need to be scanned, but I've resolved this issue. Some redundant functions that didn't need to be scanned were also scanned by me. This is one aspect. I didn't give it a long enough chain of thought. I simply told it to scan all functions, and it didn't think carefully about which ones to scan and which ones not to. In fact, even when written in great detail, humans may not be able to quickly understand and make judgments.

1:23 ¶ 52 – 53 in P1.docx

Codes:

○ Challenge: GPT challenge: reasoning: GPT cannot do self-learning

Content:

> Interviewer: Even if ChatGPT returns a relatively detailed description, people may not be able to tell that it is wrong.
> P1: Because it's too complex, some vulnerabilities are overly complicated, and ChatGPT's descriptions aren't particularly good either. So sometimes I'll add a sentence, for example, I'll say "Please explain it to me line by line according to the specific code." I usually add this sentence because I simply can't understand those things; they're too obscure. Then a problem arises during the reasoning process: the ability to draw inferences from one instance is not strong enough. I might only ask about the key concept of a certain functionality project and not ask about anything else, and then I assume that similar vulnerabilities won't occur.

1:24 ¶ 54 in P1.docx

Codes:

○ Challenge: GPT hallucinations

Content:

> Another example is that there are some so - called key concepts that simply should not be used in this scenario. At this point, if you force a search, ChatGPT may exhibit some hallucinations, resulting in false positives. This is because ChatGPT sometimes tries to answer your questions forcefully and will start to go off on a tangent.

1:25 ¶ 55 in P1.docx

Codes:

○ Challenge: GPT can only step by step but human can be jumpy

Content:

> Another thing is that human thinking is relatively jumpy, with meaningful leaps. So in machines, it's like this. For example, many so-called math-oriented knowledge related to mathematics, but without very special logical errors, is evaluated, so basically it's all in the realm of uncertainty. Then I need to continue to improve the vulnerability library, increase knowledge, and expand knowledge. The more general knowledge there is, the more effective it will be after coordinating with planning, making it more capable of thinking leaps like humans. To put it simply, for example, if I match 10 pieces of knowledge, ChatGPT will select the first 5, and the last 5 may be missed. How can I make ChatGPT also scan out the last 5? To put it simply, I remove the first 5, extract them from the ranking, and move the last 5 forward. This is the general idea of the solution. I don't know if you can understand what I'm saying. If the first 5 are all of one category, I may group them together or take a closer look at them. I try to make the so-called key concepts it matches as diverse as possible, which is my ultimate goal. The better its diversity, the closer it can get to the result of human jumpy thinking.

1:26 ¶ 56 – 59 in P1.docx

Codes:

○ Challenge: GPT challenge: validating: GPT write reports/attack script due to the lack of context

Content:

Interviewer: So your goal is to optimize ChatGPT, improve your usage methods, reduce false alarm rates, and increase efficiency.
P1: Right, reducing false positive rates and increasing effectiveness are my current goals. Additionally, humans can actually understand most of the results returned by ChatGPT, but this is after all manual work, not automated. I think automated validating might still be a matter of context, because when humans write so-called attack scripts, they actually comprehensively consider the front and back limitations or code of the entire project, but programs actually find it difficult to consider such comprehensive code or some limitations. Because an attack script itself can be understood as a sub-function, but this sub-function calls almost all other functions of the project, which imposes a requirement that you need to have an understanding of all other functions, and humans can achieve this.
Interviewer: So basically ChatGPT can't write this attack script now.
P1: Let me put it this way. It has two indicators. One is grammatical correctness, and in terms of grammatical correctness, it's okay, but in terms of logical comprehensiveness, it's not okay. Right, that's about it.

1:27 ¶ 60 – 65 in P1.docx

Codes:
○ Challenge: GPT challenge: interaction: interaction: everytime human needs to interact with GPT from the start and copy the prompt

Content:
Interviewer: The question just asked used the term "challenge". Could you demonstrate how you interact with ChatGPT?
P1: One of the things I did before was the first solution, the manual solution, which involves inputting this thing and then directly copying the long prompt I just mentioned to it, because I've been adjusting it for a long time. Then at this point, ChatGPT will give an answer that looks very reasonable, seemingly very much like it, but sometimes it's not.
Interviewer: So it's equivalent to starting a conversation from scratch every time.
P1: It can be understood in this way. Each function has a possible occurrence of a new vulnerability each time.
Interviewer: So there are many repetitive steps in it.
P1: Right, to put it bluntly, this prompt, look at where the prompt I just found came from, I dug it out from the chat history. In fact, I don't have a good place to store this prompt. Ok, look at this part, it can give a result that looks very reasonable, and it does it very reliably, but there may be two situations. First, it's too complex, what it writes is quite complex. So at this time, it will be asked to explain this vulnerability in detail, step by step with reference to the code, and basically finish the output, and then ask it to output this.

2:4 ¶ 12 – 19 in P2.docx

Codes:
○ Challenge: GPT challenge: interaction: lack of enough context/cannot input too much code ○ Planning: ChatGPT for understanding: interaction

Content:
Interviewer: So different variables will have different associated business logic.
P2: Such a situation may exist, but the prerequisite is that you must first use your own experience or identify that it has two sets of logic.
Interviewer: So it's like when you're understanding this code, you first use your own judgment, then form a general internal understanding of this code, and then ask about the relatively minor points.
P2: This way is more efficient. If I can figure it out in a short time, it will be even more efficient. If not, I can also turn to ChatGPT and start asking from scratch.
Interviewer: Okay, in most cases, can you tell on your own or not?
P2: In most cases, people are just too lazy to read themselves and hand it over to GPT first.
Interviewer: But ChatGPT doesn't have the ability to digest the code of such a large project; instead, it can only handle a small snippet of code.
P2: Essentially, we identify them one by one as contract files; a project would be too large, so there's no other way.

2:7 ¶ 36 – 37 in P2.docx

Codes:

○ Challenge: GPT challenge: reasoning: hard to input enough knowledge

Content:

> Interviewer: This note refers to yours.
> P2: A case analysis of Hundred finance was attacked, and that attack was caused by the loss of computational precision, which is equivalent to a case. I feel that there are probably many pictures inside that cannot be uploaded, so if it can only view text, it cannot understand, because there are some screenshots that were directly pasted when doing the analysis, not in text form, and it can only upload text when uploading.

2:9 ¶ 44 – 47 in P2.docx

Codes:

○ Challenge: GPT Challenge: interaction: need to input and direct GPT step by step

Content:

> Interviewer: In the current context of ChatGPT in the code auditing process, what challenges do you think there are?
> P2: Is the challenge the inconvenient part?
> Interviewer: Right, what are the areas that you find not user-friendly and hope to see improved?
> P2: The inconvenient part is that I actually feel like I have to explain things to it slowly for it to understand what I want. From the very beginning when I input a contract, I have to tell it step by step to first analyze, and then guide it step by step to analyze that function and a certain variable. In fact, these tasks should be able to be handled by a dedicated GPT responsible for auditing in this area. As long as I provide the code, it will perform a set of standardized analysis and auditing tasks, without my having to execute them sentence by sentence and step by step all over again.

2:11 ¶ 52 – 55 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: GPT results are too broad; often report similar vulnerablities

Content:

> Interviewer: Do you think there are any challenges in finding errors?
> P2: Yes, the biggest problem is that I feel a bit contaminated. I feel that when I ask GPT to analyze security issues, it always gives the same three things: integer overflow, reentrancy vulnerability, and another issue I can't remember. Anyway, as long as you ask it about security issues, it will report these, saying there's a reentrancy risk, an integer overflow risk, but these are a bit too general.
> Interviewer: The scope is too broad.
> P2: One issue is that the scope is too broad, and the other is that it doesn't conduct any analysis at all. It doesn't check whether there is a reentrancy vulnerability based on the code. It may just rely on some data or retrieved content, and when the reentrancy vulnerability appears frequently or in certain situations, it keeps reporting a reentrancy vulnerability repeatedly. Integer overflow and reentrancy vulnerability are two types that are often reported, regardless of whether the code actually has such issues. It will prioritize reporting these frequently occurring vulnerabilities.

2:12 ¶ 56 – 57 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: interaction: GPT too much false positive

Content:

> Interviewer: Actually, you've answered this just now. When understanding this code, you felt that you needed to ask step by step before you could understand it, right? Are there any other challenges?
> P2: Think about it again. If there are relatively many false positives, this is also a problem. However, there's nothing we can do about it. False positives in GPT are definitely inevitable because it's not a vulnerability scanning tool that makes judgments based on rules.

2:16 ¶ 72 – 75 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: GPT's results are superficial

Content:

> Interviewer: Do you think you need to spend a lot of time verifying?
> P2: What to verify? The result it outputs?
> Interviewer: Right.
> P2: It takes some time. If something is particularly outrageous, you can tell it's wrong at a glance. If there's something that seems a bit uncertain when it's mentioned, I usually write a demo to test it. But it rarely has the ability to raise a question that takes a long time to verify.

3:4 ¶ 3 in P3.docx

Codes:

○ Challenge: GPT needs human to make a final decision

Content:

> Interviewer: Well, when you use ChatGPT, do you think you've encountered any difficulties or challenging aspects?Participant: Uh. I think when asking it to implement a certain function, it may not achieve the ideal result. You really need to modify a lot of things yourself. And for example, the first time you use it, the function it writes may have problems and not work, and may not achieve the desired effect. You still need to modify it yourself. This is what I think ChatGPT may still have a little problem with for now.

3:5 ¶ 3 in P3.docx

Codes:

○ Challenge: Planning: Doubt GPT's accuracy

Content:

> Interviewer: Then, as you just mentioned, ChatGPT can only understand partial information. So, in the planning stage, if you were to interact with GPT, which specific aspects of information would you mainly ask it to help you understand? In this business process, how do you interact with GPT step by step to understand the specific business process?Participant: To be honest, I haven't tried to have ChatGPT understand the entire business. Since I haven't actually asked it to analyze the whole business, I think it might be rather inaccurate and could mislead me. Another thing is that I think it would affect my thinking. Also, for some code and tasks that have information security requirements, you can't put certain business processes into ChatGPT, as I think it would be relatively insecure.

3:6 ¶ 4 in P3.docx

Codes:

○ Challenge: GPT could have over-interpretation

Content:

> Then I found that some of its code gave a feeling of over-interpretation, because there was a function inside it. I saw a function, and when it was called in the original codebase, it was just calling this function, but the implementation of this function was not written in this script. And when interpreting it, it would also interpret the function's functionality. Right, there is a function call, calling this function, and there are also the parameters for this call, but there is no implementation of this function.

3:7 ¶ 4 in P3.docx

Codes:

○ Challenge: privacy issue

Content:

> Actually, when auditing real code, it depends on what is being audited. If some code is not open source, it may involve information security issues. Directly uploading this code to ChatGPT, could there be a risk of leakage? If it is open source, I might, but if it is not open source, I won't.

4:5 ¶ 8 – 9 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: accuracy could decrease when complexity grows

Content:

> Interviewer: Just now we were asking about the entire process of using GPT for assistance, right? Can you describe the differences at different stages between the entire process with ChatGPT assistance and without ChatGPT assistance? For example, did it improve your efficiency, or did it enhance your auditing capabilities?
> P4: Actually, both efficiency and breadth have improved, but efficiency is not guaranteed; different projects may not necessarily see improvements. Some are very broad and concise, some are relatively basic contracts, and GPT can provide relatively high-level evaluations, and it can point out those security issues. However, when dealing with some complex projects, the issues reported by GPT require you to gradually check whether the issues it presents actually exist and whether they are reasonable. These validations are necessary.

4:6 ¶ 10 – 11 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: accuracy could decrease when complexity grows

Content:

> Interviewer: That is to say, after the complexity of this project increases, specifically how, that is, how the complexity increases, it will lead to some problems. What problems will arise when using ChatGPT?
> P4: Regarding the code, it should be the roles existing within the code. If we introduce Oracle, introduce elements like users, as well as some other pools, and also things like staking, the more roles and functions it has, the more its accuracy will decline. This is because it has to consider more factors. Since all you provide is just the code, but you may not have provided all the preconditions. If you don't feed these to GPT, it will make some guesses based on its current environment, which you can be sure are invalid, but GPT will still give them to you, and you still need to spend time to verify them. At this level, the efficiency will be relatively low, and it is also quite troublesome to verify, because some preconditions you think are possible, but GPT may have some other justifications, and you still need to confirm these.

4:7 ¶ 12 – 13 in P4.docx

Codes:

○ Challenge: Comparison GPT vs. human: tradeoff: more knowledge but more verification work; but indeed helpful

Content:

> Interviewer: So, based on what you just described about the entire process, it can actually be divided into understanding, finding vulnerabilities, and then verifying vulnerabilities, right? So, when the task complexity increases, does ChatGPT cause a decrease in efficiency in the verification phase, or does it not necessarily improve efficiency in the early stage?
> P4: It mainly enhances in that aspect, primarily helping you understand how the listed projects operate. It has a relatively clear flowchart or function capabilities, clearly listing what specific functions are being implemented and with whom it interacts. However, the relationships between them may introduce some security issues. Since ChatGPT is equivalent to having a database, it stores more information than normal auditors, and it may also introduce security issues beyond the website. It may not be homogeneous, but it will take them into account. When it presents you with corresponding issues and you need to verify them, you will also need other knowledge, which is where efficiency is somewhat reduced. If your knowledge is insufficient, it will be quite difficult to verify the issues presented by GPT. However, if it provides some understanding, it will make your understanding faster.

4:8 ¶ 16 – 19 in P4.docx

Codes:

○ Challenge: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate ○ user behavior: has some basic knowledge but don't have lists

Content:

Interviewer: Since you have approximately two and a half years of experience in code auditing, you have a relatively systematic and mature knowledge system of your own. Then, when you conduct manual comparisons, for example, you can directly refer to this knowledge system to make a comparison.
P4: Regarding auditing, apart from those scattered security knowledge, when you talk about precision, such as reentrancy, but in fact, specifically, you still need to combine it with the code. For which projects, which reentrancy might lead to some stop-loss issues, and which reentrancy might lead to some calculation issues. For example, read-only reentrancy, its reentrancy logic is different, and the issues are also different. When these accumulate, if you are very familiar with the code and then see similar code snippets, you may consider the corresponding issues. These can be regarded as personal accumulation. However, GPT can think of more code snippets, but it is not very effective when you try to verify them.
Interviewer: Do you make some comparisons? For example, if you have a list, when you look at these codes, do you search for some comparisons, taking some vulnerabilities as your priority to look for? Or do you still rely on your intuition?
P4: With basic code knowledge, first you need to understand that some characteristics of a project, such as price manipulation in a recent reported project, exist. Based on the project, there are some vulnerabilities in its characteristics, as well as some common vulnerabilities and some permission-related vulnerabilities. These actually fall into several levels, including common code-level, no, language-level vulnerabilities, such as the simplest overflow issues, overflow issues before 0.8. You may need to take these into account, based on the version of the language it uses and other factors. There are language-level, logical-level, and common permission-level vulnerabilities, which can be divided into several major categories. When auditing a project, you can apply these accordingly. It can be understood as being similar to the self-study course mentioned earlier, which is actually based on your existing knowledge to see which items need to be applied to which projects. However, when it comes to the actual audit, you still need to understand the project and what modifications it has made. If these modifications are at a new level, you need to understand this and then consider some other issues.

4:9 ¶ 20 – 21 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: low accuracy caused much time wasting on validation

Content:

Interviewer: And the next question, when you use ChatGPT to assist you in code auditing, what do you think are some of its biggest challenges? You just mentioned a challenge in verification, which is the biggest one for you. Are there any others?
P4: Actually, the biggest problem is that the accuracy is not high enough. Some of the issues you need to verify may exist, but if you want to verify them, the efficiency is low. Or, some of the content provided by ChatGPT may lead you astray. You were originally thinking about one thing and focused on this aspect, but GPT gives you other ideas, which may not be correct. When you try to verify these ideas, you may end up going in other directions, which will waste some time.

5:5 ¶ 17 – 34 in P5.docx

Codes:

○ Challenge: Comparison GPT vs. human: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate

Content:

Interviewer: Have you tried using ChatGPT at different stages, for example, when we divide code auditing into different stages?
P5: It might be the two aspects just mentioned, namely the final result output and then the organization of some of his issues in the middle. Regarding what you said earlier about using it for sorting out some basic information of projects, I may not have done that.

> Interviewer: You haven't used ChatGPT for information organization. Why is that?
> P5: Maybe didn't expect it, didn't think of doing it this way before.
> Interviewer: If you were to use it now, you can imagine how you would use it. You can fully imagine, and then how you can make full use of GPT during this process.
> P5: Just throw the white paper of that project at it, let it study it first, and then ask questions using code. This might be a bit better. Previously, we just gave it the code directly, which might have been more difficult to read.
> Interviewer: What was the purpose of having it read the white paper?
> P5: Let it have some understanding of this project, because that's how my current audit process works, so it and I...
> Interviewer: So your goal is to have it simulate that process. Because your process is equivalent to first having a more high-level understanding, and then looking at some details, right? You hope GPT will do the same. What kind of goal do you hope to achieve through this?
> P5: If I were to say something about the auditing project itself, reading the white paper might only give me a limited understanding of the project. In fact, I may not have a very in-depth understanding of it. For many of those relatively professional aspects, they may only be briefly mentioned. If I were to let GPT analyze it, I think its understanding might be deeper. I feel this way because I haven't tried this before.
> Interviewer: What specific professional knowledge are you referring to?
> P5: For example, some mathematical models, economics, etc. This may be because they involve some knowledge related to mathematics, and if you want to calculate, it will be very time-consuming.
> Interviewer: Do you think you've learned something during the process of using it to help you understand? Have you learned something new, or is it just a supplementary tool and you already had this knowledge?
> P5: I probably didn't learn much because he just gave me a result directly, and I have no way of knowing his specific thinking process. It would be better if he could output how his solution came about.
> Interviewer: So you hope that he not only gives you a result, but also provides you with some explanations.
> P5: Right.
> Interviewer: But will you read these explanations carefully once they are provided?
> P5: If his accuracy rate is relatively high, it is still worth learning. However, at present, his learning cost is relatively high, and his false alarm rate is also relatively high. After learning, you may not even know whether what you've learned is correct, which is a bit troublesome.

5:6 ¶ 35 – 42 in P5.docx

Codes:

○ Challenge: why use LLMs not frequently

Content:

> Interviewer: So you don't quite trust some of its results, right?
> P5: Right, the main decision still has to be made by oneself. Because sometimes what it says sounds very reasonable, but upon reflection, there are still significant problems.
> Interviewer: So you use some other software, such as VSCode extensions, or some online (audio) auditing platforms.
> P5: It's auxiliary auditing, right? It's used relatively infrequently.
> Interviewer: What's the reason?
> P5: The reason might be that I haven't had much exposure to tools. Audit tools are relevant, or auxiliary relevant, and it could also be that there aren't currently any good ones, or ones that I'm used to using.
> Interviewer: Okay. So you think using ChatGPT actually doesn't help you save any energy or time, right? And how do you think its performance compares to using traditional methods?
> P5: Sometimes, ChatGPT can be helpful, for example, when you don't have much time or the audit period is relatively short. It can directly provide you with some answers, and you just need to directly judge whether they are correct or not. In this regard, if the audit schedule is tight, it still has a certain effect.

5:8 ¶ 49 – 52 in P5.docx

Codes:

- ○ Challenge: labor intensive to investigate false positives

Content:

> Interviewer: When it's relatively urgent, the output provided to you should have false alarms, right? At this time, doesn't it actually reduce your accuracy?
> P5: Right, those relatively obvious false alarms can be directly investigated and eliminated, and then for those that are clearly confirmed at first glance, you can briefly write them down.
> Interviewer: So it can still be of some help if you use it.
> P5: It is relatively labor-intensive to investigate false positives.

5:9 ¶ 53 – 60 in P5.docx

Codes:

- ○ Challenge: ChatGPT: conversation flow will be interpreted

Content:

> Interviewer: When you use ChatGPT to assist you in code auditing, what do you think is the biggest challenge you've encountered? It doesn't have to be that serious; just some difficulties you've faced or areas you think are not very user-friendly, aside from the false positives just mentioned.
> P5: Let me think. Also, sometimes, for example, after you give it a piece of code, and it has many questions. After you ask question a, when you then ask question b, there is a process equivalent to an interruption at that point. Then if you ask question b, it will combine the answer to question a to give you the answer to question b. This might be rather troublesome, and you need to take a new table to ask question b.
> Interviewer: So it's pretty much like this, right? For example, after having several rounds of conversation with him, you suddenly want to ask another question about the previous one, but also want it to have the context, yet he thinks you want him to continue from the previous question.
> P5: Right.
> Interviewer: I think this is quite interesting.
> P5: In this case, we have to create a new tab to ask it
> Interviewer: But if you start a new conversation to ask it, you'll have to train it again.
> P5: Right, so this is rather troublesome. It might also be that the way I asked led to its misunderstanding. I'm not quite sure either. Anyway, it's always difficult to handle whenever this happens.

5:10 ¶ 61 – 64 in P5.docx

Codes:

- ○ Challenge: false positives

Content:

> Interviewer: Understood. Are there any other areas you find inconvenient, not only when using ChatGPT but also when using traditional methods?
> P5: They're all pretty much the same. They share a commonality in addressing the issue of false positives. Whether it's traditional tools, which also have many false positives, or GPT, both will have them.
> Interviewer: Is this a major consideration for you not to use these tools?
> P5: On the one hand, and on the other hand, the things it reports are not very good. Either they are relatively basic things that you can tell at a glance. If the logicality is relatively high, it fails to recognize them, which is very contradictory. That is, if you use it, it feels inferior to doing it yourself, and if you don't use it, you haven't really used it.

5:13 ¶ 69 – 70 in P5.docx

Codes:

- ○ Challenge: decomposed question regarding the diagram

Content:

> Interviewer: Is it something like generating a logic diagram, which contains different modules and indicates the relationships between them, and then letting GPT do something?
> P5: Right, after expanding, then for some minor points, we can ask GPT about different points.

○ Challenge: ChatGPT: conversation flow will be interpreted

1 Quotations:

5:9 ¶ 53 – 60 in P5.docx

Codes:

○ Challenge: ChatGPT: conversation flow will be interpreted

Content:

> Interviewer: When you use ChatGPT to assist you in code auditing, what do you think is the biggest challenge you've encountered? It doesn't have to be that serious; just some difficulties you've faced or areas you think are not very user-friendly, aside from the false positives just mentioned.
> P5: Let me think. Also, sometimes, for example, after you give it a piece of code, and it has many questions. After you ask question a, when you then ask question b, there is a process equivalent to an interruption at that point. Then if you ask question b, it will combine the answer to question a to give you the answer to question b. This might be rather troublesome, and you need to take a new table to ask question b.
> Interviewer: So it's pretty much like this, right? For example, after having several rounds of conversation with him, you suddenly want to ask another question about the previous one, but also want it to have the context, yet he thinks you want him to continue from the previous question.
> P5: Right.
> Interviewer: I think this is quite interesting.
> P5: In this case, we have to create a new tab to ask it
> Interviewer: But if you start a new conversation to ask it, you'll have to train it again.
> P5: Right, so this is rather troublesome. It might also be that the way I asked led to its misunderstanding. I'm not quite sure either. Anyway, it's always difficult to handle whenever this happens.

○ Challenge: Comparison GPT vs. human: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate

1 Quotations:

5:5 ¶ 17 – 34 in P5.docx

Codes:

○ Challenge: Comparison GPT vs. human: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate

Content:

> Interviewer: Have you tried using ChatGPT at different stages, for example, when we divide code auditing into different stages?
> P5: It might be the two aspects just mentioned, namely the final result output and then the organization of some of his issues in the middle. Regarding what you said earlier about using it for sorting out some basic information of projects, I may not have done that.
> Interviewer: You haven't used ChatGPT for information organization. Why is that?
> P5: Maybe didn't expect it, didn't think of doing it this way before.
> Interviewer: If you were to use it now, you can imagine how you would use it. You can fully imagine, and then how you can make full use of GPT during this process.
> P5: Just throw the white paper of that project at it, let it study it first, and then ask questions using code. This might be a bit better. Previously, we just gave it the code directly, which might have been more difficult to read.
> Interviewer: What was the purpose of having it read the white paper?
> P5: Let it have some understanding of this project, because that's how my current audit process works, so it and I...
> Interviewer: So your goal is to have it simulate that process. Because your process is equivalent to first having a more high-level understanding, and then looking at some details, right? You hope GPT will do the same. What kind of goal do you hope to achieve through this?
> P5: If I were to say something about the auditing project itself, reading the white paper might only give me a limited understanding of the project. In fact, I may not have a very in-depth understanding of it. For many of those relatively professional aspects, they may only be briefly mentioned. If I were

to let GPT analyze it, I think its understanding might be deeper. I feel this way because I haven't tried this before.
Interviewer: What specific professional knowledge are you referring to?
P5: For example, some mathematical models, economics, etc. This may be because they involve some knowledge related to mathematics, and if you want to calculate, it will be very time-consuming.
Interviewer: Do you think you've learned something during the process of using it to help you understand? Have you learned something new, or is it just a supplementary tool and you already had this knowledge?
P5: I probably didn't learn much because he just gave me a result directly, and I have no way of knowing his specific thinking process. It would be better if he could output how his solution came about.
Interviewer: So you hope that he not only gives you a result, but also provides you with some explanations.
P5: Right.
Interviewer: But will you read these explanations carefully once they are provided?
P5: If his accuracy rate is relatively high, it is still worth learning. However, at present, his learning cost is relatively high, and his false alarm rate is also relatively high. After learning, you may not even know whether what you've learned is correct, which is a bit troublesome.

○ Challenge: Comparison GPT vs. human: tradeoff: more knowledge but more verification work; but indeed helpful

1 Quotations:

4:7 ¶ 12 – 13 in P4.docx

Codes:

○ Challenge: Comparison GPT vs. human: tradeoff: more knowledge but more verification work; but indeed helpful

Content:

Interviewer: So, based on what you just described about the entire process, it can actually be divided into understanding, finding vulnerabilities, and then verifying vulnerabilities, right? So, when the task complexity increases, does ChatGPT cause a decrease in efficiency in the verification phase, or does it not necessarily improve efficiency in the early stage?
P4: It mainly enhances in that aspect, primarily helping you understand how the listed projects operate. It has a relatively clear flowchart or function capabilities, clearly listing what specific functions are being implemented and with whom it interacts. However, the relationships between them may introduce some security issues. Since ChatGPT is equivalent to having a database, it stores more information than normal auditors, and it may also introduce security issues beyond the website. It may not be homogeneous, but it will take them into account. When it presents you with corresponding issues and you need to verify them, you will also need other knowledge, which is where efficiency is somewhat reduced. If your knowledge is insufficient, it will be quite difficult to verify the issues presented by GPT. However, if it provides some understanding, it will make your understanding faster.

○ Challenge: decomposed question regarding the diagram

1 Quotations:

5:13 ¶ 69 – 70 in P5.docx

Codes:

○ Challenge: decomposed question regarding the diagram

Content:

Interviewer: Is it something like generating a logic diagram, which contains different modules and indicates the relationships between them, and then letting GPT do something?
P5: Right, after expanding, then for some minor points, we can ask GPT about different points.

○ Challenge: false positives

1 Quotations:

5:10 ¶ 61 – 64 in P5.docx

Codes:

○ Challenge: false positives

Content:

> Interviewer: Understood. Are there any other areas you find inconvenient, not only when using ChatGPT but also when using traditional methods?
> P5: They're all pretty much the same. They share a commonality in addressing the issue of false positives. Whether it's traditional tools, which also have many false positives, or GPT, both will have them.
> Interviewer: Is this a major consideration for you not to use these tools?
> P5: On the one hand, and on the other hand, the things it reports are not very good. Either they are relatively basic things that you can tell at a glance. If the logicality is relatively high, it fails to recognize them, which is very contradictory. That is, if you use it, it feels inferior to doing it yourself, and if you don't use it, you haven't really used it.

○ Challenge: GPT can only step by step but human can be jumpy

1 Quotations:

1:25 ¶ 55 in P1.docx

Codes:

○ Challenge: GPT can only step by step but human can be jumpy

Content:

> Another thing is that human thinking is relatively jumpy, with meaningful leaps. So in machines, it's like this. For example, many so-called math-oriented knowledge related to mathematics, but without very special logical errors, is evaluated, so basically it's all in the realm of uncertainty. Then I need to continue to improve the vulnerability library, increase knowledge, and expand knowledge. The more general knowledge there is, the more effective it will be after coordinating with planning, making it more capable of thinking leaps like humans. To put it simply, for example, if I match 10 pieces of knowledge, ChatGPT will select the first 5, and the last 5 may be missed. How can I make ChatGPT also scan out the last 5? To put it simply, I remove the first 5, extract them from the ranking, and move the last 5 forward. This is the general idea of the solution. I don't know if you can understand what I'm saying. If the first 5 are all of one category, I may group them together or take a closer look at them. I try to make the so-called key concepts it matches as diverse as possible, which is my ultimate goal. The better its diversity, the closer it can get to the result of human jumpy thinking.

○ Challenge: GPT challenge: interaction: interaction: everytime human needs to interact with GPT from the start and copy the prompt

1 Quotations:

1:27 ¶ 60 – 65 in P1.docx

Codes:

○ Challenge: GPT challenge: interaction: interaction: everytime human needs to interact with GPT from the start and copy the prompt

Content:

> Interviewer: The question just asked used the term "challenge". Could you demonstrate how you interact with ChatGPT?
> P1: One of the things I did before was the first solution, the manual solution, which involves inputting this thing and then directly copying the long prompt I just mentioned to it, because I've

> been adjusting it for a long time. Then at this point, ChatGPT will give an answer that looks very reasonable, seemingly very much like it, but sometimes it's not.
> Interviewer: So it's equivalent to starting a conversation from scratch every time.
> P1: It can be understood in this way. Each function has a possible occurrence of a new vulnerability each time.
> Interviewer: So there are many repetitive steps in it.
> P1: Right, to put it bluntly, this prompt, look at where the prompt I just found came from, I dug it out from the chat history. In fact, I don't have a good place to store this prompt. Ok, look at this part, it can give a result that looks very reasonable, and it does it very reliably, but there may be two situations. First, it's too complex, what it writes is quite complex. So at this time, it will be asked to explain this vulnerability in detail, step by step with reference to the code, and basically finish the output, and then ask it to output this.

○ Challenge: GPT challenge: interaction: lack of enough context/cannot input too much code

1 Quotations:

2:4 ¶ 12 – 19 in P2.docx

Codes:

○ Challenge: GPT challenge: interaction: lack of enough context/cannot input too much code ○ Planning: ChatGPT for understanding: interaction

Content:

> Interviewer: So different variables will have different associated business logic.
> P2: Such a situation may exist, but the prerequisite is that you must first use your own experience or identify that it has two sets of logic.
> Interviewer: So it's like when you're understanding this code, you first use your own judgment, then form a general internal understanding of this code, and then ask about the relatively minor points.
> P2: This way is more efficient. If I can figure it out in a short time, it will be even more efficient. If not, I can also turn to ChatGPT and start asking from scratch.
> Interviewer: Okay, in most cases, can you tell on your own or not?
> P2: In most cases, people are just too lazy to read themselves and hand it over to GPT first.
> Interviewer: But ChatGPT doesn't have the ability to digest the code of such a large project; instead, it can only handle a small snippet of code.
> P2: Essentially, we identify them one by one as contract files; a project would be too large, so there's no other way.

○ Challenge: GPT Challenge: interaction: need to input and direct GPT step by step

1 Quotations:

2:9 ¶ 44 – 47 in P2.docx

Codes:

○ Challenge: GPT Challenge: interaction: need to input and direct GPT step by step

Content:

> Interviewer: In the current context of ChatGPT in the code auditing process, what challenges do you think there are?
> P2: Is the challenge the inconvenient part?
> Interviewer: Right, what are the areas that you find not user-friendly and hope to see improved?
> P2: The inconvenient part is that I actually feel like I have to explain things to it slowly for it to understand what I want. From the very beginning when I input a contract, I have to tell it step by step to first analyze, and then guide it step by step to analyze that function and a certain variable. In fact, these tasks should be able to be handled by a dedicated GPT responsible for auditing in this area. As long as I provide the code, it will perform a set of standardized analysis and auditing tasks, without my having to execute them sentence by sentence and step by step all over again.

○ Challenge: GPT challenge: planning: lack of customized scanning

1 Quotations:

1:22 ¶ 50 – 51 in P1.docx

Codes:

○ Challenge: GPT challenge: planning: lack of customized scanning

Content:

> Interviewer: Is there any other challenge in the second stage?
> P1: Specifically, I don't remember very clearly either. There might be redundant processes in knowledge. During the planning phase, knowledge might have some redundant processes. Specifically, there are some protected functions that ChatGPT also scanned, which is actually the so-called "only owner" part mentioned earlier. That is, ChatGPT also scanned some functions that didn't need to be scanned, but I've resolved this issue. Some redundant functions that didn't need to be scanned were also scanned by me. This is one aspect. I didn't give it a long enough chain of thought. I simply told it to scan all functions, and it didn't think carefully about which ones to scan and which ones not to. In fact, even when written in great detail, humans may not be able to quickly understand and make judgments.

○ Challenge: GPT challenge: reasoning: context is limited (due to the data cannot be imported totally), especially for reasoning and validating

1 Quotations:

1:21 ¶ 44 – 45 in P1.docx

Codes:

○ Challenge: GPT challenge: reasoning: context is limited (due to the data cannot be imported totally), especially for reasoning and validating

Content:

> Interviewer: Right, I understand, so it's equivalent to saying that ChatGPT can participate in all three stages of your planning, reasoning, and validating. When using this conversation-based approach to execute your process, what do you think is the biggest challenge you've encountered?
> P1: There are quite a few. The first issue is that ChatGPT has insufficient context, resulting in its scope being incomplete, while humans can consider things from a global perspective. For example, if using a checklist, it often produces some false positives. The cause of these false positives is actually because I am querying and detecting at the function level, so sometimes it will produce some false positives. That is, because there are already checks, corresponding judgments, and corresponding restrictions elsewhere, it still thinks there are vulnerabilities, which is definitely a false positive. In fact, it is still a context issue.

○ Challenge: GPT challenge: reasoning: GPT cannot do self-learning

1 Quotations:

1:23 ¶ 52 – 53 in P1.docx

Codes:

○ Challenge: GPT challenge: reasoning: GPT cannot do self-learning

Content:

> Interviewer: Even if ChatGPT returns a relatively detailed description, people may not be able to tell that it is wrong.
> P1: Because it's too complex, some vulnerabilities are overly complicated, and ChatGPT's descriptions aren't particularly good either. So sometimes I'll add a sentence, for example, I'll say "Please explain it to me line by line according to the specific code." I usually add this sentence because I simply can't understand those things; they're too obscure. Then a problem arises during the reasoning process: the ability to draw inferences from one instance is not strong enough. I

> might only ask about the key concept of a certain functionality project and not ask about anything else, and then I assume that similar vulnerabilities won't occur.

○ Challenge: GPT challenge: reasoning: hard to input enough knowledge

1 Quotations:

2:7 ¶ 36 – 37 in P2.docx

Codes:

○ Challenge: GPT challenge: reasoning: hard to input enough knowledge

Content:

> Interviewer: This note refers to yours.
> P2: A case analysis of Hundred finance was attacked, and that attack was caused by the loss of computational precision, which is equivalent to a case. I feel that there are probably many pictures inside that cannot be uploaded, so if it can only view text, it cannot understand, because there are some screenshots that were directly pasted when doing the analysis, not in text form, and it can only upload text when uploading.

○ Challenge: GPT challenge: validating: accuracy could decrease when complexity grows

2 Quotations:

4:5 ¶ 8 – 9 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: accuracy could decrease when complexity grows

Content:

> Interviewer: Just now we were asking about the entire process of using GPT for assistance, right? Can you describe the differences at different stages between the entire process with ChatGPT assistance and without ChatGPT assistance? For example, did it improve your efficiency, or did it enhance your auditing capabilities?
> P4: Actually, both efficiency and breadth have improved, but efficiency is not guaranteed; different projects may not necessarily see improvements. Some are very broad and concise, some are relatively basic contracts, and GPT can provide relatively high-level evaluations, and it can point out those security issues. However, when dealing with some complex projects, the issues reported by GPT require you to gradually check whether the issues it presents actually exist and whether they are reasonable. These validations are necessary.

4:6 ¶ 10 – 11 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: accuracy could decrease when complexity grows

Content:

> Interviewer: That is to say, after the complexity of this project increases, specifically how, that is, how the complexity increases, it will lead to some problems. What problems will arise when using ChatGPT?
> P4: Regarding the code, it should be the roles existing within the code. If we introduce Oracle, introduce elements like users, as well as some other pools, and also things like staking, the more roles and functions it has, the more its accuracy will decline. This is because it has to consider more factors. Since all you provide is just the code, but you may not have provided all the preconditions. If you don't feed these to GPT, it will make some guesses based on its current environment, which you can be sure are invalid, but GPT will still give them to you, and you still need to spend time to verify them. At this level, the efficiency will be relatively low, and it is also quite troublesome to verify, because some preconditions you think are possible, but GPT may have some other justifications, and you still need to confirm these.

○ Challenge: GPT challenge: validating: GPT results are too broad; often report similar vulnerablities

1 Quotations:

2:11 ¶ 52 – 55 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: GPT results are too broad; often report similar vulnerablities

Content:

> Interviewer: Do you think there are any challenges in finding errors?
> P2: Yes, the biggest problem is that I feel a bit contaminated. I feel that when I ask GPT to analyze security issues, it always gives the same three things: integer overflow, reentrancy vulnerability, and another issue I can't remember. Anyway, as long as you ask it about security issues, it will report these, saying there's a reentrancy risk, an integer overflow risk, but these are a bit too general.
> Interviewer: The scope is too broad.
> P2: One issue is that the scope is too broad, and the other is that it doesn't conduct any analysis at all. It doesn't check whether there is a reentrancy vulnerability based on the code. It may just rely on some data or retrieved content, and when the reentrancy vulnerability appears frequently or in certain situations, it keeps reporting a reentrancy vulnerability repeatedly. Integer overflow and reentrancy vulnerability are two types that are often reported, regardless of whether the code actually has such issues. It will prioritize reporting these frequently occurring vulnerablities.

○ Challenge: GPT challenge: validating: GPT write reports/attack script due to the lack of context

1 Quotations:

1:26 ¶ 56 – 59 in P1.docx

Codes:

○ Challenge: GPT challenge: validating: GPT write reports/attack script due to the lack of context

Content:

> Interviewer: So your goal is to optimize ChatGPT, improve your usage methods, reduce false alarm rates, and increase efficiency.
> P1: Right, reducing false positive rates and increasing effectiveness are my current goals. Additionally, humans can actually understand most of the results returned by ChatGPT, but this is after all manual work, not automated. I think automated validating might still be a matter of context, because when humans write so-called attack scripts, they actually comprehensively consider the front and back limitations or code of the entire project, but programs actually find it difficult to consider such comprehensive code or some limitations. Because an attack script itself can be understood as a sub-function, but this sub-function calls almost all other functions of the project, which imposes a requirement that you need to have an understanding of all other functions, and humans can achieve this.
> Interviewer: So basically ChatGPT can't write this attack script now.
> P1: Let me put it this way. It has two indicators. One is grammatical correctness, and in terms of grammatical correctness, it's okay, but in terms of logical comprehensiveness, it's not okay. Right, that's about it.

○ Challenge: GPT challenge: validating: GPT's results are superficial

1 Quotations:

2:16 ¶ 72 – 75 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: GPT's results are superficial

Content:

> Interviewer: Do you think you need to spend a lot of time verifying?
> P2: What to verify? The result it outputs?
> Interviewer: Right.
> P2: It takes some time. If something is particularly outrageous, you can tell it's wrong at a glance. If there's something that seems a bit uncertain when it's mentioned, I usually write a demo to test it. But it rarely has the ability to raise a question that takes a long time to verify.

○ Challenge: GPT challenge: validating: interaction: GPT too much false positive

1 Quotations:

2:12 ¶ 56 – 57 in P2.docx

Codes:

○ Challenge: GPT challenge: validating: interaction: GPT too much false positive

Content:

> Interviewer: Actually, you've answered this just now. When understanding this code, you felt that you needed to ask step by step before you could understand it, right? Are there any other challenges?
> P2: Think about it again. If there are relatively many false positives, this is also a problem. However, there's nothing we can do about it. False positives in GPT are definitely inevitable because it's not a vulnerability scanning tool that makes judgments based on rules.

○ Challenge: GPT challenge: validating: low accuracy caused much time wasting on validation

1 Quotations:

4:9 ¶ 20 – 21 in P4.docx

Codes:

○ Challenge: GPT challenge: validating: low accuracy caused much time wasting on validation

Content:

> Interviewer: And the next question, when you use ChatGPT to assist you in code auditing, what do you think are some of its biggest challenges? You just mentioned a challenge in verification, which is the biggest one for you. Are there any others?
> P4: Actually, the biggest problem is that the accuracy is not high enough. Some of the issues you need to verify may exist, but if you want to verify them, the efficiency is low. Or, some of the content provided by ChatGPT may lead you astray. You were originally thinking about one thing and focused on this aspect, but GPT gives you other ideas, which may not be correct. When you try to verify these ideas, you may end up going in other directions, which will waste some time.

○ Challenge: GPT could have over-interpretation

1 Quotations:

3:6 ¶ 4 in P3.docx

Codes:

○ Challenge: GPT could have over-interpretation

Content:

> Then I found that some of its code gave a feeling of over-interpretation, because there was a function inside it. I saw a function, and when it was called in the original codebase, it was just calling this function, but the implementation of this function was not written in this script. And when interpreting it, it would also interpret the function's functionality. Right, there is a function call, calling this function, and there are also the parameters for this call, but there is no implementation of this function.

○ Challenge: GPT hallucinations

1 Quotations:

1:24 ¶ 54 in P1.docx

Codes:

○ Challenge: GPT hallucinations

Content:

> Another example is that there are some so - called key concepts that simply should not be used in this scenario. At this point, if you force a search, ChatGPT may exhibit some hallucinations, resulting in false positives. This is because ChatGPT sometimes tries to answer your questions forcefully and will start to go off on a tangent.

○ Challenge: GPT needs human to make a final decision

1 Quotations:

3:4 ¶ 3 in P3.docx

Codes:

○ Challenge: GPT needs human to make a final decision

Content:

> Interviewer: Well, when you use ChatGPT, do you think you've encountered any difficulties or challenging aspects?Participant: Uh. I think when asking it to implement a certain function, it may not achieve the ideal result. You really need to modify a lot of things yourself. And for example, the first time you use it, the function it writes may have problems and not work, and may not achieve the desired effect. You still need to modify it yourself. This is what I think ChatGPT may still have a little problem with for now.

○ Challenge: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate

1 Quotations:

4:8 ¶ 16 – 19 in P4.docx

Codes:

○ Challenge: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate ○ user behavior: has some basic knowledge but don't have lists

Content:

> Interviewer: Since you have approximately two and a half years of experience in code auditing, you have a relatively systematic and mature knowledge system of your own. Then, when you conduct manual comparisons, for example, you can directly refer to this knowledge system to make a comparison.
> P4: Regarding auditing, apart from those scattered security knowledge, when you talk about precision, such as reentrancy, but in fact, specifically, you still need to combine it with the code. For which projects, which reentrancy might lead to some stop-loss issues, and which reentrancy might lead to some calculation issues. For example, read-only reentrancy, its reentrancy logic is different, and the issues are also different. When these accumulate, if you are very familiar with the code and then see similar code snippets, you may consider the corresponding issues. These can be regarded as personal accumulation. However, GPT can think of more code snippets, but it is not very effective when you try to verify them.
> Interviewer: Do you make some comparisons? For example, if you have a list, when you look at these codes, do you search for some comparisons, taking some vulnerabilities as your priority to look for? Or do you still rely on your intuition?
> P4: With basic code knowledge, first you need to understand that some characteristics of a project, such as price manipulation in a recent reported project, exist. Based on the project, there are some

> vulnerabilities in its characteristics, as well as some common vulnerabilities and some permission-related vulnerabilities. These actually fall into several levels, including common code-level, no, language-level vulnerabilities, such as the simplest overflow issues, overflow issues before 0.8. You may need to take these into account, based on the version of the language it uses and other factors. There are language-level, logical-level, and common permission-level vulnerabilities, which can be divided into several major categories. When auditing a project, you can apply these accordingly. It can be understood as being similar to the self-study course mentioned earlier, which is actually based on your existing knowledge to see which items need to be applied to which projects. However, when it comes to the actual audit, you still need to understand the project and what modifications it has made. If these modifications are at a new level, you need to understand this and then consider some other issues.

○ Challenge: labor intensive to investigate false positives

1 Quotations:

5:8 ¶ 49 – 52 in P5.docx

Codes:

○ Challenge: labor intensive to investigate false positives

Content:

> Interviewer: When it's relatively urgent, the output provided to you should have false alarms, right? At this time, doesn't it actually reduce your accuracy?
> P5: Right, those relatively obvious false alarms can be directly investigated and eliminated, and then for those that are clearly confirmed at first glance, you can briefly write them down.
> Interviewer: So it can still be of some help if you use it.
> P5: It is relatively labor-intensive to investigate false positives.

○ Challenge: Planning: Doubt GPT's accuracy

1 Quotations:

3:5 ¶ 3 in P3.docx

Codes:

○ Challenge: Planning: Doubt GPT's accuracy

Content:

> Interviewer: Then, as you just mentioned, ChatGPT can only understand partial information. So, in the planning stage, if you were to interact with GPT, which specific aspects of information would you mainly ask it to help you understand? In this business process, how do you interact with GPT step by step to understand the specific business process?Participant: To be honest, I haven't tried to have ChatGPT understand the entire business. Since I haven't actually asked it to analyze the whole business, I think it might be rather inaccurate and could mislead me. Another thing is that I think it would affect my thinking. Also, for some code and tasks that have information security requirements, you can't put certain business processes into ChatGPT, as I think it would be relatively insecure.

○ Challenge: planning: lack of customized scanning

0 Quotations

○ Challenge: privacy issue

1 Quotations:

3:7 ¶ 4 in P3.docx

Codes:

○ Challenge: privacy issue

Content:

> Actually, when auditing real code, it depends on what is being audited. If some code is not open source, it may involve information security issues. Directly uploading this code to ChatGPT, could there be a risk of leakage? If it is open source, I might, but if it is not open source, I won't.

○ Challenge: why use LLMs not frequently

1 Quotations:

5:6 ¶ 35 – 42 in P5.docx

Codes:

○ Challenge: why use LLMs not frequently

Content:

> Interviewer: So you don't quite trust some of its results, right?
> P5: Right, the main decision still has to be made by oneself. Because sometimes what it says sounds very reasonable, but upon reflection, there are still significant problems.
> Interviewer: So you use some other software, such as VSCode extensions, or some online (audio) auditing platforms.
> P5: It's auxiliary auditing, right? It's used relatively infrequently.
> Interviewer: What's the reason?
> P5: The reason might be that I haven't had much exposure to tools. Audit tools are relevant, or auxiliary relevant, and it could also be that there aren't currently any good ones, or ones that I'm used to using.
> Interviewer: Okay. So you think using ChatGPT actually doesn't help you save any energy or time, right? And how do you think its performance compares to using traditional methods?
> P5: Sometimes, ChatGPT can be helpful, for example, when you don't have much time or the audit period is relatively short. It can directly provide you with some answers, and you just need to directly judge whether they are correct or not. In this regard, if the audit schedule is tight, it still has a certain effect.

○ ChatGPT's help

1 Quotations:

6:7 ¶ 19 – 22 in P6.docx

Codes:

○ ChatGPT's help: decrease the learning cost

Content:

> Right, another characteristic is that perhaps this supply chain has been popular or emerged for only one year or half a year. There's a question: do auditors have enough energy to quickly and proficiently master this language? Previously, for auditors, if, for example, a certain supply chain emerged, and there were several mainstream languages used in this supply chain, such as GS, Move, and Rust, these three languages might be involved. Before ChatGPT, for auditors, if they wanted to audit a project, they first had to be relatively familiar with this language. They might not use it for large-scale tool development, but they definitely had to be more familiar with it if they were to conduct an audit. They needed to know where problems might occur at the language level, and only then could they audit for vulnerabilities at the business level while ensuring that there were no issues at the language level.
> Interviewer: Let me interrupt. So now, with the advent of ChatGPT, it's equivalent to spending less time on the language level.
> P6: There are fewer because for many language learning scenarios, for example, when I passed JavaScript, it may be more of a scripting language, and its one-week learning cost is relatively low. However, for some languages, if you want to learn them to a level where you can truly conduct audits, you need two to three months to study. Right, but actually most vulnerabilities are logical vulnerabilities, which are independent of the language. There's nothing wrong with the language

> itself, and the code written in it is also fine, but when different logics are assembled, vulnerabilities will appear.
> Yes, so actually when we want to solve this kind of problem, the first difficulty is the language issue. For example, some languages are relatively more difficult. Take Rust, a language that gives many people headaches, especially in the supply chain. Many people, including auditors in their work, actually use this language. Most of the projects that private auditors in auditing firms receive are on Ethereum, but the contracts on Ethereum are basically not in the form of Rust. So there are many supply chains that are relatively niche compared to Ethereum, which use Rust to write their smart contracts. However, an auditing firm may receive only two or three such orders out of 10 or 20 orders.

○ ChatGPT's help: decrease the learning cost

1 Quotations:

6:7 ¶ 19 – 22 in P6.docx

Codes:

○ ChatGPT's help: decrease the learning cost

Content:

> Right, another characteristic is that perhaps this supply chain has been popular or emerged for only one year or half a year. There's a question: do auditors have enough energy to quickly and proficiently master this language? Previously, for auditors, if, for example, a certain supply chain emerged, and there were several mainstream languages used in this supply chain, such as GS, Move, and Rust, these three languages might be involved. Before ChatGPT, for auditors, if they wanted to audit a project, they first had to be relatively familiar with this language. They might not use it for large-scale tool development, but they definitely had to be more familiar with it if they were to conduct an audit. They needed to know where problems might occur at the language level, and only then could they audit for vulnerabilities at the business level while ensuring that there were no issues at the language level.
> Interviewer: Let me interrupt. So now, with the advent of ChatGPT, it's equivalent to spending less time on the language level.
> P6: There are fewer because for many language learning scenarios, for example, when I passed JavaScript, it may be more of a scripting language, and its one-week learning cost is relatively low. However, for some languages, if you want to learn them to a level where you can truly conduct audits, you need two to three months to study. Right, but actually most vulnerabilities are logical vulnerabilities, which are independent of the language. There's nothing wrong with the language itself, and the code written in it is also fine, but when different logics are assembled, vulnerabilities will appear.
> Yes, so actually when we want to solve this kind of problem, the first difficulty is the language issue. For example, some languages are relatively more difficult. Take Rust, a language that gives many people headaches, especially in the supply chain. Many people, including auditors in their work, actually use this language. Most of the projects that private auditors in auditing firms receive are on Ethereum, but the contracts on Ethereum are basically not in the form of Rust. So there are many supply chains that are relatively niche compared to Ethereum, which use Rust to write their smart contracts. However, an auditing firm may receive only two or three such orders out of 10 or 20 orders.

○ collaboration on a single project

1 Quotations:

6:9 ¶ 28 – 29 in P6.docx

Codes:

○ collaboration on a single project

Content:

> Interviewer: I see. So generally, when a company assigns a project to your company, does your company then assign this project to, say, three or four auditors to conduct human moderation simultaneously, or does it break the project down and have each person moderate a portion?

> P6: From the companies I've seen and heard of, they should all have three or four people conducting human moderation on a single project together, because the vulnerability may not only exist in a certain part of this project; it may be caused by the logical relationship between these two modules.

○ estimation is easier for companies to make decisions

1 Quotations:

6:5 ¶ 17 in P6.docx

Codes:

○ estimation is easier for companies to make decisions

Content:

> P6: It saves a lot of trouble for auditing firms, and also saves a great deal of trouble for both auditing firms and auditors. Moreover, for example, the audit time assessment for projects provided by such large models is relatively fairer. That's one thing. Then, the first thing a project party does when approaching an auditing firm is to evaluate whether we should enter into a cooperation, and this evaluation process is basically carried out by professional auditors.

○ Expectation

15 Quotations:

1:14 ¶ 33 – 34 in P1.docx

Codes:

○ Expectation: domain knowledge augmented: opportunity: how new auditor do code auditing-solution: one GPT to ask another GPT

Content:

> Interviewer: It's equivalent to how you transfer your knowledge to a newbie, for example, how to let a newbie use the Knowledge Base of an export.
> P1: Right, so this kind of requirement is relatively difficult. I haven't come up with a good solution yet. The key point is to assume that GPT has absolutely no knowledge of these things and let it find these loopholes from scratch. There are roughly two methods here. One is to feed the results of ChatGPT back to ChatGPT. For example, I first ask what this is for, and then ask, based on this judgment, where do you think the loopholes might occur? For an auditor, they might think the loopholes could occur in the if statement. So what do you think might be missing from the judgment? I would guide it step by step to think like a human. Do you understand what I mean? Ok, this is the first solution.

1:33 ¶ 81 – 82 in P1.docx

Codes:

○ Expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

Content:

> Interviewer: Okay, and now I'd like to ask the last question. If we were to design a new interface to help you interact with ChatGPT, what expectations do you have and what features would you like it to have?
> P1: Actually, I understand this part. I don't know if you've looked at the Audit Wizard. For auditors, I think there may be two different levels of requirements. The first, with relatively high expectations, is something like the Audit Wizard, which not only can ask questions about code but also can integrate with some different tools or other things. This is perhaps what we ultimately hope for. Of course, the Audit Wizard itself is not particularly good; it doesn't make much use of AI. In contrast, the other requirement is that we may not import corresponding code but simply copy a contract file in. If that's the case, expectations may be higher. It would be best if there were some very effective trick prompts available, like the prompt I used before. If there were such prompts, I think it would be better. Because, to be honest, the completeness of the prompt greatly affects GPT's response results, which means that the better the prompt, the harder it is to obtain. The prompt I have here is

one that I adjusted after a long time. If there were a collection of such prompts, I think it would be better.

2:14 ¶ 65 – 69 in P2.docx

Codes:

○ Expectation: data flowchart: support understanding using data flows like business flowchart

Content:

P2: Saying there are a dozen is an exaggeration, but there has been an increase. I've thought about some of the issues it pointed out. For some of the bugs it identified, the risk level should be relatively low. It tends to find more bugs with low risk levels or at the Information level, but for truly high-risk vulnerabilities, GPT can point out relatively few. While the quantity has increased, in terms of quality, the output of high-quality vulnerabilities is still relatively low.
Interviewer: You think so. We plan to design and develop an interface, software, or platform to help users better interact with GPT, to assist your work. Which stage do you think you most need help with? And what functions do you most expect?
P2: I hope it can provide more information in analyzing the overall project structure, especially if it can generate some business flowcharts or function call graphs.
Interviewer: It means all the business processes related to the list you just mentioned, anyway, just to help you understand.
P2: Right, it's about reverse-engineering from the code to the entire business process, understanding what kind of business it is, which functions users can call, and what role they play in the overall business. Because after we obtain the code and then look at its official documentation, the website's promotion, or its white paper, there are actually discrepancies, or rather, it's still not easy to understand. What's written on its official website sounds great, but in terms of its code implementation, it's hard to understand and difficult to correlate with what's on the official website. However, if you use GPT, it will explain based on the code what it's doing and what its business process is like.

2:15 ¶ 70 – 71 in P2.docx

Codes:

○ Expectation: future thoughts: one-click generation of vulnerability report with high accuracy

Content:

Interviewer: During the process of assisting you in finding errors, what features do you most expect to have to better enable you to use GPT?
P2: What I most hope for is, of course, one-click generation of vulnerability reports. It should identify vulnerabilities and then generate vulnerability reports with just one click, just like a vulnerability scanner. This way, I don't have to check if there are any issues here and probe step by step. Just like a vulnerability scanner, it knows what it needs to do, then identifies, analyzes, and outputs the results in the form of a report.

2:18 ¶ 77 in P2.docx

Codes:

○ Expectation: data flowchart: 1. can we use data flowchart to support the chain? 2. can we use ata flowchartto support more context input?

Content:

Without providing any information to GPT, if we directly ask GPT to find a vulnerability, it is actually very difficult for it to discover these vulnerabilities through a highly reproducible process. It may discover them, but this process may not form a methodology, making it difficult to reproduce. However, I think that if, for example, my static engine or some other tools can provide such a trajectory, a main Line of Business or multiple parallel sub-Lines of Business, and can replace our current approach from function to scenario and from scenario to vulnerability in this way, I think it may be more effective. I think this is a very critical point because this key point actually solves our current biggest problem, which is the context problem. The biggest drawback of our current use of a single function is that it cannot cover the context, and it is difficult to obtain the context. I think if this can be provided through some other means.

2:19 ¶ 78 – 82 in P2.docx

Codes:

○ Expectation: data flowchart: finding links among nodes (variables, functions) and use it to support ChatGPT understanding

Content:

> Interviewer: You mean going to draw that diagram, right?
> P2: Right, I might, for example, it might be a point like this, then go to the next point, and then to the next point, you should be able to see it. This is another process, perhaps another process has reached this point. To put it simply, we actually want to extract different paths and different main business logics from the code. Then I'm thinking about this business logic line, but this business route is actually not very good. First of all, it cannot cover the business; it's just an incomprehensible fixed pattern. I think maybe we can use ChatGPT to extract this business-related process. The business process should actually be in our database, in our Knowledge Base, and there should be corresponding rules during the reasoning process.
> P2: Just now when we were talking about business processes, I thought about it. This business process can actually be, I'm not sure if it can be regarded as a finite-state machine. For example, in a finite-state machine, it starts with a state, then you call a function, and it will enter the next state. How each state changes depends on the modifications made to those global variables. I think one thing that may involve human-machine interaction is that what has been presented like this in the past is actually relatively critical information.
> Interviewer: How to make this information more effective?
> P2: More effectively demonstrate it, whether using the text results returned by ChatGPT and adding some analysis of visual graphs, viewing visual graphs, or using other solutions to show the main business starting points, endpoints, and processes within the current contract, as well as the relationships between different business flows. I think this may be a relatively critical aspect in the audit process. Of course, this is also helpful for auditors. How auditors ultimately use this and how different auditors use it. I think that the creation and display of a process like the one mentioned earlier actually have an impact and are helpful for all three steps of planning, reasoning, and validating. Perhaps in the end, for example, the location where vulnerabilities occur may be in those relatively long processes or in those with more relationships with other processes. Maybe those with only one or two nodes and only one or two business flows may have a lower probability of generating vulnerabilities, which may be useful for planning.

4:11 ¶ 24 – 26 in P4.docx

Codes:

○ Expectation: domain knowledge augmented: hope can augment GPT some domain knowledge for high efficiency and effectiveness

Content:

> Interviewer: Well, we're now planning to introduce a tool. This tool is equivalent to being mainly based on ChatGPT, that is, developing a tool based on ChatGPT to assist auditors in conducting audits. If you had such a tool, what do you think would be the biggest purpose for using it, as well as the future you envision?
> P4: The purpose is definitely some of his safety tips. Just like when you do something, if you directly feed him a project, he can give you some safety effectiveness issues related to the project. An assistant, because ChatGPT is like you can feed him something, and if the things you feed him are only in one domain, it will definitely be more accurate. Because during human moderation, you actually won't think of every vulnerability. If the database he stores is large enough, and he can give a reminder when you encounter a similar problem, this is actually the most preferred, because humans may forget, but what is stored in the database won't. If he can give a reminder when there are issues like something to be fixed, it can better ensure the efficiency and effectiveness of one's own audit during the audit process.
> Interviewer: It's about efficiency and effectiveness, which is equivalent to these two points.

4:12 ¶ 32 – 37 in P4.docx

Codes:

○ Expectation: novice training tool: how ChatGPT can support novice as training and guiding tool

Content:

Interviewer: It's about writing some specific rules for fuzz testing and formal verification. Currently, there are two overall approaches using ChatGPT. One is the Knowledge Base approach, and the other is pure GPT. ChatGPT has been trained with sufficient knowledge over the past few years. As long as I have an effective prompt to ask, for example, if I ask 10,000 times, I can definitely collect all the results.
P4: You're like some beginners. Beginners surely need to get started, and when getting started, they may have no ideas for practical courses. Your GPT can provide them with some ideas. If you're cultivating someone, or if they're just beginners who want to do auditing but don't know some key points, ChatGPT can provide them. Regarding the auxiliary auditing just mentioned, the auxiliary tools you've developed for beginners will surely have an impact. Also, for development, it actually requires some security foundation, and your tool can provide support for development. You don't have to target only security personnel; you can also target developers. Developers can provide feedback on some issues and what problems have been fixed.
Interviewer: Does it mean that, for example, during the entire process of your auditing, GPT monitors your actions or mental activities in real time, and when it detects that you're stuck, it gives you a prompt?
P4: Almost now there's [unclear voice] that can help you write code, which can assist you and provide prompts while you're writing.
Interviewer: Copilot, right? Copilot is more of a result provider; it gives you more results and no reasons. If it could also tell you the reasons, it would actually be a great teaching tool. But I still can't quite understand how the beginner's tutorial works. It doesn't know to what extent you understand things.
P4: Right, he doesn't need to worry about how far you go; he just needs to provide the current standard. For example, when many people are developing, they often copy some courses and then modify some code. If your modifications are incorrect, he can point out the problems. What I envision more is that for beginners writing their own code, when they encounter some logical issues or language problems, he can prompt you that there is a problem. Also, on this platform, many people participate, and you can collect all the issues they write about.

5:3 ¶ 5 – 6 in P5.docx

Codes:
○ Expectation: assisted-content: provide interesting and unexpected points ○ Expectation: expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

Content:
Interviewer: Is that information related to “information”?
P5: Right, it's mainly about optimizing some cases or providing suggestions on code style. If we use ChatGPT later, some prompts can assist in doing this. On the one hand, it can output audit results, and on the other hand, it can provide some interesting points that we might not think of, which may open up some new ideas. However, its false positive rate is still quite high, but some of them can inspire ideas and provide assistance. Additionally, it can output in a certain format. For example, if you describe something to it, it can output the relevant description or format of the code, and you can directly use it as an assistant to paste some code into the report. Generally, it mainly helps with output format and content.

5:11 ¶ 65 – 66 in P5.docx

Codes:
○ Expectation: hope AI can help analyse automatically

Content:
Interviewer: We plan to develop a tool to assist auditors in better interacting with ChatGPT, and through this improved interaction, help auditors better identify those vulnerabilities. So, do you have any ideas about this tool or expectations for its functionality?
P5: I think this is really good, equivalent to creating a customized GPT specifically for auditors. Actually, considering from my own experience, regarding the coherence we just mentioned, it's like you can input an entire project, and it will automatically output everything, whether it's the project background, the overall execution process, or the specific audit results, right? There will be a general overview, and then you can ask specific questions, which is quite good.

5:12 ¶ 67 – 68 in P5.docx

Codes:

○ Expectation: hope AI can help decompose

Content:

> Interviewer: First, let it give you a general framework understanding, and then proceed to local understanding.
> P5: But actually, in my understanding, this process is more like a kind of Modularization analysis, or for example, although a project is very large and cannot be analyzed all at once, it can be split into various small files, and then finally we combine the conclusions of these small files, which I think might be achievable.

5:14 ¶ 70 in P5.docx

Codes:

○ Expectation: providing interactivity and if the questions are asked crossly, it should not be mixed

Content:

> This is still quite helpful for auditors. Moreover, I think there's a rather crucial aspect here, which is the issue of interactivity, that is, it can resolve a previous concern of mine. If questions about a and b are asked in a cross manner, it may not be very clear. With this interactive approach, for example, if you ask in detail about each function of contract a and each function of contract b, there won't be any cross - referencing, and it won't mix up the answers when reading them.

5:15 ¶ 72 – 76 in P5.docx

Codes:

○ Expectation: ask GPT based on the comments

Content:

> P5: Actually, there's another form that I think is also quite good, which is a VSCode plugin. It's like adding comments for you and so on.
> Interviewer: Provide explanations, right? Provide explanations for different Code Blocks.
> P5: Right, it's like adding comments. You can ask questions based on the comments, and this kind of approach might be more convenient.
> Interviewer: What kind of tool, specifically what kind of effect it achieves, would make you really want to use it?
> P5: In its final form, it will directly generate a report for you. For the intermediate form, it is capable of, for example, answering detailed questions. After expanding this, it can accurately describe all details, and when asked questions related to vulnerabilities, it can also provide relatively accurate descriptions.

5:16 ¶ 76 – 78 in P5.docx

Codes:

○ Expectation: hope the tool can directly generate some reports

Content:

> P5: In its final form, it will directly generate a report for you. For the intermediate form, it is capable of, for example, answering detailed questions. After expanding this, it can accurately describe all details, and when asked questions related to vulnerabilities, it can also provide relatively accurate descriptions.
> Interviewer: So you think accuracy is one of your priorities.
> P5: Right, as long as the false alarm rate is not too high, it's okay. But false alarms are definitely allowed. However, if the false alarm rate is relatively high, it may still take too much time to address those false alarms.

6:8 ¶ 23 – 27 in P6.docx

Codes:

○ Expectation: need tool assistance on quickly understand the logic; especially for different lanagauge

Content:

> Interviewer: So when the auditor audits your project, sometimes they actually haven't studied this language either, right?
> P6: Right. Another question is what? Actually, our company rarely uses this language; it's just that occasionally a project pops up that requires using this language. So, if I were to learn this language, would it take up too much energy? Right. So, the requirement from the auditor is that I won't spend too much time learning this language, but I need a tool to help me quickly understand the logic in smart contracts when I don't know this language or am not very proficient in its grammar. Because I just want to see if there are any such vulnerabilities I've seen in its logic. Right, this is the benefit that ChatGPT brings. For example, we may have corresponding ones here. Let me look and see if we can search by language. Sometimes we want to find this vulnerability, a bug at the non-language level, that is, at the logical level, and we also want to quickly understand this code. For example, the same piece of code can be written in Study, Python, or Rast. Maybe this code is written in three different languages, and they are language-independent, but they will have the same type of vulnerability. However, if I don't understand Rast, it may take four or five lines to write in Python, but it may take five or six lines, seven or eight lines, or even ten lines in Rast. It will be very difficult and challenging for you to analyze.
> Yes, another point I'd like to make is that so many languages bring challenges to auditors, and as I just mentioned, many supply chains only stay popular for half a year or a year. During this period, they may develop many intelligent science projects based on the supply chain, but after half a year or a year, the popularity of the supply chain drops sharply, and basically no developers are working on it anymore.
> Interviewer: So it's equivalent to reducing your input, cutting down on unnecessary input, right?
> P6: Right, it has reduced a great deal. In fact, it has reduced a great deal of unnecessary investment in this area, because for human moderators, learning a lot of language related to the supply chain is actually a very headache-inducing thing.

○ Expectation: ask GPT based on the comments

1 Quotations:

5:15 ¶ 72 – 76 in P5.docx

Codes:

○ Expectation: ask GPT based on the comments

Content:

> P5: Actually, there's another form that I think is also quite good, which is a VSCode plugin. It's like adding comments for you and so on.
> Interviewer: Provide explanations, right? Provide explanations for different Code Blocks.
> P5: Right, it's like adding comments. You can ask questions based on the comments, and this kind of approach might be more convenient.
> Interviewer: What kind of tool, specifically what kind of effect it achieves, would make you really want to use it?
> P5: In its final form, it will directly generate a report for you. For the intermediate form, it is capable of, for example, answering detailed questions. After expanding this, it can accurately describe all details, and when asked questions related to vulnerabilities, it can also provide relatively accurate descriptions.

○ Expectation: assisted-content: provide interesting and unexpected points

1 Quotations:

5:3 ¶ 5 – 6 in P5.docx

Codes:

○ Expectation: assisted-content: provide interesting and unexpected points ○ Expectation: expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

Content:

> Interviewer: Is that information related to “information”?

> P5: Right, it's mainly about optimizing some cases or providing suggestions on code style. If we use ChatGPT later, some prompts can assist in doing this. On the one hand, it can output audit results, and on the other hand, it can provide some interesting points that we might not think of, which may open up some new ideas. However, its false positive rate is still quite high, but some of them can inspire ideas and provide assistance. Additionally, it can output in a certain format. For example, if you describe something to it, it can output the relevant description or format of the code, and you can directly use it as an assistant to paste some code into the report. Generally, it mainly helps with output format and content.

○ Expectation: data flowchart: 1. can we use data flowchart to support the chain? 2. can we use ata flowchartto support more context input?

1 Quotations:

2:18 ¶ 77 in P2.docx

Codes:

○ Expectation: data flowchart: 1. can we use data flowchart to support the chain? 2. can we use ata flowchartto support more context input?

Content:

> Without providing any information to GPT, if we directly ask GPT to find a vulnerability, it is actually very difficult for it to discover these vulnerabilities through a highly reproducible process. It may discover them, but this process may not form a methodology, making it difficult to reproduce. However, I think that if, for example, my static engine or some other tools can provide such a trajectory, a main Line of Business or multiple parallel sub-Lines of Business, and can replace our current approach from function to scenario and from scenario to vulnerability in this way, I think it may be more effective. I think this is a very critical point because this key point actually solves our current biggest problem, which is the context problem. The biggest drawback of our current use of a single function is that it cannot cover the context, and it is difficult to obtain the context. I think if this can be provided through some other means.

○ Expectation: data flowchart: finding links among nodes (variables, functions) and use it to support ChatGPT understanding

1 Quotations:

2:19 ¶ 78 – 82 in P2.docx

Codes:

○ Expectation: data flowchart: finding links among nodes (variables, functions) and use it to support ChatGPT understanding

Content:

> Interviewer: You mean going to draw that diagram, right?
> P2: Right, I might, for example, it might be a point like this, then go to the next point, and then to the next point, you should be able to see it. This is another process, perhaps another process has reached this point. To put it simply, we actually want to extract different paths and different main business logics from the code. Then I'm thinking about this business logic line, but this business route is actually not very good. First of all, it cannot cover the business; it's just an incomprehensible fixed pattern. I think maybe we can use ChatGPT to extract this business-related process. The business process should actually be in our database, in our Knowledge Base, and there should be corresponding rules during the reasoning process.
> P2: Just now when we were talking about business processes, I thought about it. This business process can actually be, I'm not sure if it can be regarded as a finite-state machine. For example, in a finite-state machine, it starts with a state, then you call a function, and it will enter the next state. How each state changes depends on the modifications made to those global variables. I think one thing that may involve human-machine interaction is that what has been presented like this in the past is actually relatively critical information.
> Interviewer: How to make this information more effective?
> P2: More effectively demonstrate it, whether using the text results returned by ChatGPT and adding some analysis of visual graphs, viewing visual graphs, or using other solutions to show the main

business starting points, endpoints, and processes within the current contract, as well as the relationships between different business flows. I think this may be a relatively critical aspect in the audit process. Of course, this is also helpful for auditors. How auditors ultimately use this and how different auditors use it. I think that the creation and display of a process like the one mentioned earlier actually have an impact and are helpful for all three steps of planning, reasoning, and validating. Perhaps in the end, for example, the location where vulnerabilities occur may be in those relatively long processes or in those with more relationships with other processes. Maybe those with only one or two nodes and only one or two business flows may have a lower probability of generating vulnerabilities, which may be useful for planning.

○ Expectation: data flowchart: support understanding using data flows like business flowchart

1 Quotations:

2:14 ¶ 65 – 69 in P2.docx

Codes:

○ Expectation: data flowchart: support understanding using data flows like business flowchart

Content:

P2: Saying there are a dozen is an exaggeration, but there has been an increase. I've thought about some of the issues it pointed out. For some of the bugs it identified, the risk level should be relatively low. It tends to find more bugs with low risk levels or at the Information level, but for truly high-risk vulnerabilities, GPT can point out relatively few. While the quantity has increased, in terms of quality, the output of high-quality vulnerabilities is still relatively low.
Interviewer: You think so. We plan to design and develop an interface, software, or platform to help users better interact with GPT, to assist your work. Which stage do you think you most need help with? And what functions do you most expect?
P2: I hope it can provide more information in analyzing the overall project structure, especially if it can generate some business flowcharts or function call graphs.
Interviewer: It means all the business processes related to the list you just mentioned, anyway, just to help you understand.
P2: Right, it's about reverse-engineering from the code to the entire business process, understanding what kind of business it is, which functions users can call, and what role they play in the overall business. Because after we obtain the code and then look at its official documentation, the website's promotion, or its white paper, there are actually discrepancies, or rather, it's still not easy to understand. What's written on its official website sounds great, but in terms of its code implementation, it's hard to understand and difficult to correlate with what's on the official website. However, if you use GPT, it will explain based on the code what it's doing and what its business process is like.

○ Expectation: domain knowledge augmented: hope can augment GPT some domain knowledge for high efficiency and effectiveness

1 Quotations:

4:11 ¶ 24 – 26 in P4.docx

Codes:

○ Expectation: domain knowledge augmented: hope can augment GPT some domain knowledge for high efficiency and effectiveness

Content:

Interviewer: Well, we're now planning to introduce a tool. This tool is equivalent to being mainly based on ChatGPT, that is, developing a tool based on ChatGPT to assist auditors in conducting audits. If you had such a tool, what do you think would be the biggest purpose for using it, as well as the future you envision?
P4: The purpose is definitely some of his safety tips. Just like when you do something, if you directly feed him a project, he can give you some safety effectiveness issues related to the project. An assistant, because ChatGPT is like you can feed him something, and if the things you feed him are only in one domain, it will definitely be more accurate. Because during human moderation, you

actually won't think of every vulnerability. If the database he stores is large enough, and he can give a reminder when you encounter a similar problem, this is actually the most preferred, because humans may forget, but what is stored in the database won't. If he can give a reminder when there are issues like something to be fixed, it can better ensure the efficiency and effectiveness of one's own audit during the audit process.
Interviewer: It's about efficiency and effectiveness, which is equivalent to these two points.

○ Expectation: domain knowledge augmented: opportunity: how new auditor do code auditing-solution: one GPT to ask another GPT

1 Quotations:

1:14 ¶ 33 – 34 in P1.docx

Codes:

○ Expectation: domain knowledge augmented: opportunity: how new auditor do code auditing-solution: one GPT to ask another GPT

Content:

Interviewer: It's equivalent to how you transfer your knowledge to a newbie, for example, how to let a newbie use the Knowledge Base of an export.
P1: Right, so this kind of requirement is relatively difficult. I haven't come up with a good solution yet. The key point is to assume that GPT has absolutely no knowledge of these things and let it find these loopholes from scratch. There are roughly two methods here. One is to feed the results of ChatGPT back to ChatGPT. For example, I first ask what this is for, and then ask, based on this judgment, where do you think the loopholes might occur? For an auditor, they might think the loopholes could occur in the if statement. So what do you think might be missing from the judgment? I would guide it step by step to think like a human. Do you understand what I mean? Ok, this is the first solution.

○ Expectation: expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

1 Quotations:

5:3 ¶ 5 – 6 in P5.docx

Codes:

○ Expectation: assisted-content: provide interesting and unexpected points ○ Expectation: expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

Content:

Interviewer: Is that information related to “information”?
P5: Right, it's mainly about optimizing some cases or providing suggestions on code style. If we use ChatGPT later, some prompts can assist in doing this. On the one hand, it can output audit results, and on the other hand, it can provide some interesting points that we might not think of, which may open up some new ideas. However, its false positive rate is still quite high, but some of them can inspire ideas and provide assistance. Additionally, it can output in a certain format. For example, if you describe something to it, it can output the relevant description or format of the code, and you can directly use it as an assistant to paste some code into the report. Generally, it mainly helps with output format and content.

○ Expectation: future thoughts: one-click generation of vulnerability report with high accuracy

1 Quotations:

2:15 ¶ 70 – 71 in P2.docx

Codes:

○ Expectation: future thoughts: one-click generation of vulnerability report with high accuracy

Content:

> Interviewer: During the process of assisting you in finding errors, what features do you most expect to have to better enable you to use GPT?
> P2: What I most hope for is, of course, one-click generation of vulnerability reports. It should identify vulnerabilities and then generate vulnerability reports with just one click, just like a vulnerability scanner. This way, I don't have to check if there are any issues here and probe step by step. Just like a vulnerability scanner, it knows what it needs to do, then identifies, analyzes, and outputs the results in the form of a report.

○ Expectation: hope AI can help analyse automatically

1 Quotations:

5:11 ¶ 65 – 66 in P5.docx

Codes:

○ Expectation: hope AI can help analyse automatically

Content:

> Interviewer: We plan to develop a tool to assist auditors in better interacting with ChatGPT, and through this improved interaction, help auditors better identify those vulnerabilities. So, do you have any ideas about this tool or expectations for its functionality?
> P5: I think this is really good, equivalent to creating a customized GPT specifically for auditors. Actually, considering from my own experience, regarding the coherence we just mentioned, it's like you can input an entire project, and it will automatically output everything, whether it's the project background, the overall execution process, or the specific audit results, right? There will be a general overview, and then you can ask specific questions, which is quite good.

○ Expectation: hope AI can help decompose

1 Quotations:

5:12 ¶ 67 – 68 in P5.docx

Codes:

○ Expectation: hope AI can help decompose

Content:

> Interviewer: First, let it give you a general framework understanding, and then proceed to local understanding.
> P5: But actually, in my understanding, this process is more like a kind of Modularization analysis, or for example, although a project is very large and cannot be analyzed all at once, it can be split into various small files, and then finally we combine the conclusions of these small files, which I think might be achievable.

○ Expectation: hope the tool can directly generate some reports

1 Quotations:

5:16 ¶ 76 – 78 in P5.docx

Codes:

○ Expectation: hope the tool can directly generate some reports

Content:

> P5: In its final form, it will directly generate a report for you. For the intermediate form, it is capable of, for example, answering detailed questions. After expanding this, it can accurately describe all details, and when asked questions related to vulnerabilities, it can also provide relatively accurate descriptions.
> Interviewer: So you think accuracy is one of your priorities.

> P5: Right, as long as the false alarm rate is not too high, it's okay. But false alarms are definitely allowed. However, if the false alarm rate is relatively high, it may still take too much time to address those false alarms.

○ Expectation: need tool assistance on quickly understand the logic; especially for different lanagauge

1 Quotations:

6:8 ¶ 23 – 27 in P6.docx

Codes:

○ Expectation: need tool assistance on quickly understand the logic; especially for different lanagauge

Content:

> Interviewer: So when the auditor audits your project, sometimes they actually haven't studied this language either, right?
> P6: Right. Another question is what? Actually, our company rarely uses this language; it's just that occasionally a project pops up that requires using this language. So, if I were to learn this language, would it take up too much energy? Right. So, the requirement from the auditor is that I won't spend too much time learning this language, but I need a tool to help me quickly understand the logic in smart contracts when I don't know this language or am not very proficient in its grammar. Because I just want to see if there are any such vulnerabilities I've seen in its logic. Right, this is the benefit that ChatGPT brings. For example, we may have corresponding ones here. Let me look and see if we can search by language. Sometimes we want to find this vulnerability, a bug at the non-language level, that is, at the logical level, and we also want to quickly understand this code. For example, the same piece of code can be written in Study, Python, or Rast. Maybe this code is written in three different languages, and they are language-independent, but they will have the same type of vulnerability. However, if I don't understand Rast, it may take four or five lines to write in Python, but it may take five or six lines, seven or eight lines, or even ten lines in Rast. It will be very difficult and challenging for you to analyze.
> Yes, another point I'd like to make is that so many languages bring challenges to auditors, and as I just mentioned, many supply chains only stay popular for half a year or a year. During this period, they may develop many intelligent science projects based on the supply chain, but after half a year or a year, the popularity of the supply chain drops sharply, and basically no developers are working on it anymore.
> Interviewer: So it's equivalent to reducing your input, cutting down on unnecessary input, right?
> P6: Right, it has reduced a great deal. In fact, it has reduced a great deal of unnecessary investment in this area, because for human moderators, learning a lot of language related to the supply chain is actually a very headache-inducing thing.

○ Expectation: novice training tool: how ChatGPT can support novice as training and guiding tool

1 Quotations:

4:12 ¶ 32 – 37 in P4.docx

Codes:

○ Expectation: novice training tool: how ChatGPT can support novice as training and guiding tool

Content:

> Interviewer: It's about writing some specific rules for fuzz testing and formal verification. Currently, there are two overall approaches using ChatGPT. One is the Knowledge Base approach, and the other is pure GPT. ChatGPT has been trained with sufficient knowledge over the past few years. As long as I have an effective prompt to ask, for example, if I ask 10,000 times, I can definitely collect all the results.
> P4: You're like some beginners. Beginners surely need to get started, and when getting started, they may have no ideas for practical courses. Your GPT can provide them with some ideas. If you're cultivating someone, or if they're just beginners who want to do auditing but don't know some key points, ChatGPT can provide them. Regarding the auxiliary auditing just mentioned, the

> auxiliary tools you've developed for beginners will surely have an impact. Also, for development, it actually requires some security foundation, and your tool can provide support for development. You don't have to target only security personnel; you can also target developers. Developers can provide feedback on some issues and what problems have been fixed.
> Interviewer: Does it mean that, for example, during the entire process of your auditing, GPT monitors your actions or mental activities in real time, and when it detects that you're stuck, it gives you a prompt?
> P4: Almost now there's [unclear voice] that can help you write code, which can assist you and provide prompts while you're writing.
> Interviewer: Copilot, right? Copilot is more of a result provider; it gives you more results and no reasons. If it could also tell you the reasons, it would actually be a great teaching tool. But I still can't quite understand how the beginner's tutorial works. It doesn't know to what extent you understand things.
> P4: Right, he doesn't need to worry about how far you go; he just needs to provide the current standard. For example, when many people are developing, they often copy some courses and then modify some code. If your modifications are incorrect, he can point out the problems. What I envision more is that for beginners writing their own code, when they encounter some logical issues or language problems, he can prompt you that there is a problem. Also, on this platform, many people participate, and you can collect all the issues they write about.

○ Expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

1 Quotations:

1:33 ¶ 81 – 82 in P1.docx

Codes:

○ Expectation: pre-defined prompts: pre-defined buttons that are relevant to the task prompt

Content:

> Interviewer: Okay, and now I'd like to ask the last question. If we were to design a new interface to help you interact with ChatGPT, what expectations do you have and what features would you like it to have?
> P1: Actually, I understand this part. I don't know if you've looked at the Audit Wizard. For auditors, I think there may be two different levels of requirements. The first, with relatively high expectations, is something like the Audit Wizard, which not only can ask questions about code but also can integrate with some different tools or other things. This is perhaps what we ultimately hope for. Of course, the Audit Wizard itself is not particularly good; it doesn't make much use of AI. In contrast, the other requirement is that we may not import corresponding code but simply copy a contract file in. If that's the case, expectations may be higher. It would be best if there were some very effective trick prompts available, like the prompt I used before. If there were such prompts, I think it would be better. Because, to be honest, the completeness of the prompt greatly affects GPT's response results, which means that the better the prompt, the harder it is to obtain. The prompt I have here is one that I adjusted after a long time. If there were a collection of such prompts, I think it would be better.

○ Expectation: providing interactivity and if the questions are asked crossly, it should not be mixed

1 Quotations:

5:14 ¶ 70 in P5.docx

Codes:

○ Expectation: providing interactivity and if the questions are asked crossly, it should not be mixed

Content:

> This is still quite helpful for auditors. Moreover, I think there's a rather crucial aspect here, which is the issue of interactivity, that is, it can resolve a previous concern of mine. If questions about a and b are asked in a cross manner, it may not be very clear. With this interactive approach, for example,

> if you ask in detail about each function of contract a and each function of contract b, there won't be any cross - referencing, and it won't mix up the answers when reading them.

○ expert behavior

11 Quotations:

1:4 ¶ 10 in P1.docx

Codes:

○ expert behavior: personal written workflow

Content:

> P1: Not entirely. This was written by myself, but other companies may have differences, but basically it's not too far off. This is an audit workflow. For example, the read me or white paper mentioned earlier is a written description of what this project is about. After obtaining it, I will then proceed to conduct an audit, specifically a general audit. First, I will roughly identify the overall logic. Based on the proposal it mentions, this logic is the main logic, so I will first look at its specific implementation.

1:11 ¶ 22 – 28 in P1.docx

Codes:

○ expert behavior: using key concept/functionality structure ○ Reasoning: ChatGPT: generate function description, and do matching using key concept

Content:

> Interviewer: Let me ask again. Regarding your Knowledge Base, as you can see, it's listed from top to bottom, and it seems to lack a structure. When you perform matching, you can only rely on your own memory and experience to do so.
> P1: The matching process is roughly like this, that is, there are two ways to match. One is to match based on the code content. So, just now this was only a type list. In fact, in this library, since this library might be quite large, we will match based on the content code of a function, the data function code. For example, if we input this function code and then convert it into a functionality through GPT, this refers to the functional description of a function. For instance, let me give an example. Suppose we randomly select one, and it is a description of a function. Then we match the functional description I input with the functional descriptions in this list, and choose the corresponding vulnerability of the one that is most similar, because what are these data? These are vulnerability data. Each function has a possible vulnerability below it. So, after the matching, then ChatGPT, and generally when I add this, I usually use the key concept to ask questions. This is the functional description, this is the vulnerability description, the vulnerability description of the key concept, and I will ask questions in this way. As for the second category or this, this is the corresponding classification point I mentioned just now. What is it for? This is what you just said, it is to draw inferences from one instance. I will probably do it like this. What if it cannot be found? I will look for its parent class.
> Interviewer: Is it based on content? For example, is the key concept a simplification of functionality?
> P1: You can understand it this way: functionality is the scenario in which I use this knowledge. Key concept is this piece of knowledge, and we humans have defined a basic human behavior here, which is the behavior of using this knowledge in this scenario.
> P1: Right, that is to say, first I will match through functionality. I will construct all these functionalities, about 960 to 1000 of them, into a database. Then, by inputting a functionality, I will match the most similar functionality, find the corresponding key concept, and then ask ChatGPT. That is to say, what I actually match in the end is a key concept. That is, input a piece of code, and then get the most suitable key concept for it. Finally, I will ask questions about the key concept and the code, asking whether it is possible for such a vulnerability to occur in this code.
> Interviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to

occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

1:20 ¶ 46 – 49 in P1.docx

Codes:

○ expert behavior: purpose: best to find more

Content:

Interviewer: This false alarm means either it reports an error or it underreports.
P1: Right, yes, of course, it usually results in false reporting. Underreporting means omission, and in my opinion, when there is omission, if we check the checklist, I feel that the Knowledge Base is not large enough.
Interviewer: Since you're doing this, there must be several goals. One goal is to accurately identify vulnerabilities, and then you need to report the identified ones to, I can understand it as the owner of the project itself. Another thing is that if you can't find all of them, it actually has no impact on you, right? The more you find, the better, but it's okay if you don't find them all.
P1: Failure to identify all potential issues may have an impact on reputation. Yes, that's roughly it. The number of logical vulnerabilities, or such vulnerabilities in general, is unlimited, meaning it will never be completely secure, and there will always be some undetected problems. Of course, it's not the case for all code; the simpler the code, the fewer such issues there are.

1:28 ¶ 66 – 67 in P1.docx

Codes:

○ expert behavior: interaction: identify the correctness of GPT response along the generation process

Content:

Interviewer: Do you usually wait until it has fully generated before looking at its response?
P1: Not entirely. For example, it provides so many complex things here. I will focus on the specific content of some of the code it provides. For instance, one of the most critical aspects is that it is related to the domain knowledge of my smart contract, and here there is a term called "negative number". In the domain knowledge of smart contracts, the term "negative number" rarely appears because smart contracts can only handle positive numbers. So, once a negative number appears, I need to pay attention. Sometimes ChatGPT will produce something where it doesn't seem to know that negative numbers can exist in smart contracts. That's why, as you can see, I specifically mentioned in the prompt that all numbers in the code are positive, all are positive, all numbers are positive. But apparently, GPT doesn't seem to have fully considered this. So I need to look back and see if the so-called liquid paper change could be a negative number. Looking back at the original code, it is int256, not uint256. The so-called positive number actually means there must be a "u" in front, uint256. Here it is int256, indicating that its definition itself can be negative.

1:29 ¶ 68 – 71 in P1.docx

Codes:

○ expert behavior: interaction: combate with GPT can enhance the correctness with ChatGPT

Content:

Ok, I roughly understand that it's possible for this to be a negative number, so ok, ChatGPT doesn't seem to be wrong here. It says there's indeed no problem, and it could indeed be a complex number. Then, after it becomes liquid paper, it only has some calculations, and perhaps it's judged to be a vulnerability or something. I'll still look into it. Another point is that if I don't want to think too much, I'll simply tell it that I don't think it's a vulnerability and ask it to take a look. To put it bluntly, I'm just following the deceptive train of thought in the prompt just now, getting it to refute my opinion. If it can refute successfully, it might indicate that it's probably a vulnerability; if it agrees with my view, it might mean it's not a vulnerability.
In my opinion, ChatGPT does take into account opposing views in this regard. You can see that in this section, it acknowledges its own error in the analysis, but upon reexamining the code, it notes that it has already considered the negative number scenario. So, we might look back at this part above to see where it was considered. Just now, what I read was this section, where it should consider the situation if it is less than 0. Its conclusion is okay. If we put it this way, I would usually

skip this loophole because if I go through it quickly, but if I take my time, I will carefully read to see if it could potentially be a problem.
Interviewer: So in this case, it is more likely that there is a problem with its judgment.
P1: I feel that it's probably around a 73% proportion. If I reply with this statement and it says it's not a vulnerability, I feel that there's a 70% chance it should not be one, but there's a 30% possibility that it might still be a vulnerability. It's just that I haven't looked into it carefully, so I think I should continue to look into those.

1:30 ¶ 72 – 74 in P1.docx

Codes:

○ expert behavior: interaction: 1) excuation for a few times, it will narrow. 2) reuse prompts 3)validation is hard and running multiple times could find all bugs

Content:

Interviewer: This time, your interaction with ChatGPT, such as your debate with it, that is, you refuting its viewpoints. For example, can these interactions be repurposed for other code when you interact with it next time?
P1: Right, there's another type that might not be particularly easy to repurpose into other prompts. For example, after I just read this, it mentions negative numbers, but it seems that negative numbers have already been considered here. At this point, I would ask about it, but it seems that negative numbers less than 0 have already been considered. How to explain this? I would still ask it like this. Sometimes in such cases, it actually admits its mistakes, but sometimes it also refutes me. Look at its second sentence, yet the potential issue may no longer be whether it considers handling negative numbers.
I get really annoyed when there are so many, and actually sometimes what bothers me the most is precisely these things, that is, I have to verify whether it is actually a vulnerability. This is actually a rather painful aspect for me, because in my view, this prompt, after it has been executed a sufficient number of times, is capable of finding all the vulnerabilities within this function. That is to say, it has an advantage, which I feel is beneficial, that is, this prompt can be somewhat convergent or the vulnerability can be slightly convergent. ChatGPT's vulnerability in answering questions can be convergent. So sometimes I basically keep going back and forth to conduct secondary questioning. In fact, in the first few times of secondary questioning, it will probably output some different vulnerabilities, but in the end, it will increasingly resemble or be exactly the same as the previous vulnerabilities.

1:31 ¶ 75 in P1.docx

Codes:

○ expert behavior: validating: vulnerablities could be narrowed down after a few times try

Content:

Right, so sometimes after I generally find this thing, I won't ask about it again. Because for the person submitting the vulnerability, it falls into two scenarios. One scenario is the internal audit of the project company, which must ensure that the report you provide to the project party is correct, without false positives. However, bug bounty or vulnerability bounty, or audit competition, usually doesn't consider whether the vulnerability you provided has false positives. There is someone on that side to help you check. There is a specific position, an identity called warden, which is used to check whether the information of the vulnerabilities provided by all participants is a real vulnerability. There will be such a check, and this check actually saves us effort. That is to say, after I generate 100 vulnerabilities, for example, after completing this task, I will write the vulnerability into an English description, including English translation, including description, recommendation, title, and maybe add a severity. Generally, we are required to submit these 4 items, and it will generate a relatively complete vulnerability report. At this time, I can directly copy and paste it onto their platform.

1:32 ¶ 76 – 77 in P1.docx

Codes:

○ expert behavior: writing reports: can write in a way that it is important with GPT help

Content:

> P1: Naturally, someone will check. I don't want to deal with this, so this is also a place where we've always wanted to take a bit of a shortcut. Well, there might be another aspect here, that is, sometimes we'll modify this part, because there's also an unwritten rule here. If you describe a vulnerability using method a, it might be a low-dimensional one, but if you describe it using method b, it might be a high-dimensional one. This has a significant impact on us, especially on our bounty, so I'll add a sentence later to describe this vulnerability as extremely serious. Right, there are indeed actual cases, there really are, but not many, there are truly actual cases. It will give a different description, a very serious one. Right, there are probably some shortcuts taken in there. I can give an example, like some of the audits I did before. You can take a direct look, you should be able to see this too. This is a report on the number of vulnerabilities I submitted before.
> For example, after clicking on this, there is a "my submission", where you can see that in this submission, I submitted 3 critical issues, 1 media issue, and a total of 9 issues. However, only 5 were approved, identified, and confirmed as vulnerabilities, which means 4 were not vulnerabilities and would be rejected by Secure Three. You can see why they were rejected. For example, "Merge the same issue for out of scope" or "Please give us more detail about the specific danger also".

1:34 ¶ 83 – 89 in P1.docx

Codes:

○ expert behavior: ChatGPT improved human performance (decrease time and labor costs) a lot

Content:

> Interviewer: Do you think you used other traditional software a lot in the past, or do you prefer to rely on your own checklist?
> P1: In fact, we all have such scanning requirements, but such requirements actually have some drawbacks. The reason for saying there are drawbacks now is that there are some things it can't scan out. To put it simply, these things are all static engines or purely program analysis tools, and their drawback is that they are unable to recognize business logic and unable to understand business logic.
> Interviewer: How much do you think GPT has improved your efficiency approximately?
> P1: For me, it may have improved significantly. I basically don't think at all when auditing now. Previously, it took 100 units of time, and now it might be only 1/10, or about 1/10. To put it simply, just look at the things I asked just now. For example, in the past, after getting this code, I would carefully examine each one, and the time was uncertain. Manually, the efficiency was about 300 to 500 lines per day, with 8 hours a day. If it's ChatGPT, it might take 30 to 50 minutes at most, and this still includes the time for you to keep asking questions, verifying, and having conversations. If you completely ignore the output effect and just want to get the job done for the money, it can be completed in 3 to 5 minutes. It used to take 8 hours, or perhaps several hundred minutes, around four or five hundred minutes.
> Interviewer: How much do you think your performance has improved approximately?
> P1: I think there's at least a doubling of performance improvement, because there are some things that I really can't tell if I look at them myself, at least a doubling is possible.
> Interviewer: Okay, that should be all the questions.

2:17 ¶ 76 in P2.docx

Codes:

○ expert behavior: functions=>scenario=> rules=> vulnerability: this chain is not always working

Content:

> Additionally, I'm wondering what the business logic in a smart contract should be. Is it multiple entry points that then extend into different lines, with these different lines intersecting each other, but each entry point having a main line, which actually represents the so-called business logic of a certain business, such as an auction list? That is to say, within a contract, there may be many parallel businesses, and there may be intersections between these parallel businesses. These intersections may be function intersections, and they may also be intersections of state variables. I'm not sure if this description of the business logic of a contract or a business logic flowchart is accurate. I'm wondering if it's difficult for GPT to detect errors in such intersecting business processes.

5:7 ¶ 43 – 48 in P5.docx

Codes:

○ expert behavior: get some thinking from other perspectives

Content:

> Interviewer: So you said that you would use it when you're in a hurry.
> P5: Also, you've basically completed the audit, and then see if it can provide some very different perspectives of thinking.
> Interviewer: I find this part quite interesting. Could you elaborate on how you used it to gain multi-angle thinking?
> P5: If you've already reviewed this project, for example, the reports are basically completed, and there's still some reserved time left, then you feed that information to it, and let it output different justifications. Then you take a look. Sometimes, although it may make mistakes, the scope of its thinking might broaden your perspective. Right? If you follow its imagination, you might make different discoveries, right?
> Interviewer: So after you finally generate the report, you not only want him to check whether the report is correct or see if he can help you improve it, but more importantly, you will also ask him if he has any other ideas, right? That's how you use it.
> P5: If the actual construction period is relatively tight, then he helps me output some things.

○ expert behavior: ChatGPT improved human performance (decrease time and labor costs) a lot

1 Quotations:

1:34 ¶ 83 – 89 in P1.docx

Codes:

○ expert behavior: ChatGPT improved human performance (decrease time and labor costs) a lot

Content:

> Interviewer: Do you think you used other traditional software a lot in the past, or do you prefer to rely on your own checklist?
> P1: In fact, we all have such scanning requirements, but such requirements actually have some drawbacks. The reason for saying there are drawbacks now is that there are some things it can't scan out. To put it simply, these things are all static engines or purely program analysis tools, and their drawback is that they are unable to recognize business logic and unable to understand business logic.
> Interviewer: How much do you think GPT has improved your efficiency approximately?
> P1: For me, it may have improved significantly. I basically don't think at all when auditing now. Previously, it took 100 units of time, and now it might be only 1/10, or about 1/10. To put it simply, just look at the things I asked just now. For example, in the past, after getting this code, I would carefully examine each one, and the time was uncertain. Manually, the efficiency was about 300 to 500 lines per day, with 8 hours a day. If it's ChatGPT, it might take 30 to 50 minutes at most, and this still includes the time for you to keep asking questions, verifying, and having conversations. If you completely ignore the output effect and just want to get the job done for the money, it can be completed in 3 to 5 minutes. It used to take 8 hours, or perhaps several hundred minutes, around four or five hundred minutes.
> Interviewer: How much do you think your performance has improved approximately?
> P1: I think there's at least a doubling of performance improvement, because there are some things that I really can't tell if I look at them myself, at least a doubling is possible.
> Interviewer: Okay, that should be all the questions.

○ expert behavior: context is limited (due to the data cannot be imported totally), especially for reasoning and validating

0 Quotations

○ expert behavior: functions=>scenario=> rules=> vulnerability: this chain is not always working

1 Quotations:

2:17 ¶ 76 in P2.docx

Codes:

○ expert behavior: functions=>scenario=> rules=> vulnerability: this chain is not always working

Content:

> Additionally, I'm wondering what the business logic in a smart contract should be. Is it multiple entry points that then extend into different lines, with these different lines intersecting each other, but each entry point having a main line, which actually represents the so-called business logic of a certain business, such as an auction list? That is to say, within a contract, there may be many parallel businesses, and there may be intersections between these parallel businesses. These intersections may be function intersections, and they may also be intersections of state variables. I'm not sure if this description of the business logic of a contract or a business logic flowchart is accurate. I'm wondering if it's difficult for GPT to detect errors in such intersecting business processes.

○ expert behavior: get some thinking from other perspectives

1 Quotations:

5:7 ¶ 43 – 48 in P5.docx

Codes:

○ expert behavior: get some thinking from other perspectives

Content:

> Interviewer: So you said that you would use it when you're in a hurry.
> P5: Also, you've basically completed the audit, and then see if it can provide some very different perspectives of thinking.
> Interviewer: I find this part quite interesting. Could you elaborate on how you used it to gain multi-angle thinking?
> P5: If you've already reviewed this project, for example, the reports are basically completed, and there's still some reserved time left, then you feed that information to it, and let it output different justifications. Then you take a look. Sometimes, although it may make mistakes, the scope of its thinking might broaden your perspective. Right? If you follow its imagination, you might make different discoveries, right?
> Interviewer: So after you finally generate the report, you not only want him to check whether the report is correct or see if he can help you improve it, but more importantly, you will also ask him if he has any other ideas, right? That's how you use it.
> P5: If the actual construction period is relatively tight, then he helps me output some things.

○ expert behavior: interaction: 1) excuation for a few times, it will narrow. 2) reuse prompts 3)validation is hard and running multiple times could find all bugs

1 Quotations:

1:30 ¶ 72 – 74 in P1.docx

Codes:

○ expert behavior: interaction: 1) excuation for a few times, it will narrow. 2) reuse prompts 3)validation is hard and running multiple times could find all bugs

Content:

> Interviewer: This time, your interaction with ChatGPT, such as your debate with it, that is, you refuting its viewpoints. For example, can these interactions be repurposed for other code when you interact with it next time?
> P1: Right, there's another type that might not be particularly easy to repurpose into other prompts. For example, after I just read this, it mentions negative numbers, but it seems that negative numbers have already been considered here. At this point, I would ask about it, but it seems that negative numbers less than 0 have already been considered. How to explain this? I would still ask it like this. Sometimes in such cases, it actually admits its mistakes, but sometimes it also refutes me.

> Look at its second sentence, yet the potential issue may no longer be whether it considers handling negative numbers.
> I get really annoyed when there are so many, and actually sometimes what bothers me the most is precisely these things, that is, I have to verify whether it is actually a vulnerability. This is actually a rather painful aspect for me, because in my view, this prompt, after it has been executed a sufficient number of times, is capable of finding all the vulnerabilities within this function. That is to say, it has an advantage, which I feel is beneficial, that is, this prompt can be somewhat convergent or the vulnerability can be slightly convergent. ChatGPT's vulnerability in answering questions can be convergent. So sometimes I basically keep going back and forth to conduct secondary questioning. In fact, in the first few times of secondary questioning, it will probably output some different vulnerabilities, but in the end, it will increasingly resemble or be exactly the same as the previous vulnerabilities.

○ expert behavior: interaction: combate with GPT can enhance the correctness with ChatGPT

1 Quotations:

1:29 ¶ 68 – 71 in P1.docx

Codes:

○ expert behavior: interaction: combate with GPT can enhance the correctness with ChatGPT

Content:

> Ok, I roughly understand that it's possible for this to be a negative number, so ok, ChatGPT doesn't seem to be wrong here. It says there's indeed no problem, and it could indeed be a complex number. Then, after it becomes liquid paper, it only has some calculations, and perhaps it's judged to be a vulnerability or something. I'll still look into it. Another point is that if I don't want to think too much, I'll simply tell it that I don't think it's a vulnerability and ask it to take a look. To put it bluntly, I'm just following the deceptive train of thought in the prompt just now, getting it to refute my opinion. If it can refute successfully, it might indicate that it's probably a vulnerability; if it agrees with my view, it might mean it's not a vulnerability.
> In my opinion, ChatGPT does take into account opposing views in this regard. You can see that in this section, it acknowledges its own error in the analysis, but upon reexamining the code, it notes that it has already considered the negative number scenario. So, we might look back at this part above to see where it was considered. Just now, what I read was this section, where it should consider the situation if it is less than 0. Its conclusion is okay. If we put it this way, I would usually skip this loophole because if I go through it quickly, but if I take my time, I will carefully read to see if it could potentially be a problem.
> Interviewer: So in this case, it is more likely that there is a problem with its judgment.
> P1: I feel that it's probably around a 73% proportion. If I reply with this statement and it says it's not a vulnerability, I feel that there's a 70% chance it should not be one, but there's a 30% possibility that it might still be a vulnerability. It's just that I haven't looked into it carefully, so I think I should continue to look into those.

○ expert behavior: interaction: identify the correctness of GPT response along the generation process

1 Quotations:

1:28 ¶ 66 – 67 in P1.docx

Codes:

○ expert behavior: interaction: identify the correctness of GPT response along the generation process

Content:

> Interviewer: Do you usually wait until it has fully generated before looking at its response?
> P1: Not entirely. For example, it provides so many complex things here. I will focus on the specific content of some of the code it provides. For instance, one of the most critical aspects is that it is related to the domain knowledge of my smart contract, and here there is a term called "negative number". In the domain knowledge of smart contracts, the term "negative number" rarely appears

> because smart contracts can only handle positive numbers. So, once a negative number appears, I need to pay attention. Sometimes ChatGPT will produce something where it doesn't seem to know that negative numbers can exist in smart contracts. That's why, as you can see, I specifically mentioned in the prompt that all numbers in the code are positive, all are positive, all numbers are positive. But apparently, GPT doesn't seem to have fully considered this. So I need to look back and see if the so-called liquid paper change could be a negative number. Looking back at the original code, it is int256, not uint256. The so-called positive number actually means there must be a "u" in front, uint256. Here it is int256, indicating that its definition itself can be negative.

○ expert behavior: personal written workflow

1 Quotations:

1:4 ¶ 10 in P1.docx

Codes:

○ expert behavior: personal written workflow

Content:

> P1: Not entirely. This was written by myself, but other companies may have differences, but basically it's not too far off. This is an audit workflow. For example, the read me or white paper mentioned earlier is a written description of what this project is about. After obtaining it, I will then proceed to conduct an audit, specifically a general audit. First, I will roughly identify the overall logic. Based on the proposal it mentions, this logic is the main logic, so I will first look at its specific implementation.

○ expert behavior: purpose: best to find more

1 Quotations:

1:20 ¶ 46 – 49 in P1.docx

Codes:

○ expert behavior: purpose: best to find more

Content:

> Interviewer: This false alarm means either it reports an error or it underreports.
> P1: Right, yes, of course, it usually results in false reporting. Underreporting means omission, and in my opinion, when there is omission, if we check the checklist, I feel that the Knowledge Base is not large enough.
> Interviewer: Since you're doing this, there must be several goals. One goal is to accurately identify vulnerabilities, and then you need to report the identified ones to, I can understand it as the owner of the project itself. Another thing is that if you can't find all of them, it actually has no impact on you, right? The more you find, the better, but it's okay if you don't find them all.
> P1: Failure to identify all potential issues may have an impact on reputation. Yes, that's roughly it. The number of logical vulnerabilities, or such vulnerabilities in general, is unlimited, meaning it will never be completely secure, and there will always be some undetected problems. Of course, it's not the case for all code; the simpler the code, the fewer such issues there are.

○ expert behavior: using key concept/functionality structure

1 Quotations:

1:11 ¶ 22 – 28 in P1.docx

Codes:

○ expert behavior: using key concept/functionality structure ○ Reasoning: ChatGPT: generate function description, and do matching using key concept

Content:

> Interviewer: Let me ask again. Regarding your Knowledge Base, as you can see, it's listed from top to bottom, and it seems to lack a structure. When you perform matching, you can only rely on your own memory and experience to do so.
> P1: The matching process is roughly like this, that is, there are two ways to match. One is to match based on the code content. So, just now this was only a type list. In fact, in this library, since this library might be quite large, we will match based on the content code of a function, the data function code. For example, if we input this function code and then convert it into a functionality through GPT, this refers to the functional description of a function. For instance, let me give an example. Suppose we randomly select one, and it is a description of a function. Then we match the functional description I input with the functional descriptions in this list, and choose the corresponding vulnerability of the one that is most similar, because what are these data? These are vulnerability data. Each function has a possible vulnerability below it. So, after the matching, then ChatGPT, and generally when I add this, I usually use the key concept to ask questions. This is the functional description, this is the vulnerability description, the vulnerability description of the key concept, and I will ask questions in this way. As for the second category or this, this is the corresponding classification point I mentioned just now. What is it for? This is what you just said, it is to draw inferences from one instance. I will probably do it like this. What if it cannot be found? I will look for its parent class.
> Interviewer: Is it based on content? For example, is the key concept a simplification of functionality?
> P1: You can understand it this way: functionality is the scenario in which I use this knowledge. Key concept is this piece of knowledge, and we humans have defined a basic human behavior here, which is the behavior of using this knowledge in this scenario.
> P1: Right, that is to say, first I will match through functionality. I will construct all these functionalities, about 960 to 1000 of them, into a database. Then, by inputting a functionality, I will match the most similar functionality, find the corresponding key concept, and then ask ChatGPT. That is to say, what I actually match in the end is a key concept. That is, input a piece of code, and then get the most suitable key concept for it. Finally, I will ask questions about the key concept and the code, asking whether it is possible for such a vulnerability to occur in this code.
> Interviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

○ expert behavior: validating: vulnerablities could be narrowed down after a few times try

1 Quotations:

1:31 ¶ 75 in P1.docx

Codes:

○ expert behavior: validating: vulnerablities could be narrowed down after a few times try

Content:

> Right, so sometimes after I generally find this thing, I won't ask about it again. Because for the person submitting the vulnerability, it falls into two scenarios. One scenario is the internal audit of the project company, which must ensure that the report you provide to the project party is correct, without false positives. However, bug bounty or vulnerability bounty, or audit competition, usually doesn't consider whether the vulnerability you provided has false positives. There is someone on that side to help you check. There is a specific position, an identity called warden, which is used to check whether the information of the vulnerabilities provided by all participants is a real vulnerability. There will be such a check, and this check actually saves us effort. That is to say, after I generate 100 vulnerabilities, for example, after completing this task, I will write the vulnerability into an English description, including English translation, including description, recommendation, title, and maybe add a severity. Generally, we are required to submit these 4 items, and it will generate a relatively complete vulnerability report. At this time, I can directly copy and paste it onto their platform.

○ expert behavior: writing reports: can write in a way that it is important with GPT help

1 Quotations:

1:32 ¶ 76 – 77 in P1.docx

Codes:

○ expert behavior: writing reports: can write in a way that it is important with GPT help

Content:

> P1: Naturally, someone will check. I don't want to deal with this, so this is also a place where we've always wanted to take a bit of a shortcut. Well, there might be another aspect here, that is, sometimes we'll modify this part, because there's also an unwritten rule here. If you describe a vulnerability using method a, it might be a low-dimensional one, but if you describe it using method b, it might be a high-dimensional one. This has a significant impact on us, especially on our bounty, so I'll add a sentence later to describe this vulnerability as extremely serious. Right, there are indeed actual cases, there really are, but not many, there are truly actual cases. It will give a different description, a very serious one. Right, there are probably some shortcuts taken in there. I can give an example, like some of the audits I did before. You can take a direct look, you should be able to see this too. This is a report on the number of vulnerabilities I submitted before.
> For example, after clicking on this, there is a "my submission", where you can see that in this submission, I submitted 3 critical issues, 1 media issue, and a total of 9 issues. However, only 5 were approved, identified, and confirmed as vulnerabilities, which means 4 were not vulnerabilities and would be rejected by Secure Three. You can see why they were rejected. For example, "Merge the same issue for out of scope" or "Please give us more detail about the specific danger also".

○ Global understanding

0 Quotations

○ Global understanding: Background

0 Quotations

○ no need to learn new lanagauges very fast

1 Quotations:

6:6 ¶ 18 in P6.docx

Codes:

○ no need to learn new lanagauges very fast

Content:

> Then, the second thing is that after officially starting the audit, what is the difference between the process without ChatGPT and the process with ChatGPT? After entering the audit function, many current supply chains may implement their own EVMs. After implementing their own EVMs, what difference will there be? Different EVMs have different languages, but these different languages implement the same business model. So, at this time, there will be a difference.

○ Planning

20 Quotations:

1:2 ¶ 4 – 6 in P1.docx

Codes:

○ Planning: understanding code

Content:

> Interviewer: Is this project, for example, a public open source project or an internal project of your company?
> P1: Open source. For open source projects, after obtaining the project, I will probably read through some of its written content, including the white paper and its GitHub README, as shown here. That is to say, for some basic aspects of its design, during the reading process, I will not read very carefully. I only need to know what each specific contract is for. For example, as shown and described here, this.sol file refers to a smart contract file, and what it is used for. This is the first step of reading through.
> In the second step, after obtaining the specific code, we will choose to conduct an audit. Usually, we will also perform the audit based on the results of a general review. When we learn that it may be a lending project, or an investment project or a collateral project such as a deposit project, we usually prioritize the collateral logic within it. For example, it may involve a contract called options and futures, so I will look into it.

1:3 ¶ 8 in P1.docx

Codes:

○ Planning: filter key information: project selection through rough understanding

Content:

> P1: Right, yes, usually we choose which project to audit based on these specific proper nouns.

1:5 ¶ 11 in P1.docx

Codes:

○ Planning: understanding code: understanding functions of codebook

Content:

> So basically the first step is to read through the function names of the entire contract to see what functions the contract as a whole contains. For example, here I can clearly identify that the contract includes functions such as balance off, updating up, and a series of other functions. Then I will extract from these functions some that I can immediately tell what they do. For example, balance off clearly is used to obtain the corresponding balance. And functions like get founding rate and update founding rate clearly are used to set some key variables. So we can roughly classify them into several types. One type is functions for basic functions that we are relatively familiar with, such as balance off, or update, or set, or get. These basic function functions are only used to set or read some key variables in the contract and are used for some centralized adjustments. This is the first type.

1:6 ¶ 12 in P1.docx

Codes:

○ Planning: understanding code: understanding the logic and variables, whitebook

Content:

> For the second category, we may specifically examine the actual implementation of business logic, that is, the functions of those attributes that users can call. These functions are roughly divided into read and write functions. Reading is accessible to all, but writing is only allowed for administrators. For example, there is an "only owner" identifier here, which means that this function can only be written and manipulated by administrators. This is the first category. The second category is like "liquid", which is called by users. What we are most concerned about are also those functions that can be called by users. This is roughly our second step. After we have determined which contract to audit and how to audit it, our second step is to select the functions to be audited and split their functions from a certain contract. This is the second step.

1:7 ¶ 13 – 14 in P1.docx

Codes:

○ Planning: understanding through short report of external tools

Content:

> Interviewer: Previously, you mentioned a VSCode plugin called Solidity. Does it provide the functionality to highlight function names and variable names?
> P1: Since this contract is relatively short, if it were longer, we would use Solidity Visual Developer to make a rough judgment. How to judge?[Screen Sharing] You can understand these options as an overview or report. You can see that, in fact, it automatically generates a series of information about the function's scope and role based on the content of these functions. Of course, its format may not be ideal; it should be in Markdown format, read like Markdown, and it will show which parts can be read. I'll save it for you to take a look. This includes the function name, visibility, modifier, and whether it simulates pronunciation. This is a quick overview of the function I need to audit for the entire contract. It also includes some visibility settings, for example, External means it can only be called externally, and internal means it can only be called internally.

2:1 ¶ 5 in P2.docx

Codes:

○ Planning: understanding whitebook and code/logic

Content:

> P2: Let me see if I can find something from my previous audits. I'll share it. [Screen Sharing] Can you see this in VSCode? For example, when we receive an audit project, after getting it, we first check how much code it has. This is a small project with only one file. If it has only one file, it has more than 300 lines, which is considered relatively small. If it's a small file, then I'll first check what it does. We can look at the name of the contract. It may also provide some technical documents. We'll check what the entire project is about, and then look at its imports. For example, here there's a 1155, and access control. Here it uses signature-related stuff. This is 12 lines. This is to prevent reentrancy, and then there's a library for string processing. Looking at the contract name, it's probably a game, a contract related to a game. This is just a guess. All these are guesses. Then at this point, we can use GPT. We can also ask GPT what this contract is about, what its functions are.

2:2 ¶ 6 – 8 in P2.docx

Codes:

○ Planning: GPT for understanding: interaction with ChatGPT: understanding its variables

Content:

> Interviewer: Let me interrupt. How do you ask ChatGPT when you copy code and then ask it in GPT?
> P2: I'll just directly ask what functions this contract implements and what its purpose is, just a very simple question, and it will give a summary description, not too long. First, take a look at what the overall situation is like, and then ChatGPT will describe to you what kind of contract it is and what functions it implements. Next, you can ask about its global variables, for example, in this part from lines 28 to 52, what the purpose and function of each global variable is, and which business logic each corresponds to?
> Sometimes I also ask it from a different perspective, for example, in which functions a certain variable appears, and which functions modify it. In this way, if I think this variable is relatively important and may cause some serious risks if not handled properly, then I will focus on this variable. I will go to ChatGPT and ask it which functions use this variable and which functions modify it, so that I can focus on those functions. After understanding some of the functions and roles of global variables, we can then enter each function. This is different from what I do in VSCODE. Generally, I will provide a piece of code to it and ask it, and it will describe to me that this code is for handling token exchanges, and it will give me a detailed step-by-step analysis of this code. Today, I tried a contract and directly asked it. For example, this structure is an auction-related structure. If it is an auction-related structure, I want to know which functions it involves, and then I will ask it about the relevant business process. The sequence of the entire business process is the execution order of these functions. For example, it generally initializes first (i.e., init), and then the role and calling order of user-invoked functions.

2:3 ¶ 9 – 11 in P2.docx

Codes:

○ Planning: ChatGPT for easier understanding: ChatGPT: easier understanding

Content:

Interviewer: Right, it's like when you ask it to explain, you'll specify which aspects of which code you want to understand, right? For example, variables, which functions, and the business process between different functions, how different functions are combined to complete a business? You focus on understanding these points to complete your understanding of this code.
P2: Since a contract has two trading methods, one is auction, which relies on the "auction listing" structure, and the other is fixed-price, where the price is fixed and not auction-based. There are two trading business processes within the same contract. If I need to distinguish them, I'll just let ChatGPT handle it, because if I were to check them one by one myself, I might mix up these one or two business processes. Then ChatGPT will tell me that the "init auction" function is for initialization, and users call the "byte" function to place bids, and finally call this function to complete the auction.
If the next price is fixed, it also initializes first, then calls this function to make a purchase, and finally ends the fixed price. The two are different, but if they are written in the same contract, it will be difficult to distinguish for those who are just starting out. So, I will ask about it by understanding the business logic related to this variable.

2:4 ¶ 12 – 19 in P2.docx

Codes:

○ Challenge: GPT challenge: interaction: lack of enough context/cannot input too much code ○ Planning: ChatGPT for understanding: interaction

Content:

Interviewer: So different variables will have different associated business logic.
P2: Such a situation may exist, but the prerequisite is that you must first use your own experience or identify that it has two sets of logic.
Interviewer: So it's like when you're understanding this code, you first use your own judgment, then form a general internal understanding of this code, and then ask about the relatively minor points.
P2: This way is more efficient. If I can figure it out in a short time, it will be even more efficient. If not, I can also turn to ChatGPT and start asking from scratch.
Interviewer: Okay, in most cases, can you tell on your own or not?
P2: In most cases, people are just too lazy to read themselves and hand it over to GPT first.
Interviewer: But ChatGPT doesn't have the ability to digest the code of such a large project; instead, it can only handle a small snippet of code.
P2: Essentially, we identify them one by one as contract files; a project would be too large, so there's no other way.

3:1 ¶ 2 – 3 in P3.docx

Codes:

○ Planning: understanding code

Content:

Interviewer: OK, that's fine. Then I'll start by asking roughly. During the process of code auditing, do you use ChatGPT? If so, how do you use ChatGPT to assist you?
P3: Actually, I do use it, but perhaps not very much. Well, when I use it, it's mainly for some relatively more difficult grammar issues. I might just throw the code directly into ChatGPT and let it analyze what the code is roughly doing. Well, like how it processes the data. That's mainly it, but I generally don't put very long code in there, just a few lines, definitely no more than 10 lines of code.Interviewer: Oh, definitely no more than 10 lines. So, if you encounter very complex code, like a very complex project with, say, a dozen or twenty files, how do you use ChatGPT to assist you?P3: Generally, I don't put many files into ChatGPT. Well, actually, I haven't tried it either, because I always feel that it might be more accurate for it to analyze a small piece of code. Analyzing a large piece might, I think, be better done manually, like analyzing the logic of the code and such. It might be clearer to analyze manually.

3:2 ¶ 3 in P3.docx

Codes:

○ Planning: understanding code

Content:

> Interviewer: Oh, the process of code auditing generally consists of three stages: first planning, then reasoning, and then validating. I wonder if this is how you operate in your practice? For planning, it's probably something like first looking at these codes as a whole, then getting to know these codes, then trying to understand what they do. Then, for bug reasoning, it's about looking for logical loopholes in them. And thirdly, it's about writing a report to see if the loophole I've found is indeed there. Do you think your usual operating habits are like this?P3: It's probably like this. Because first, you definitely have to look at these codes. For example, when I look at a large block of code, I first need to know what its function is, what it's generally doing. Well, what kind of business is it handling? Well, first, you need to know this in general. Then, if you're analyzing the logic, you definitely have to trace the data flow. What kind of data is it processing? How is it processed step by step? Well, only in this process can you find out if there are any processing loopholes, right? Well, and thirdly, for example, if I find this loophole, then I definitely need to, for example, write a POC script or EXP to see if this loophole actually exists. It mainly revolves around these three points.

3:3 ¶ 3 in P3.docx

Codes:

○ Planning: use GPT for complex statements understanding

Content:

> Then, in this process, I think I use ChatGPT relatively more. Specifically, if a piece of code is relatively complex, I might put some complex statements into ChatGPT for interpretation. Well, and in the last stage, for example, when writing some utilization methods or scripts, I might tell ChatGPT a function and ask it to implement a certain function. So, it's mainly in these two stages that I use it. Well, and for myself, when doing logical analysis, I think it's more about manual analysis, just analyzing it myself.

4:1 ¶ 2 – 3 in P4.docx

Codes:

○ Planning: understanding code: understanding the logic and variables, whitebook

Content:

> Interviewer: Ok. Then the first question is, could you describe your practice in code auditing? For example, after you receive the project code, how do you conduct code auditing to identify its vulnerabilities? It would be best if you could share your screen and explain the entire process in detail by combining specific examples or projects.
> P4: When getting a project, first look at its project background, white paper, etc. First, understand the background of a project, what it does, which libraries it uses, and its preconditions. After knowing the project, then look at its code, and you will have a corresponding understanding. Then you can sort out the general framework, including what its main components are and what it interfaces with externally. All these can be learned from the white paper and project introduction. Then, when looking at the code, actually compare it. First, look at the main description of its code implementation, and check whether the comments are consistent with the precondition description, and whether they are consistent with the introduction or white paper.Then take a look at the process from the entry point of the entire project, that is, how to start and play the project, how to enter from that project, and where the data goes. The entire flowchart here needs to be sorted out. After sorting out the flowchart, you can refer to the overall framework to understand what this project is doing, and whether it is consistent at the code level. From the data passed in by the user, the modified state, where this framework is passed, where it is modified, and where it is stored.

5:1 ¶ 2 – 3 in P5.docx

Codes:

○ Planning: understanding code

Content:

> Interviewer: Let's get started. You can share an example of a code auditing task you've done before and explain how ChatGPT helped in the process.

> P5: In a very traditional audit, the thinking is all pretty much the same. First, look at the general whitepaper or some relevant materials about its technical architecture to facilitate a general understanding of what the project is about, right? Then read through the code, look at the framework and so on, and then look at those key functions and such. First, just do a general review of the entire framework to figure out what the project is mainly about and which aspects are relatively important for this project, such as those involving funds or some upgrades and the like. Pay attention and mark them. Then take a closer look. After getting an impression from the first reading, the second reading focuses on some important function interfaces, right? And then the entire process of some variables. At the beginning, some variables are defined, and then check if there are any possible abnormal situations in the state changes during the execution of the entire set of functions.

5:2 ¶ 4 in P5.docx

Codes:

○ Planning: identify key information

Content:

> Right, and then when conducting an audit, after marking those items, we should first list the high-risk issues that can be identified, and finally write the report. The reason for this report is that it not only includes high-risk issues, nor does every project have high-risk issues. Generally, there are relatively more medium-risk and low-risk issues to address. Based on some experience and common sense, we need to do some sorting of info and the like. If we use fewer tools, in fact, I use them relatively less. Because for static analysis now, it may be helpful for that kind of info, but there are fewer simple ones, and I often don't use it because that kind of info can be seen at a glance.

6:1 ¶ 2 – 3 in P6.docx

Codes:

○ Planning: understanding code

Content:

> Interviewer: Could you describe your practice process by combining an example of code auditing?
> P6: If I were to give an example, it would probably be something like taking a publicly available dataset of audit reports, which is perhaps used more frequently by everyone. This dataset contains some bug reports publicly announced by major audit firms regarding many projects.Before the audit, it was correct, but there is a drawback that we may not be able to see it from here. It has done classification, but we may not be able to see a complete project from it. For example, some submitted projects may not have published the complete project, but it has published problematic code snippets. Take this, this is a report I audited before. For example, if we didn't have ChatGPT or these large models before, when we got something, the first step in the audit process was to evaluate this thing. Whether it was a company or individual audit, now that we have these contracts, I will first give a pricing based on the number of lines of your code, your business model, and the complexity of, for example, mathematics implemented in the business model. Then the process of giving this pricing depends on the auditor's experience. Maybe based on your business model, we charge a certain amount of money. If there is a problem now, these companies rely heavily on or like these experienced auditors. But now, after the emergence of large models, what's the situation? When we interpret the business model of these projects or read the semantics of these codes, we no longer rely so much on the auditor's experience. We just need to Ctrl+C and Ctrl+V copy it into the corresponding ChatGPT, for example, throw it in here, and we can, for example.

6:2 ¶ 4 – 5 in P6.docx

Codes:

○ Planning: ChatGPT for project evaluation

Content:

> Interviewer: It's like you're using it to understand the code you want to audit, right?
> P6: Right, from manually assessing the complexity of a project, to for example, now I may need a lot of manual operations, but in the future, for example, we can use some APIs of ChatGPT to write a prompt, and then we can very quickly conduct an assessment for a specific project. Of course,

> many companies are already doing this now. Then, as long as we have enough datasets, we can provide, for example, how many days we can save and how much money we can charge for this project. Previously, it might have taken two or three days to assess, but now we may only need one or two minutes to get the job done.

6:10 ¶ 30 – 32 in P6.docx

Codes:

○ Planning: collaboration scenario

Content:

> Interviewer: So everyone has to look at the same code, right? It's like everyone points out some bugs, no, some vulnerabilities, and then sits down together to discuss which vulnerabilities should be eliminated.
> P6: That's right, and this is the vulnerability under review. When dealing with different languages, ChatGPT has provided us with a great deal of assistance, allowing us to avoid the need to master, or proficiently master, different languages, thus saving us the cost of learning these languages. Right.
> The second point is that, first, after evaluating the project, second, look at the language required for this project, and third, when we actually review and enter this project, as an auditor, you may need to read the code. For example, there are two methods. Before ChatGPT, what did we do first when we got a project? We first read its documentation to see what this thing is for. Right, after you master the documentation, then compare whether there are any differences between the code implementation and the documentation, and then compare whether there are any differences between the implementation of the comments and the documentation. There will be many comments on the code. Are there any differences between the implementation of these comments and the actual implementation? Right, then you will look, but if ChatGPT is available, it is actually very simple. We have a lot of things. Only after you read this can you know what this code is roughly doing, and then you have a general basic framework. For example, we check whether there is, of course, some companies will draw this diagram, such as drawing an architecture diagram similar to this, while some companies won't.

6:11 ¶ 33 in P6.docx

Codes:

○ Planning: understanding code: have a mindmap in mind and then decompose

Content:

> When you read code and comments, you may first need to have such a mental map in your mind, and then delve into it step by step. This is the process of auditing in the companies I've been involved with or in my personal auditing. This was the practice before ChatGPT, and then you go in and read it thoroughly. For example, when we look inside, say from the very beginning entry point, okay, what kind of issues might this contract have? This is the experience of reviewers. In your mind, okay, what kind of responsibility does it assume in this project, and what kind of vulnerabilities might this type of file have given the logic it implements? You go to look at the corresponding contracts and time limits with these questions in mind. For example, if we randomly click into a pre-sale, say in this pre-sale contract, okay, could there be overselling due to allowlist restrictions? Okay, we go through it line by line. That's roughly the idea. ChatGPT can help us understand the semantics of this stuff. For example, auditors audit in this way, and after having ChatGPT, another thing it can do for us is that, of course, many auditing companies may have their own large models, some trained by themselves, and of course, some may only do prompt engineering. For example, they would give it such prompt words in the front, and I'll show you what its output might look like.

6:12 ¶ 34 in P6.docx

Codes:

○ Planning: understanding code: help us understand the import relationship

Content:

> He can, for example, help us conduct detailed analysis, even identifying what you have imported, which contracts you have imported, what potential issues might occur with the imported contracts,

> including the logical relationships within the contracts, such as what variables I have defined, in which functions I have manipulated these variables, and so on. And it can help us analyze what a function does. Currently, we might be inputting prompts ourselves to analyze the contract, but if the company is working on this, it will have its own set of prompts, and we won't need to input prompts ourselves. It will analyze the contract based on different business models, such as the business model we just mentioned, where the contract is for pre-sale or airdrop, checking if there is an allowlist, and then analyze the contract using the company's internal prompts, and output corresponding content to help us understand the contract. This way, we can read the contract faster.

○ Planning and Reasoning: ChatGPT's help, understanding the logic flow faster

1 Quotations:

6:13 ¶ 35 – 36 in P6.docx

Codes:

○ Planning and Reasoning: ChatGPT's help, understanding the logic flow faster

Content:

> Interviewer: Right, if you think from a more high-level perspective, you first understand it, and after understanding, then you go on to look for the loopholes. So when ChatGPT helps you find different loopholes, what role does it play?
> P6: Helps me understand the logic of the contract more quickly, because of what? For example, when I'm looking at this contract, it's because I've previously reviewed relevant contracts or seen similar business logic, and I know what kind of vulnerabilities it might have. So, with these ideas in mind, when I look at the contract again, I'll be much clearer. However, the role of auditing is that, in many cases, it's these vulnerabilities or this business model that no one has written before, and logical vulnerabilities that have never occurred online before will also occur in this business model that no one has written before. But at this time, because no one has written this business model, the auditor has never seen it either. This is when ChatGPT is very useful. When we throw this code in, it will tell us what this thing is doing. When we read it again, it's like ChatGPT can help us draw a picture like this, except it's in text form, and it will be very convenient for us to read.

○ Planning preparation

1 Quotations:

1:1 ¶ 3 in P1.docx

Codes:

○ Planning preparation: data collection: obtaining background and requirement (whitebook)

Content:

> P1: Okay, let me give an example. For instance, if there's a smart contract project, I usually check the official website and go to GitHub to obtain some basic requirements for auditing the smart contract. This is the first step, to gather background information or understand the project, usually from its whitepaper.

○ Planning preparation: data collection: obtaining background and requirement (whitebook)

1 Quotations:

1:1 ¶ 3 in P1.docx

Codes:

○ Planning preparation: data collection: obtaining background and requirement (whitebook)

Content:

> P1: Okay, let me give an example. For instance, if there's a smart contract project, I usually check the official website and go to GitHub to obtain some basic requirements for auditing the smart contract. This is the first step, to gather background information or understand the project, usually from its whitepaper.

○ Planning: ChatGPT for easier understanding: ChatGPT: easier understanding

1 Quotations:

2:3 ¶ 9 – 11 in P2.docx

Codes:

○ Planning: ChatGPT for easier understanding: ChatGPT: easier understanding

Content:

> Interviewer: Right, it's like when you ask it to explain, you'll specify which aspects of which code you want to understand, right? For example, variables, which functions, and the business process between different functions, how different functions are combined to complete a business? You focus on understanding these points to complete your understanding of this code.
> P2: Since a contract has two trading methods, one is auction, which relies on the "auction listing" structure, and the other is fixed-price, where the price is fixed and not auction-based. There are two trading business processes within the same contract. If I need to distinguish them, I'll just let ChatGPT handle it, because if I were to check them one by one myself, I might mix up these one or two business processes. Then ChatGPT will tell me that the "init auction" function is for initialization, and users call the "byte" function to place bids, and finally call this function to complete the auction.
> If the next price is fixed, it also initializes first, then calls this function to make a purchase, and finally ends the fixed price. The two are different, but if they are written in the same contract, it will be difficult to distinguish for those who are just starting out. So, I will ask about it by understanding the business logic related to this variable.

○ Planning: ChatGPT for project evaluation

1 Quotations:

6:2 ¶ 4 – 5 in P6.docx

Codes:

○ Planning: ChatGPT for project evaluation

Content:

> Interviewer: It's like you're using it to understand the code you want to audit, right?
> P6: Right, from manually assessing the complexity of a project, to for example, now I may need a lot of manual operations, but in the future, for example, we can use some APIs of ChatGPT to write a prompt, and then we can very quickly conduct an assessment for a specific project. Of course, many companies are already doing this now. Then, as long as we have enough datasets, we can provide, for example, how many days we can save and how much money we can charge for this project. Previously, it might have taken two or three days to assess, but now we may only need one or two minutes to get the job done.

○ Planning: ChatGPT for understanding: interaction

1 Quotations:

2:4 ¶ 12 – 19 in P2.docx

Codes:

○ Challenge: GPT challenge: interaction: lack of enough context/cannot input too much code ○ Planning: ChatGPT for understanding: interaction

Content:

> Interviewer: So different variables will have different associated business logic.

P2: Such a situation may exist, but the prerequisite is that you must first use your own experience or identify that it has two sets of logic.
Interviewer: So it's like when you're understanding this code, you first use your own judgment, then form a general internal understanding of this code, and then ask about the relatively minor points.
P2: This way is more efficient. If I can figure it out in a short time, it will be even more efficient. If not, I can also turn to ChatGPT and start asking from scratch.
Interviewer: Okay, in most cases, can you tell on your own or not?
P2: In most cases, people are just too lazy to read themselves and hand it over to GPT first.
Interviewer: But ChatGPT doesn't have the ability to digest the code of such a large project; instead, it can only handle a small snippet of code.
P2: Essentially, we identify them one by one as contract files; a project would be too large, so there's no other way.

○ Planning: collaboration scenario

1 Quotations:

6:10 ¶ 30 – 32 in P6.docx

Codes:

○ Planning: collaboration scenario

Content:

Interviewer: So everyone has to look at the same code, right? It's like everyone points out some bugs, no, some vulnerabilities, and then sits down together to discuss which vulnerabilities should be eliminated.
P6: That's right, and this is the vulnerability under review. When dealing with different languages, ChatGPT has provided us with a great deal of assistance, allowing us to avoid the need to master, or proficiently master, different languages, thus saving us the cost of learning these languages. Right.
The second point is that, first, after evaluating the project, second, look at the language required for this project, and third, when we actually review and enter this project, as an auditor, you may need to read the code. For example, there are two methods. Before ChatGPT, what did we do first when we got a project? We first read its documentation to see what this thing is for. Right, after you master the documentation, then compare whether there are any differences between the code implementation and the documentation, and then compare whether there are any differences between the implementation of the comments and the documentation. There will be many comments on the code. Are there any differences between the implementation of these comments and the actual implementation? Right, then you will look, but if ChatGPT is available, it is actually very simple. We have a lot of things. Only after you read this can you know what this code is roughly doing, and then you have a general basic framework. For example, we check whether there is, of course, some companies will draw this diagram, such as drawing an architecture diagram similar to this, while some companies won't.

○ Planning: filter key information: project selection through rough understanding

1 Quotations:

1:3 ¶ 8 in P1.docx

Codes:

○ Planning: filter key information: project selection through rough understanding

Content:

P1: Right, yes, usually we choose which project to audit based on these specific proper nouns.

○ Planning: GPT for understanding: interaction with ChatGPT: understanding its variables

1 Quotations:

2:2 ¶ 6 – 8 in P2.docx

Codes:

- ○ Planning: GPT for understanding: interaction with ChatGPT: understanding its variables

Content:

> Interviewer: Let me interrupt. How do you ask ChatGPT when you copy code and then ask it in GPT?
> P2: I'll just directly ask what functions this contract implements and what its purpose is, just a very simple question, and it will give a summary description, not too long. First, take a look at what the overall situation is like, and then ChatGPT will describe to you what kind of contract it is and what functions it implements. Next, you can ask about its global variables, for example, in this part from lines 28 to 52, what the purpose and function of each global variable is, and which business logic each corresponds to?
> Sometimes I also ask it from a different perspective, for example, in which functions a certain variable appears, and which functions modify it. In this way, if I think this variable is relatively important and may cause some serious risks if not handled properly, then I will focus on this variable. I will go to ChatGPT and ask it which functions use this variable and which functions modify it, so that I can focus on those functions. After understanding some of the functions and roles of global variables, we can then enter each function. This is different from what I do in VSCODE. Generally, I will provide a piece of code to it and ask it, and it will describe to me that this code is for handling token exchanges, and it will give me a detailed step-by-step analysis of this code. Today, I tried a contract and directly asked it. For example, this structure is an auction-related structure. If it is an auction-related structure, I want to know which functions it involves, and then I will ask it about the relevant business process. The sequence of the entire business process is the execution order of these functions. For example, it generally initializes first (i.e., init), and then the role and calling order of user-invoked functions.

○ Planning: identify key information

1 Quotations:

5:2 ¶ 4 in P5.docx

Codes:

- ○ Planning: identify key information

Content:

> Right, and then when conducting an audit, after marking those items, we should first list the high-risk issues that can be identified, and finally write the report. The reason for this report is that it not only includes high-risk issues, nor does every project have high-risk issues. Generally, there are relatively more medium-risk and low-risk issues to address. Based on some experience and common sense, we need to do some sorting of info and the like. If we use fewer tools, in fact, I use them relatively less. Because for static analysis now, it may be helpful for that kind of info, but there are fewer simple ones, and I often don't use it because that kind of info can be seen at a glance.

○ Planning: understanding code

5 Quotations:

1:2 ¶ 4 – 6 in P1.docx

Codes:

- ○ Planning: understanding code

Content:

> Interviewer: Is this project, for example, a public open source project or an internal project of your company?
> P1: Open source. For open source projects, after obtaining the project, I will probably read through some of its written content, including the white paper and its GitHub README, as shown here. That is to say, for some basic aspects of its design, during the reading process, I will not read very carefully. I only need to know what each specific contract is for. For example, as shown and

described here, this.sol file refers to a smart contract file, and what it is used for. This is the first step of reading through.
In the second step, after obtaining the specific code, we will choose to conduct an audit. Usually, we will also perform the audit based on the results of a general review. When we learn that it may be a lending project, or an investment project or a collateral project such as a deposit project, we usually prioritize the collateral logic within it. For example, it may involve a contract called options and futures, so I will look into it.

3:1 ¶ 2 – 3 in P3.docx

Codes:

○ Planning: understanding code

Content:

Interviewer: OK, that's fine. Then I'll start by asking roughly. During the process of code auditing, do you use ChatGPT? If so, how do you use ChatGPT to assist you?
P3: Actually, I do use it, but perhaps not very much. Well, when I use it, it's mainly for some relatively more difficult grammar issues. I might just throw the code directly into ChatGPT and let it analyze what the code is roughly doing. Well, like how it processes the data. That's mainly it, but I generally don't put very long code in there, just a few lines, definitely no more than 10 lines of code.Interviewer: Oh, definitely no more than 10 lines. So, if you encounter very complex code, like a very complex project with, say, a dozen or twenty files, how do you use ChatGPT to assist you?P3: Generally, I don't put many files into ChatGPT. Well, actually, I haven't tried it either, because I always feel that it might be more accurate for it to analyze a small piece of code. Analyzing a large piece might, I think, be better done manually, like analyzing the logic of the code and such. It might be clearer to analyze manually.

3:2 ¶ 3 in P3.docx

Codes:

○ Planning: understanding code

Content:

Interviewer: Oh, the process of code auditing generally consists of three stages: first planning, then reasoning, and then validating. I wonder if this is how you operate in your practice? For planning, it's probably something like first looking at these codes as a whole, then getting to know these codes, then trying to understand what they do. Then, for bug reasoning, it's about looking for logical loopholes in them. And thirdly, it's about writing a report to see if the loophole I've found is indeed there. Do you think your usual operating habits are like this?P3: It's probably like this. Because first, you definitely have to look at these codes. For example, when I look at a large block of code, I first need to know what its function is, what it's generally doing. Well, what kind of business is it handling? Well, first, you need to know this in general. Then, if you're analyzing the logic, you definitely have to trace the data flow. What kind of data is it processing? How is it processed step by step? Well, only in this process can you find out if there are any processing loopholes, right? Well, and thirdly, for example, if I find this loophole, then I definitely need to, for example, write a POC script or EXP to see if this loophole actually exists. It mainly revolves around these three points.

5:1 ¶ 2 – 3 in P5.docx

Codes:

○ Planning: understanding code

Content:

Interviewer: Let's get started. You can share an example of a code auditing task you've done before and explain how ChatGPT helped in the process.
P5: In a very traditional audit, the thinking is all pretty much the same. First, look at the general whitepaper or some relevant materials about its technical architecture to facilitate a general understanding of what the project is about, right? Then read through the code, look at the framework and so on, and then look at those key functions and such. First, just do a general review of the entire framework to figure out what the project is mainly about and which aspects are

> relatively important for this project, such as those involving funds or some upgrades and the like. Pay attention and mark them. Then take a closer look. After getting an impression from the first reading, the second reading focuses on some important function interfaces, right? And then the entire process of some variables. At the beginning, some variables are defined, and then check if there are any possible abnormal situations in the state changes during the execution of the entire set of functions.

6:1 ¶ 2 – 3 in P6.docx

Codes:

○ Planning: understanding code

Content:

> Interviewer: Could you describe your practice process by combining an example of code auditing?
> P6: If I were to give an example, it would probably be something like taking a publicly available dataset of audit reports, which is perhaps used more frequently by everyone. This dataset contains some bug reports publicly announced by major audit firms regarding many projects.Before the audit, it was correct, but there is a drawback that we may not be able to see it from here. It has done classification, but we may not be able to see a complete project from it. For example, some submitted projects may not have published the complete project, but it has published problematic code snippets. Take this, this is a report I audited before. For example, if we didn't have ChatGPT or these large models before, when we got something, the first step in the audit process was to evaluate this thing. Whether it was a company or individual audit, now that we have these contracts, I will first give a pricing based on the number of lines of your code, your business model, and the complexity of, for example, mathematics implemented in the business model. Then the process of giving this pricing depends on the auditor's experience. Maybe based on your business model, we charge a certain amount of money. If there is a problem now, these companies rely heavily on or like these experienced auditors. But now, after the emergence of large models, what's the situation? When we interpret the business model of these projects or read the semantics of these codes, we no longer rely so much on the auditor's experience. We just need to Ctrl+C and Ctrl+V copy it into the corresponding ChatGPT, for example, throw it in here, and we can, for example.

○ Planning: understanding code: have a mindmap in mind and then decompose

1 Quotations:

6:11 ¶ 33 in P6.docx

Codes:

○ Planning: understanding code: have a mindmap in mind and then decompose

Content:

> When you read code and comments, you may first need to have such a mental map in your mind, and then delve into it step by step. This is the process of auditing in the companies I've been involved with or in my personal auditing. This was the practice before ChatGPT, and then you go in and read it thoroughly. For example, when we look inside, say from the very beginning entry point, okay, what kind of issues might this contract have? This is the experience of reviewers. In your mind, okay, what kind of responsibility does it assume in this project, and what kind of vulnerabilities might this type of file have given the logic it implements? You go to look at the corresponding contracts and time limits with these questions in mind. For example, if we randomly click into a pre-sale, say in this pre-sale contract, okay, could there be overselling due to allowlist restrictions? Okay, we go through it line by line. That's roughly the idea. ChatGPT can help us understand the semantics of this stuff. For example, auditors audit in this way, and after having ChatGPT, another thing it can do for us is that, of course, many auditing companies may have their own large models, some trained by themselves, and of course, some may only do prompt engineering. For example, they would give it such prompt words in the front, and I'll show you what its output might look like.

○ Planning: understanding code: help us understand the import relationship

1 Quotations:

6:12 ¶ 34 in P6.docx

Codes:

○ Planning: understanding code: help us understand the import relationship

Content:

> He can, for example, help us conduct detailed analysis, even identifying what you have imported, which contracts you have imported, what potential issues might occur with the imported contracts, including the logical relationships within the contracts, such as what variables I have defined, in which functions I have manipulated these variables, and so on. And it can help us analyze what a function does. Currently, we might be inputting prompts ourselves to analyze the contract, but if the company is working on this, it will have its own set of prompts, and we won't need to input prompts ourselves. It will analyze the contract based on different business models, such as the business model we just mentioned, where the contract is for pre-sale or airdrop, checking if there is an allowlist, and then analyze the contract using the company's internal prompts, and output corresponding content to help us understand the contract. This way, we can read the contract faster.

○ Planning: understanding code: understanding functions of codebook

1 Quotations:

1:5 ¶ 11 in P1.docx

Codes:

○ Planning: understanding code: understanding functions of codebook

Content:

> So basically the first step is to read through the function names of the entire contract to see what functions the contract as a whole contains. For example, here I can clearly identify that the contract includes functions such as balance off, updating up, and a series of other functions. Then I will extract from these functions some that I can immediately tell what they do. For example, balance off clearly is used to obtain the corresponding balance. And functions like get founding rate and update founding rate clearly are used to set some key variables. So we can roughly classify them into several types. One type is functions for basic functions that we are relatively familiar with, such as balance off, or update, or set, or get. These basic function functions are only used to set or read some key variables in the contract and are used for some centralized adjustments. This is the first type.

○ Planning: understanding code: understanding the logic and variables, whitebook

2 Quotations:

1:6 ¶ 12 in P1.docx

Codes:

○ Planning: understanding code: understanding the logic and variables, whitebook

Content:

> For the second category, we may specifically examine the actual implementation of business logic, that is, the functions of those attributes that users can call. These functions are roughly divided into read and write functions. Reading is accessible to all, but writing is only allowed for administrators. For example, there is an "only owner" identifier here, which means that this function can only be written and manipulated by administrators. This is the first category. The second category is like "liquid", which is called by users. What we are most concerned about are also those functions that can be called by users. This is roughly our second step. After we have determined which contract to audit and how to audit it, our second step is to select the functions to be audited and split their functions from a certain contract. This is the second step.

4:1 ¶ 2 – 3 in P4.docx

Codes:

○ Planning: understanding code: understanding the logic and variables, whitebook

Content:

> Interviewer: Ok. Then the first question is, could you describe your practice in code auditing? For example, after you receive the project code, how do you conduct code auditing to identify its vulnerabilities? It would be best if you could share your screen and explain the entire process in detail by combining specific examples or projects.
> P4: When getting a project, first look at its project background, white paper, etc. First, understand the background of a project, what it does, which libraries it uses, and its preconditions. After knowing the project, then look at its code, and you will have a corresponding understanding. Then you can sort out the general framework, including what its main components are and what it interfaces with externally. All these can be learned from the white paper and project introduction. Then, when looking at the code, actually compare it. First, look at the main description of its code implementation, and check whether the comments are consistent with the precondition description, and whether they are consistent with the introduction or white paper.Then take a look at the process from the entry point of the entire project, that is, how to start and play the project, how to enter from that project, and where the data goes. The entire flowchart here needs to be sorted out. After sorting out the flowchart, you can refer to the overall framework to understand what this project is doing, and whether it is consistent at the code level. From the data passed in by the user, the modified state, where this framework is passed, where it is modified, and where it is stored.

○ Planning: understanding through short report of external tools

1 Quotations:

1:7 ¶ 13 – 14 in P1.docx

Codes:

○ Planning: understanding through short report of external tools

Content:

> Interviewer: Previously, you mentioned a VSCode plugin called Solidity. Does it provide the functionality to highlight function names and variable names?
> P1: Since this contract is relatively short, if it were longer, we would use Solidity Visual Developer to make a rough judgment. How to judge?[Screen Sharing] You can understand these options as an overview or report. You can see that, in fact, it automatically generates a series of information about the function's scope and role based on the content of these functions. Of course, its format may not be ideal; it should be in Markdown format, read like Markdown, and it will show which parts can be read. I'll save it for you to take a look. This includes the function name, visibility, modifier, and whether it simulates pronunciation. This is a quick overview of the function I need to audit for the entire contract. It also includes some visibility settings, for example, External means it can only be called externally, and internal means it can only be called internally.

○ Planning: understanding whitebook and code/logic

1 Quotations:

2:1 ¶ 5 in P2.docx

Codes:

○ Planning: understanding whitebook and code/logic

Content:

> P2: Let me see if I can find something from my previous audits. I'll share it. [Screen Sharing] Can you see this in VSCode? For example, when we receive an audit project, after getting it, we first check how much code it has. This is a small project with only one file. If it has only one file, it has more than 300 lines, which is considered relatively small. If it's a small file, then I'll first check what it does. We can look at the name of the contract. It may also provide some technical documents. We'll check what the entire project is about, and then look at its imports. For example, here there's a 1155, and access control. Here it uses signature-related stuff. This is 12 lines. This is to prevent reentrancy, and then there's a library for string processing. Looking at the contract name, it's

> probably a game, a contract related to a game. This is just a guess. All these are guesses. Then at this point, we can use GPT. We can also ask GPT what this contract is about, what its functions are.

○ Planning: use GPT for complex statements understanding

1 Quotations:

⊜ 3:3 ¶ 3 in P3.docx

Codes:

○ Planning: use GPT for complex statements understanding

Content:

> Then, in this process, I think I use ChatGPT relatively more. Specifically, if a piece of code is relatively complex, I might put some complex statements into ChatGPT for interpretation. Well, and in the last stage, for example, when writing some utilization methods or scripts, I might tell ChatGPT a function and ask it to implement a certain function. So, it's mainly in these two stages that I use it. Well, and for myself, when doing logical analysis, I think it's more about manual analysis, just analyzing it myself.

○ Reasoning

12 Quotations:

⊜ 1:8 ¶ 15 – 16 in P1.docx

Codes:

○ Reasoning: Manual reasoning: check its comprehensiveness

Content:

> Interviewer: After you've looked at this variable and understood what it generally does, how do you then go about discovering its vulnerability?
> P1: The first way is that we first try to understand what this function is actually doing. Understanding what this function is actually doing, in this step, in the past when there was no ChatGPT, we would read it manually. For example, I would read it like this: first, I know it defines two things; second, I know it returns a series of data. Where does this data come from? It comes from request liquidation. I roughly know that the name "liquidate" generally means liquidation. So, in general, liquidation means that the user first has to request liquidation. According to its logic, that means we can first request liquidation to know what specific contents liquidation includes, or what things the content of liquidation will change, which are actually things like paper change and credit change. After I know it has requested this data, I continue to read the following series of logic, and I won't go into the specific logic. So, if I find there might be a problem here, how did I find it? When I read to this point, I will carefully consider whether this place is complete, that is, whether it has been fully considered. To put it simply, the most critical thing about a vulnerability is whether it has been fully considered. For example, if equator paper change is less than 0, can it directly execute the following operations? Do we still need to verify some other things?This step is actually about identifying the most critical point in finding vulnerabilities, which means that you need to understand the entire logic before determining whether it verifies what it should during the liquidation process. If it fails to verify, then it may be a vulnerability. What kind of vulnerability do we usually classify this as? Generally, it is a vulnerability of inconsistency or "inconsistency". This is also a type of vulnerability I recently compiled. It should do something but fails to do it, it should update but fails to update, and it should check but fails to check. If it updates one part but fails to update another part that also needs to be updated synchronously, then we consider this an inconsistent vulnerability. This is one type.

⊜ 1:9 ¶ 17 – 19 in P1.docx

Codes:

○ Reasoning: Manual reasoning: inconsistency vulnerability

Content:

Another approach is that based on the existing domain knowledge I currently have, I will match it to our domain knowledge. For example, what are some of the domain knowledge items in the context of liquidation? For instance, during the liquidation process, insufficient verification of the input amount may occur. When I initiate a liquidation to redeem my funds, a common form of vulnerability is that it may not verify my income amount, and simply liquidate and transfer the funds directly, which is definitely not acceptable. I will search within this framework according to a pattern to check for any corresponding potential vulnerabilities. This could be the second possible method.
Interviewer: Right, it means that in the first approach, you're essentially doing a bottom-up analysis without having the original code, and then trying to understand what still needs to be done but hasn't been. And in the second approach, you actually already know some common mistakes or vulnerabilities from your past experience, and then you match them to see where they are.
P1: Yes, you can look at this column [Screen Sharing], and you can understand that this is my Knowledge Base. This is my Knowledge Base. Of course, the reason I've summarized the Knowledge Base now is that I can find vulnerabilities based on this pattern and code. For example, ChatGPT can also do this, but what we're currently working on is still this project, a project where GPT automatically discovers vulnerabilities.

1:10 ¶ 20 – 21 in P1.docx

Codes:

○ Reasoning: Manual reasoning: checklist

Content:

Interviewer: For those without experience, like some others, is it impossible for them to quickly match through this, and they can only be more inclined to use the first method?
P1: It depends on the individual. One thing is that the more you review, the better. The second is that if there are such things, it also has a name called a check list. You can understand it as a tick list that might be used for the normal operation of machines in a factory. You should tick and check this check list before going live.

1:11 ¶ 22 – 28 in P1.docx

Codes:

○ expert behavior: using key concept/functionality structure ○ Reasoning: ChatGPT: generate function description, and do matching using key concept

Content:

Interviewer: Let me ask again. Regarding your Knowledge Base, as you can see, it's listed from top to bottom, and it seems to lack a structure. When you perform matching, you can only rely on your own memory and experience to do so.
P1: The matching process is roughly like this, that is, there are two ways to match. One is to match based on the code content. So, just now this was only a type list. In fact, in this library, since this library might be quite large, we will match based on the content code of a function, the data function code. For example, if we input this function code and then convert it into a functionality through GPT, this refers to the functional description of a function. For instance, let me give an example. Suppose we randomly select one, and it is a description of a function. Then we match the functional description I input with the functional descriptions in this list, and choose the corresponding vulnerability of the one that is most similar, because what are these data? These are vulnerability data. Each function has a possible vulnerability below it. So, after the matching, then ChatGPT, and generally when I add this, I usually use the key concept to ask questions. This is the functional description, this is the vulnerability description, the vulnerability description of the key concept, and I will ask questions in this way. As for the second category or this, this is the corresponding classification point I mentioned just now. What is it for? This is what you just said, it is to draw inferences from one instance. I will probably do it like this. What if it cannot be found? I will look for its parent class.
Interviewer: Is it based on content? For example, is the key concept a simplification of functionality?
P1: You can understand it this way: functionality is the scenario in which I use this knowledge. Key concept is this piece of knowledge, and we humans have defined a basic human behavior here, which is the behavior of using this knowledge in this scenario.
P1: Right, that is to say, first I will match through functionality. I will construct all these functionalities, about 960 to 1000 of them, into a database. Then, by inputting a functionality, I will match the most similar functionality, find the corresponding key concept, and then ask ChatGPT. That is to say, what I actually match in the end is a key concept. That is, input a piece of code, and

> then get the most suitable key concept for it. Finally, I will ask questions about the key concept and the code, asking whether it is possible for such a vulnerability to occur in this code.
> Interviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

1:12 ¶ 27 – 28 in P1.docx

Codes:

○ Reasoning: ChatGPT: matching rules for GPT

Content:

> nterviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

1:13 ¶ 29 – 32 in P1.docx

Codes:

○ Reasoning: Comparison: the usage of domain knowledge between human vs. ChatGPT

Content:

> Interviewer: The Excel table you just showed is your second method, which uses the Knowledge Base for matching.
> P1: Regarding the method of matching the Knowledge Base, yes, actually the first one is also quite similar. In fact, you can find that it also mentions "inconsist" here. The first one can actually be classified into this category. To put it simply, the current general way for the people I've defined, the auditors, to detect vulnerabilities is actually based on the knowledge in their own minds. Is there really a situation where this knowledge emerges from scratch, completely without any initial knowledge to refer to? I haven't done this in our current program yet, but it's indeed inevitable that we'll encounter this requirement. Then this requirement will need GPT to fully simulate human thinking.
> Interviewer: What does "starting from 0" mean?
> P1: You can see that a very necessary condition for me to detect vulnerabilities just now is that I have so much data, so much functionality, and key concepts to ask about, right? What if I don't have them? If I have nothing to ask, does it mean that vulnerabilities can't be detected? What if you're a newbie auditor, a newbie who knows nothing, and your so-called second category, functionality, and key concepts are empty?

1:15 ¶ 35 in P1.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: deceive GPT to get the vulnerbalities

Content:

> Then, for the second approach, I won't go through the so-called step-by-step questioning as before. I don't want to guide you anymore. Instead, I will fully rely on ChatGPT's own thinking mode. I will deceive it by saying that there is a vulnerability in this code, and asking it to help me find it. In fact, ChatGPT is more receptive to task-driven prompts, and sometimes it seems not very receptive to question-prompted prompts. For example, Your task is to pinpoint [vulnerabilities]. The second

> paragraph, we have already confirmed that the contract contains only one exploitable logic bug, is equivalent to me deceiving it and myself. I don't know if there is a vulnerability in this code, but I can ask it while telling it there is one and asking it to help me find it. This approach works very well for ChatGPT. Then, the following steps are all about restricting its output. So, overall, for the current ChatGPT, I have roughly three methods to make it find vulnerabilities: the first is based on checklists, the second is based on functionality, and the third is based on key concepts.

1:16 ¶ 36 – 37 in P1.docx

Codes:

○ Reasoning: Prompt engineering

Content:

> Interviewer: But this is equivalent to you having already selected one of them and then letting it go. Have you ever tried directly uploading this document of yours to ChatGPT?
> P1: The effect is very poor, and ChatGPT's output ability is very weak. This is roughly such a comparison. Currently, our third functionality uses this, where contract a, b cannot C. In fact, this contract a is exactly the so-called functionality just mentioned, and B cannot BC is the key concept here. Based on knowledge, what vulnerability? Then this is the second solution, and the last solution just mentioned is the so-called deception, and the third is actually a combination of the two methods, which can also produce output. So, how about their accuracy and output quality? So, what is accuracy? Is there actually any false alarm? And output quality refers to whether the output is a real vulnerability or just a security recommendation. That's why I choose this one instead of the first one, because the first one has too many so-called security recommendations, too many security recommendations, many of which are completely useless.

2:5 ¶ 20 – 25 in P2.docx

Codes:

○ Reasoning: ChatGPT: 1) gain some initial direction 2) let ChatGPT to search in some specific direction

Content:

> Interviewer: Okay, if it's at the project level, you have to understand it on your own, right?
> P2: Right, if it's a large-scale project, we can only first manually divide it into several modules, then further break it down into each module, and then use ChatGPT to learn and analyze such programs.
> Interviewer: When you understand it, how do you make it help you?
> P2: First, now it lists all these functions. For example, if I want to study this business process, I need to focus on these functions. If I want to find them through GPT, I feel it's a bit of a shot in the dark. For example, directly below, I'll ask, "Please analyze this function, which has some security issues. Please analyze the security issues therein and point them out one by one." When I input this paragraph, I'm not sure. I'm not even certain if it has any problems, and then I input it for it to point out. Then it will, based on its experience and potential security considerations, check whether the auction has started. It says that within the time limit of this function, it doesn't check whether the auction has started, and then it suggests making a clear check first. This kind of analysis is rather general; it's not tied to a specific problem in a specific piece of code. This is a bit of a shot in the dark, to see if what it gives is valuable, because if I have no direction, I'll just see what it says first.
> Interviewer: Do you think that in most cases you first let it search and then it gives you some directions, or you have already come up with some directions and then let it confirm them for you?
> P2: I think both situations exist. If I have no ideas, I'll teach it; if I do have ideas, for example, when there are some arithmetic operations within a function, especially those involving division, I'll tell it that division can lead to precision loss. Then I'll ask you to help me analyze the calculation issues within this function, and I'll emphasize the calculation issues, specifically whether there will be precision loss in the calculations within this function.

2:6 ¶ 26 – 35 in P2.docx

Codes:

○ Reasoning: ChatGPT: GPTs: using some prompts notes with addons

Content:

Interviewer: Let me check the question. You just mentioned different types of vulnerability. Okay, so when it comes to different types of vulnerability, do you have different debugging methods related to GDP?
P2: Specifically, under what scenario?
Interviewer: For example, some vulnerabilities are logical issues, while others may refer to variables, such as a variable not being allowed to be negative. When interacting with GPT, are there any differences in dealing with different types of such errors?
P2: I have tried to do this. I have used a numerical calculation myself, which is what I just mentioned. I pay special attention to the calculation issues of this numerical value, whether the parameter range is correct, whether it needs to be restricted, and then the issue of precision loss.
Interviewer: Did you write this prompt yourself? And it didn't provide this to you.
P2: No, this is based on my own experience. For example, regarding the problems that may arise in calculations, I just jotted down a few. Here, only these three paragraphs are written, and it is specifically used to analyze calculation problems.
Interviewer: Were these originally there or did you come up with them?
P2: This was generated by itself; I don't think I have any impression of writing these. Then here, it's my own note, and I just threw it up with a try-it-out attitude.
Interviewer: Do you think it's useful? Does it make use of this note?
P2: There is no particularly obvious feeling.

4:2 ¶ 3 – 4 in P4.docx

Codes:

○ Reasoning: ChatGPT: matching rules for GPT ○ Reasoning: comparision with domain knowledge

Content:

After having a general understanding, you can apply some original risk items, describe some historical experiences, or summarize some security bugs. You can check which aspects may have which problems. After a general review, you can go deeper. Going deeper means that after reviewing the basic security information, you can identify these basic issues. Going even deeper can help you understand the operating principle of the entire project thoroughly, because the pre-information is already clear, and there won't be some other relatively low-level issues. Then you can delve into some of its code rules and logic, and judge where it may be the weakest. For example, for some data passed in by the user, which ones must have permission control but don't, you should judge whether it has permission control, which interfaces are accessible to users, and which are data exposure interfaces. It is in these places that you may read abnormal data, and they can use this data to do things, or you can directly modify some data. This is roughly delving into it. Then, the mathematical aspects are not quite the same as normal vulnerabilities. You need to consider some precision issues, some manual input issues, as well as some other calculation methods, and it may also involve some aspects related to the company. This set of logic is quite different from that of vulnerabilities, just at these levels. As for the general overall framework, that is, the thinking of an audit, first judge the project information, then based on the project information, analyze its code logic, and then under the judgment of the code structure, check whether there are some basic issues. If not, you can delve deeper to identify some weak points of the project, then try to address some corresponding issues, and finally look at the mathematical aspects to see if there are any issues with formulas or other aspects. Roughly speaking, it is like this.

5:4 ¶ 7 – 16 in P5.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: 1. auditing based on domain knowledge; 2. deceive; 3. original conversation with CoT ○ Validating: manual work: GPT results need to be manually checked

Content:

Interviewer: Assistance with output format and content. Does it refer to the format of the final report?
P5: Right. You can give him a template, then paste the code and describe it, and then he will give you a specific template and specific output based on the template.
Interviewer: Just now you mentioned that you look for vulnerabilities, which means these vulnerabilities come in different types, right? Then these different types of vulnerabilities are, so to speak, some you assign to GPT to look for, while you look for others yourself. Let me rephrase my question. You just mentioned different types of issues, such as some being variables and some

being function functionalities. So which ones do you think you can rely on GPT to help you find, and which ones do you prefer to find on your own?
P5: For this, I haven't fully studied and understood how to use ChatGPT to handle these things. I can only give it a good prompt, then provide it with the code and ask it to output some content. But when it comes to having it do specific tasks like sorting out the state changes of its variables, I may not have attempted that yet.
Interviewer: So when you use ChatGPT now, it's just equivalent to a single-turn conversation, right? For example, you input code to it, then let it directly give you an output, and then you'll ask step by step like this to guide it to find some bugs.
P5: Right.
Interviewer: Are you referring to the second type, or the single-round dialogue?
P5: I'm the kind of person who, when given the code, either can switch it or, if you're not satisfied with the result, can retry. I'll just keep retrying to see if I can get something effective.
Interviewer: You're not saying that, for example, after you give it something, it generates a result, and then after you look at this result, you continue to ask further in-depth questions based on this result, right?
P5: Some will. If you can tell at a glance that it's a false alarm, there's no need to let it retry; just let it provide a different answer and don't dig deeper. If it's ambiguous but somewhat relevant, you may need to ask. Right, but basically, I only make a judgment when it's relatively certain. If it's ambiguous, I'll ask a bit. If the answer is still rather vague after asking, I'll directly retry and let it take a different path.

○ Reasoning: ChatGPT prompting strategy: 1. auditing based on domain knowledge; 2. deceive; 3. original conversation with CoT

1 Quotations:

5:4 ¶ 7 – 16 in P5.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: 1. auditing based on domain knowledge; 2. deceive; 3. original conversation with CoT ○ Validating: manual work: GPT results need to be manually checked

Content:

Interviewer: Assistance with output format and content. Does it refer to the format of the final report?
P5: Right. You can give him a template, then paste the code and describe it, and then he will give you a specific template and specific output based on the template.
Interviewer: Just now you mentioned that you look for vulnerabilities, which means these vulnerabilities come in different types, right? Then these different types of vulnerabilities are, so to speak, some you assign to GPT to look for, while you look for others yourself. Let me rephrase my question. You just mentioned different types of issues, such as some being variables and some
being function functionalities. So which ones do you think you can rely on GPT to help you find, and which ones do you prefer to find on your own?
P5: For this, I haven't fully studied and understood how to use ChatGPT to handle these things. I can only give it a good prompt, then provide it with the code and ask it to output some content. But when it comes to having it do specific tasks like sorting out the state changes of its variables, I may not have attempted that yet.
Interviewer: So when you use ChatGPT now, it's just equivalent to a single-turn conversation, right? For example, you input code to it, then let it directly give you an output, and then you'll ask step by step like this to guide it to find some bugs.
P5: Right.
Interviewer: Are you referring to the second type, or the single-round dialogue?
P5: I'm the kind of person who, when given the code, either can switch it or, if you're not satisfied with the result, can retry. I'll just keep retrying to see if I can get something effective.
Interviewer: You're not saying that, for example, after you give it something, it generates a result, and then after you look at this result, you continue to ask further in-depth questions based on this result, right?
P5: Some will. If you can tell at a glance that it's a false alarm, there's no need to let it retry; just let it provide a different answer and don't dig deeper. If it's ambiguous but somewhat relevant, you may need to ask. Right, but basically, I only make a judgment when it's relatively certain. If it's

> ambiguous, I'll ask a bit. If the answer is still rather vague after asking, I'll directly retry and let it take a different path.

○ Reasoning: ChatGPT prompting strategy: deceive GPT to get the vulnerbalities

1 Quotations:

1:15 ¶ 35 in P1.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: deceive GPT to get the vulnerbalities

Content:

> Then, for the second approach, I won't go through the so-called step-by-step questioning as before. I don't want to guide you anymore. Instead, I will fully rely on ChatGPT's own thinking mode. I will deceive it by saying that there is a vulnerability in this code, and asking it to help me find it. In fact, ChatGPT is more receptive to task-driven prompts, and sometimes it seems not very receptive to question-prompted prompts. For example, Your task is to pinpoint [vulnerabilities]. The second paragraph, we have already confirmed that the contract contains only one exploitable logic bug, is equivalent to me deceiving it and myself. I don't know if there is a vulnerability in this code, but I can ask it while telling it there is one and asking it to help me find it. This approach works very well for ChatGPT. Then, the following steps are all about restricting its output. So, overall, for the current ChatGPT, I have roughly three methods to make it find vulnerabilities: the first is based on checklists, the second is based on functionality, and the third is based on key concepts.

○ Reasoning: ChatGPT: 1) gain some initial direction 2) let ChatGPT to search in some specific direction

1 Quotations:

2:5 ¶ 20 – 25 in P2.docx

Codes:

○ Reasoning: ChatGPT: 1) gain some initial direction 2) let ChatGPT to search in some specific direction

Content:

> Interviewer: Okay, if it's at the project level, you have to understand it on your own, right?
> P2: Right, if it's a large-scale project, we can only first manually divide it into several modules, then further break it down into each module, and then use ChatGPT to learn and analyze such programs.
> Interviewer: When you understand it, how do you make it help you?
> P2: First, now it lists all these functions. For example, if I want to study this business process, I need to focus on these functions. If I want to find them through GPT, I feel it's a bit of a shot in the dark. For example, directly below, I'll ask, "Please analyze this function, which has some security issues. Please analyze the security issues therein and point them out one by one." When I input this paragraph, I'm not sure. I'm not even certain if it has any problems, and then I input it for it to point out. Then it will, based on its experience and potential security considerations, check whether the auction has started. It says that within the time limit of this function, it doesn't check whether the auction has started, and then it suggests making a clear check first. This kind of analysis is rather general; it's not tied to a specific problem in a specific piece of code. This is a bit of a shot in the dark, to see if what it gives is valuable, because if I have no direction, I'll just see what it says first.
> Interviewer: Do you think that in most cases you first let it search and then it gives you some directions, or you have already come up with some directions and then let it confirm them for you?
> P2: I think both situations exist. If I have no ideas, I'll teach it; if I do have ideas, for example, when there are some arithmetic operations within a function, especially those involving division, I'll tell it that division can lead to precision loss. Then I'll ask you to help me analyze the calculation issues within this function, and I'll emphasize the calculation issues, specifically whether there will be precision loss in the calculations within this function.

○ Reasoning: ChatGPT: generate function description, and do matching using key concept

1 Quotations:

1:11 ¶ 22 – 28 in P1.docx

Codes:

○ expert behavior: using key concept/functionality structure ○ Reasoning: ChatGPT: generate function description, and do matching using key concept

Content:

> Interviewer: Let me ask again. Regarding your Knowledge Base, as you can see, it's listed from top to bottom, and it seems to lack a structure. When you perform matching, you can only rely on your own memory and experience to do so.
> P1: The matching process is roughly like this, that is, there are two ways to match. One is to match based on the code content. So, just now this was only a type list. In fact, in this library, since this library might be quite large, we will match based on the content code of a function, the data function code. For example, if we input this function code and then convert it into a functionality through GPT, this refers to the functional description of a function. For instance, let me give an example. Suppose we randomly select one, and it is a description of a function. Then we match the functional description I input with the functional descriptions in this list, and choose the corresponding vulnerability of the one that is most similar, because what are these data? These are vulnerability data. Each function has a possible vulnerability below it. So, after the matching, then ChatGPT, and generally when I add this, I usually use the key concept to ask questions. This is the functional description, this is the vulnerability description, the vulnerability description of the key concept, and I will ask questions in this way. As for the second category or this, this is the corresponding classification point I mentioned just now. What is it for? This is what you just said, it is to draw inferences from one instance. I will probably do it like this. What if it cannot be found? I will look for its parent class.
> Interviewer: Is it based on content? For example, is the key concept a simplification of functionality?
> P1: You can understand it this way: functionality is the scenario in which I use this knowledge. Key concept is this piece of knowledge, and we humans have defined a basic human behavior here, which is the behavior of using this knowledge in this scenario.
> P1: Right, that is to say, first I will match through functionality. I will construct all these functionalities, about 960 to 1000 of them, into a database. Then, by inputting a functionality, I will match the most similar functionality, find the corresponding key concept, and then ask ChatGPT. That is to say, what I actually match in the end is a key concept. That is, input a piece of code, and then get the most suitable key concept for it. Finally, I will ask questions about the key concept and the code, asking whether it is possible for such a vulnerability to occur in this code.
> Interviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

○ Reasoning: ChatGPT: GPTs: using some prompts notes with addons

1 Quotations:

2:6 ¶ 26 – 35 in P2.docx

Codes:

○ Reasoning: ChatGPT: GPTs: using some prompts notes with addons

Content:

> Interviewer: Let me check the question. You just mentioned different types of vulnerability. Okay, so when it comes to different types of vulnerability, do you have different debugging methods related to GDP?
> P2: Specifically, under what scenario?
> Interviewer: For example, some vulnerabilities are logical issues, while others may refer to variables, such as a variable not being allowed to be negative. When interacting with GPT, are there any differences in dealing with different types of such errors?
> P2: I have tried to do this. I have used a numerical calculation myself, which is what I just mentioned. I pay special attention to the calculation issues of this numerical value, whether the parameter range is correct, whether it needs to be restricted, and then the issue of precision loss.
> Interviewer: Did you write this prompt yourself? And it didn't provide this to you.
> P2: No, this is based on my own experience. For example, regarding the problems that may arise in calculations, I just jotted down a few. Here, only these three paragraphs are written, and it is specifically used to analyze calculation problems.
> Interviewer: Were these originally there or did you come up with them?
> P2: This was generated by itself; I don't think I have any impression of writing these. Then here, it's my own note, and I just threw it up with a try-it-out attitude.
> Interviewer: Do you think it's useful? Does it make use of this note?
> P2: There is no particularly obvious feeling.

○ Reasoning: ChatGPT: matching rules for GPT

2 Quotations:

1:12 ¶ 27 – 28 in P1.docx

Codes:
○ Reasoning: ChatGPT: matching rules for GPT

Content:

> nterviewer: Essentially, your functionality is to describe what that code does, and then your key concept could be the potential vulnerabilities that might arise within the functionality.
> P1: Right, if the key concept is not found, I will look for its category. It belongs to the category of repeated array elements, and then I will ask other key concepts under this category. Can you understand what I mean? First, look for the parent class, then the child class. For example, I think there might be some pricing issues here, that is, there might be pricing problems in between. However, when humans think, human thinking might have some problems, but it seems not to occur, unless it is generalized or extended to something similar. Is there a possibility that there might be some similar problems, but not exactly the same as this one?

4:2 ¶ 3 – 4 in P4.docx

Codes:
○ Reasoning: ChatGPT: matching rules for GPT ○ Reasoning: comparision with domain knowledge

Content:

> After having a general understanding, you can apply some original risk items, describe some historical experiences, or summarize some security bugs. You can check which aspects may have which problems. After a general review, you can go deeper. Going deeper means that after reviewing the basic security information, you can identify these basic issues. Going even deeper can help you understand the operating principle of the entire project thoroughly, because the pre-information is already clear, and there won't be some other relatively low-level issues. Then you can delve into some of its code rules and logic, and judge where it may be the weakest. For example, for some data passed in by the user, which ones must have permission control but don't, you should judge whether it has permission control, which interfaces are accessible to users, and which are data exposure interfaces. It is in these places that you may read abnormal data, and they can use this data to do things, or you can directly modify some data. This is roughly delving into it. Then, the mathematical aspects are not quite the same as normal vulnerabilities. You need to consider some precision issues, some manual input issues, as well as some other calculation methods, and it may also involve some aspects related to the company. This set of logic is quite different from that of vulnerabilities, just at these levels. As for the general overall framework, that is, the thinking of an audit, first judge the project information, then based on the project information,

> analyze its code logic, and then under the judgment of the code structure, check whether there are some basic issues. If not, you can delve deeper to identify some weak points of the project, then try to address some corresponding issues, and finally look at the mathematical aspects to see if there are any issues with formulas or other aspects. Roughly speaking, it is like this.

○ Reasoning: comparision with domain knowledge

1 Quotations:

4:2 ¶ 3 – 4 in P4.docx

Codes:

○ Reasoning: ChatGPT: matching rules for GPT ○ Reasoning: comparision with domain knowledge

Content:

> After having a general understanding, you can apply some original risk items, describe some historical experiences, or summarize some security bugs. You can check which aspects may have which problems. After a general review, you can go deeper. Going deeper means that after reviewing the basic security information, you can identify these basic issues. Going even deeper can help you understand the operating principle of the entire project thoroughly, because the pre-information is already clear, and there won't be some other relatively low-level issues. Then you can delve into some of its code rules and logic, and judge where it may be the weakest. For example, for some data passed in by the user, which ones must have permission control but don't, you should judge whether it has permission control, which interfaces are accessible to users, and which are data exposure interfaces. It is in these places that you may read abnormal data, and they can use this data to do things, or you can directly modify some data. This is roughly delving into it. Then, the mathematical aspects are not quite the same as normal vulnerabilities. You need to consider some precision issues, some manual input issues, as well as some other calculation methods, and it may also involve some aspects related to the company. This set of logic is quite different from that of vulnerabilities, just at these levels. As for the general overall framework, that is, the thinking of an audit, first judge the project information, then based on the project information, analyze its code logic, and then under the judgment of the code structure, check whether there are some basic issues. If not, you can delve deeper to identify some weak points of the project, then try to address some corresponding issues, and finally look at the mathematical aspects to see if there are any issues with formulas or other aspects. Roughly speaking, it is like this.

○ Reasoning: Comparison: the usage of domain knowledge between human vs. ChatGPT

1 Quotations:

1:13 ¶ 29 – 32 in P1.docx

Codes:

○ Reasoning: Comparison: the usage of domain knowledge between human vs. ChatGPT

Content:

> Interviewer: The Excel table you just showed is your second method, which uses the Knowledge Base for matching.
> P1: Regarding the method of matching the Knowledge Base, yes, actually the first one is also quite similar. In fact, you can find that it also mentions "inconsist" here. The first one can actually be classified into this category. To put it simply, the current general way for the people I've defined, the auditors, to detect vulnerabilities is actually based on the knowledge in their own minds. Is there really a situation where this knowledge emerges from scratch, completely without any initial knowledge to refer to? I haven't done this in our current program yet, but it's indeed inevitable that we'll encounter this requirement. Then this requirement will need GPT to fully simulate human thinking.
> Interviewer: What does "starting from 0" mean?
> P1: You can see that a very necessary condition for me to detect vulnerabilities just now is that I have so much data, so much functionality, and key concepts to ask about, right? What if I don't have them? If I have nothing to ask, does it mean that vulnerabilities can't be detected? What if

> you're a newbie auditor, a newbie who knows nothing, and your so-called second category, functionality, and key concepts are empty?

○ Reasoning: Manual reasoning strategy: matching rules: match from parent pattern to child pattern

0 Quotations

○ Reasoning: Manual reasoning: check its comprehensiveness

1 Quotations:

1:8 ¶ 15 – 16 in P1.docx

Codes:

○ Reasoning: Manual reasoning: check its comprehensiveness

Content:

> Interviewer: After you've looked at this variable and understood what it generally does, how do you then go about discovering its vulnerability?
> P1: The first way is that we first try to understand what this function is actually doing. Understanding what this function is actually doing, in this step, in the past when there was no ChatGPT, we would read it manually. For example, I would read it like this: first, I know it defines two things; second, I know it returns a series of data. Where does this data come from? It comes from request liquidation. I roughly know that the name "liquidate" generally means liquidation. So, in general, liquidation means that the user first has to request liquidation. According to its logic, that means we can first request liquidation to know what specific contents liquidation includes, or what things the content of liquidation will change, which are actually things like paper change and credit change. After I know it has requested this data, I continue to read the following series of logic, and I won't go into the specific logic. So, if I find there might be a problem here, how did I find it? When I read to this point, I will carefully consider whether this place is complete, that is, whether it has been fully considered. To put it simply, the most critical thing about a vulnerability is whether it has been fully considered. For example, if equator paper change is less than 0, can it directly execute the following operations? Do we still need to verify some other things?This step is actually about identifying the most critical point in finding vulnerabilities, which means that you need to understand the entire logic before determining whether it verifies what it should during the liquidation process. If it fails to verify, then it may be a vulnerability. What kind of vulnerability do we usually classify this as? Generally, it is a vulnerability of inconsistency or "inconsistency". This is also a type of vulnerability I recently compiled. It should do something but fails to do it, it should update but fails to update, and it should check but fails to check. If it updates one part but fails to update another part that also needs to be updated synchronously, then we consider this an inconsistent vulnerability. This is one type.

○ Reasoning: Manual reasoning: checklist

1 Quotations:

1:10 ¶ 20 – 21 in P1.docx

Codes:

○ Reasoning: Manual reasoning: checklist

Content:

> Interviewer: For those without experience, like some others, is it impossible for them to quickly match through this, and they can only be more inclined to use the first method?
> P1: It depends on the individual. One thing is that the more you review, the better. The second is that if there are such things, it also has a name called a check list. You can understand it as a tick list that might be used for the normal operation of machines in a factory. You should tick and check this check list before going live.

○ Reasoning: Manual reasoning: inconsistency vulnerability

1 Quotations:

1:9 ¶ 17 – 19 in P1.docx

Codes:

○ Reasoning: Manual reasoning: inconsistency vulnerability

Content:

> Another approach is that based on the existing domain knowledge I currently have, I will match it to our domain knowledge. For example, what are some of the domain knowledge items in the context of liquidation? For instance, during the liquidation process, insufficient verification of the input amount may occur. When I initiate a liquidation to redeem my funds, a common form of vulnerability is that it may not verify my income amount, and simply liquidate and transfer the funds directly, which is definitely not acceptable. I will search within this framework according to a pattern to check for any corresponding potential vulnerabilities. This could be the second possible method.
> Interviewer: Right, it means that in the first approach, you're essentially doing a bottom-up analysis without having the original code, and then trying to understand what still needs to be done but hasn't been. And in the second approach, you actually already know some common mistakes or vulnerabilities from your past experience, and then you match them to see where they are.
> P1: Yes, you can look at this column [Screen Sharing], and you can understand that this is my Knowledge Base. This is my Knowledge Base. Of course, the reason I've summarized the Knowledge Base now is that I can find vulnerabilities based on this pattern and code. For example, ChatGPT can also do this, but what we're currently working on is still this project, a project where GPT automatically discovers vulnerabilities.

○ Reasoning: Prompt engineering

1 Quotations:

1:16 ¶ 36 – 37 in P1.docx

Codes:

○ Reasoning: Prompt engineering

Content:

> Interviewer: But this is equivalent to you having already selected one of them and then letting it go. Have you ever tried directly uploading this document of yours to ChatGPT?
> P1: The effect is very poor, and ChatGPT's output ability is very weak. This is roughly such a comparison. Currently, our third functionality uses this, where contract a, b cannot C. In fact, this contract a is exactly the so-called functionality just mentioned, and B cannot BC is the key concept here. Based on knowledge, what vulnerability? Then this is the second solution, and the last solution just mentioned is the so-called deception, and the third is actually a combination of the two methods, which can also produce output. So, how about their accuracy and output quality? So, what is accuracy? Is there actually any false alarm? And output quality refers to whether the output is a real vulnerability or just a security recommendation. That's why I choose this one instead of the first one, because the first one has too many so-called security recommendations, too many security recommendations, many of which are completely useless.

○ the estimitation of time is saved

1 Quotations:

6:4 ¶ 12 – 15 in P6.docx

Codes:

○ the estimitation of time is saved

Content:

> Interviewer: Now that the time has been reduced, will the amount of money received also decrease?

> P6: No, it's because for an auditing firm, it's not necessary to deliberately extend the cycle of every project. What's more important for an auditing firm is its external reputation in auditing. Among the 10 projects you may have audited, only one project may have a problem, or there may be 0 problems, i.e., 0 bugs. Yes, these are the fundamental competitiveness and market competitiveness of an auditing firm.
> Interviewer: So it's equivalent to, for example, a process that previously took 10 days to complete. For instance, if our project charged 10,000, now it can be completed in 5 days but still earn 10,000.
> P6: That's not what I meant. It's that our assessment cycle has been shortened. Previously, when my client engaged an auditor, okay, I needed to provide you with an assessment cycle for your project within three or two days. Then, during the assessment process, the auditor still had to continuously communicate with the project party to understand what your project specifically did, what the model was, and review the code, etc. Right, but if there is...

○ user behavior

6 Quotations:

2:10 ¶ 48 – 51 in P2.docx

Codes:

○ user behavior: because wants GPT to assist understanding

Content:

> Interviewer: But have you ever tried, because I see you've written a rather comprehensive instruction here, and then have you ever tried directly feeding both the code and the instruction to GPT, letting it generate the desired results based on your comprehensive interaction?
> P2: Never typed code.
> Interviewer: You ask bit by bit, right?
> P2: Right, because during the auditing process, I also got to know this project bit by bit. Of course, if it's like that, actually my expectation for its use is that it can assist me, but the ultimate result still depends on my understanding and knowledge of this project. I don't want it to be like a vulnerability scanner where I just throw the project at you, not even knowing what's inside the project, and then you just give me feedback on vulnerabilities. My usage scenario may be slightly different.

2:13 ¶ 58 – 64 in P2.docx

Codes:

○ user behavior: improved performance with superficial vulnerability

Content:

> Interviewer: Do you think your efficiency has improved after using GPT?
> P2: Of course there is improvement.
> Interviewer: Approximately how many percentage points has it increased?
> P2: Let me think. If GPT can get a good understanding of a project of any size in about a day, it would probably take me 3 to 4 days.
> Interviewer: Look for errors. Do you think that, for example, within the same amount of time, the number of errors you've found has increased? Has your performance efficiency improved?
> P2: Has efficiency increased? Yes, it has.
> Interviewer: For example, you used to be able to find only one bug a day, but now you might be able to find a dozen or so in a day.

4:3 ¶ 5 – 6 in P4.docx

Codes:

○ user behavior: ask GPT to describe a code to 1) check if they has the same understanding, 2) then ask it return some security questions 3) or ask GPT to verify the security question. (Guide or Verify)

Content:

> Interviewer: Okay, then could you briefly describe how you used ChatGPT to assist you throughout the entire process?

P4: When using GPT, generally you first ask it to describe something, then paste a code snippet to it. You may check whether GPT's understanding of the code is consistent with yours, or you can directly ask ChatGPT to understand it and interpret based on its understanding. First, you understand the information GPT provides about the code snippet you gave it, then ask it for some security issues. After getting the security issues, you can verify them, or you can directly ask GPT to provide some security verifications. Then check whether your own stored security knowledge is consistent with GPT's, or if there are aspects GPT considered that you didn't, you can conduct some verifications. Or, if GPT's considerations are all wrong, they can still give you some hints, and then you can look for relevant information based on them.

4:4 ¶ 7 in P4.docx

Codes:

○ user behavior: improved performance with superficial vulnerability

Content:

Another way is guided questioning. That is, you may want to conduct some tests, but you don't have any ideas about how to do them. You can tell GPT what problems you have at this time, and guide it to help you describe how to identify the problem. Then you can make some corresponding test preparations. When you quickly review some code in the future, you can adjust it to GPT, and it can directly help you identify the problems you previously provided. That's roughly how it works.

4:8 ¶ 16 – 19 in P4.docx

Codes:

○ Challenge: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate ○ user behavior: has some basic knowledge but don't have lists

Content:

Interviewer: Since you have approximately two and a half years of experience in code auditing, you have a relatively systematic and mature knowledge system of your own. Then, when you conduct manual comparisons, for example, you can directly refer to this knowledge system to make a comparison.
P4: Regarding auditing, apart from those scattered security knowledge, when you talk about precision, such as reentrancy, but in fact, specifically, you still need to combine it with the code. For which projects, which reentrancy might lead to some stop-loss issues, and which reentrancy might lead to some calculation issues. For example, read-only reentrancy, its reentrancy logic is different, and the issues are also different. When these accumulate, if you are very familiar with the code and then see similar code snippets, you may consider the corresponding issues. These can be regarded as personal accumulation. However, GPT can think of more code snippets, but it is not very effective when you try to verify them.
Interviewer: Do you make some comparisons? For example, if you have a list, when you look at these codes, do you search for some comparisons, taking some vulnerabilities as your priority to look for? Or do you still rely on your intuition?
P4: With basic code knowledge, first you need to understand that some characteristics of a project, such as price manipulation in a recent reported project, exist. Based on the project, there are some vulnerabilities in its characteristics, as well as some common vulnerabilities and some permission-related vulnerabilities. These actually fall into several levels, including common code-level, no, language-level vulnerabilities, such as the simplest overflow issues, overflow issues before 0.8. You may need to take these into account, based on the version of the language it uses and other factors. There are language-level, logical-level, and common permission-level vulnerabilities, which can be divided into several major categories. When auditing a project, you can apply these accordingly. It can be understood as being similar to the self-study course mentioned earlier, which is actually based on your existing knowledge to see which items need to be applied to which projects. However, when it comes to the actual audit, you still need to understand the project and what modifications it has made. If these modifications are at a new level, you need to understand this and then consider some other issues.

4:10 ¶ 22 – 23 in P4.docx

Codes:

○ user behavior: mostly use GPT to support understanding

Content:

> Interviewer: Then there's this question, which is that when you usually use other software, apart from ChatGPT, what do you think are the most critical features that other software has but GPT doesn't, yet you still consider very important?
> P4: I haven't used other tools much; it seems like I haven't used them at all. In fact, I also use GPT relatively rarely because GPT is generally only used when working on complex projects. Some of its functions may take you a long time to understand and require a lot of effort. Then you can ask ChatGPT to help you interpret them so that you can quickly get started. However, when it comes to security issues, currently, unless you use a personal GPT to ask some security knowledge, which you can trust to some extent, for ordinary GPT, the information it provides is not very useful, mainly at the understanding level. And it is used relatively rarely in terms of assisting with security. As for other tools, their testing results are not very good, and I haven't used them much either.

○ user behavior: ask GPT to describe a code to 1) check if they has the same understanding, 2) then ask it return some security questions 3) or ask GPT to verify the security question. (Guide or Verify)

1 Quotations:

4:3 ¶ 5 – 6 in P4.docx

Codes:

○ user behavior: ask GPT to describe a code to 1) check if they has the same understanding, 2) then ask it return some security questions 3) or ask GPT to verify the security question. (Guide or Verify)

Content:

> Interviewer: Okay, then could you briefly describe how you used ChatGPT to assist you throughout the entire process?
> P4: When using GPT, generally you first ask it to describe something, then paste a code snippet to it. You may check whether GPT's understanding of the code is consistent with yours, or you can directly ask ChatGPT to understand it and interpret based on its understanding. First, you understand the information GPT provides about the code snippet you gave it, then ask it for some security issues. After getting the security issues, you can verify them, or you can directly ask GPT to provide some security verifications. Then check whether your own stored security knowledge is consistent with GPT's, or if there are aspects GPT considered that you didn't, you can conduct some verifications. Or, if GPT's considerations are all wrong, they can still give you some hints, and then you can look for relevant information based on them.

○ user behavior: because wants GPT to assist understanding

1 Quotations:

2:10 ¶ 48 – 51 in P2.docx

Codes:

○ user behavior: because wants GPT to assist understanding

Content:

> Interviewer: But have you ever tried, because I see you've written a rather comprehensive instruction here, and then have you ever tried directly feeding both the code and the instruction to GPT, letting it generate the desired results based on your comprehensive interaction?
> P2: Never typed code.
> Interviewer: You ask bit by bit, right?
> P2: Right, because during the auditing process, I also got to know this project bit by bit. Of course, if it's like that, actually my expectation for its use is that it can assist me, but the ultimate result still depends on my understanding and knowledge of this project. I don't want it to be like a vulnerability scanner where I just throw the project at you, not even knowing what's inside the project, and then you just give me feedback on vulnerabilities. My usage scenario may be slightly different.

○ user behavior: has some basic knowledge but don't have lists

1 Quotations:

4:8 ¶ 16 – 19 in P4.docx

Codes:

○ Challenge: GPT: vast data and similar codes but not accurate vs human: limited experience but accurate ○ user behavior: has some basic knowledge but don't have lists

Content:

> Interviewer: Since you have approximately two and a half years of experience in code auditing, you have a relatively systematic and mature knowledge system of your own. Then, when you conduct manual comparisons, for example, you can directly refer to this knowledge system to make a comparison.
> P4: Regarding auditing, apart from those scattered security knowledge, when you talk about precision, such as reentrancy, but in fact, specifically, you still need to combine it with the code. For which projects, which reentrancy might lead to some stop-loss issues, and which reentrancy might lead to some calculation issues. For example, read-only reentrancy, its reentrancy logic is different, and the issues are also different. When these accumulate, if you are very familiar with the code and then see similar code snippets, you may consider the corresponding issues. These can be regarded as personal accumulation. However, GPT can think of more code snippets, but it is not very effective when you try to verify them.
> Interviewer: Do you make some comparisons? For example, if you have a list, when you look at these codes, do you search for some comparisons, taking some vulnerabilities as your priority to look for? Or do you still rely on your intuition?
> P4: With basic code knowledge, first you need to understand that some characteristics of a project, such as price manipulation in a recent reported project, exist. Based on the project, there are some vulnerabilities in its characteristics, as well as some common vulnerabilities and some permission-related vulnerabilities. These actually fall into several levels, including common code-level, no, language-level vulnerabilities, such as the simplest overflow issues, overflow issues before 0.8. You may need to take these into account, based on the version of the language it uses and other factors. There are language-level, logical-level, and common permission-level vulnerabilities, which can be divided into several major categories. When auditing a project, you can apply these accordingly. It can be understood as being similar to the self-study course mentioned earlier, which is actually based on your existing knowledge to see which items need to be applied to which projects. However, when it comes to the actual audit, you still need to understand the project and what modifications it has made. If these modifications are at a new level, you need to understand this and then consider some other issues.

○ user behavior: improved performance with superficial vulnerability

2 Quotations:

2:13 ¶ 58 – 64 in P2.docx

Codes:

○ user behavior: improved performance with superficial vulnerability

Content:

> Interviewer: Do you think your efficiency has improved after using GPT?
> P2: Of course there is improvement.
> Interviewer: Approximately how many percentage points has it increased?
> P2: Let me think. If GPT can get a good understanding of a project of any size in about a day, it would probably take me 3 to 4 days.
> Interviewer: Look for errors. Do you think that, for example, within the same amount of time, the number of errors you've found has increased? Has your performance efficiency improved?
> P2: Has efficiency increased? Yes, it has.
> Interviewer: For example, you used to be able to find only one bug a day, but now you might be able to find a dozen or so in a day.

4:4 ¶ 7 in P4.docx

Codes:

○ user behavior: improved performance with superficial vulnerability

Content:

> Another way is guided questioning. That is, you may want to conduct some tests, but you don't have any ideas about how to do them. You can tell GPT what problems you have at this time, and guide it to help you describe how to identify the problem. Then you can make some corresponding test preparations. When you quickly review some code in the future, you can adjust it to GPT, and it can directly help you identify the problems you previously provided. That's roughly how it works.

○ user behavior: mostly use GPT to support understanding

1 Quotations:

4:10 ¶ 22 – 23 in P4.docx

Codes:

○ user behavior: mostly use GPT to support understanding

Content:

> Interviewer: Then there's this question, which is that when you usually use other software, apart from ChatGPT, what do you think are the most critical features that other software has but GPT doesn't, yet you still consider very important?
> P4: I haven't used other tools much; it seems like I haven't used them at all. In fact, I also use GPT relatively rarely because GPT is generally only used when working on complex projects. Some of its functions may take you a long time to understand and require a lot of effort. Then you can ask ChatGPT to help you interpret them so that you can quickly get started. However, when it comes to security issues, currently, unless you use a personal GPT to ask some security knowledge, which you can trust to some extent, for ordinary GPT, the information it provides is not very useful, mainly at the understanding level. And it is used relatively rarely in terms of assisting with security. As for other tools, their testing results are not very good, and I haven't used them much either.

○ Validating

4 Quotations:

1:17 ¶ 38 – 42 in P1.docx

Codes:

○ Validating: manual work: should have the knowledge base and checklist to check if the vulnerability

Content:

> Interviewer: Let me recap. These three steps, planning, reasoning, and validating, basically constitute the three main steps when you conduct an audit. As for the previous steps, planning and reasoning, they should have been covered just now, and now only the validating stage is left.
> P1: When it comes to validating, there's one more thing missing. How do you say "validating"? You may actually notice that whether it's planning or reasoning, the most commonly used methods we have now are actually based on some Knowledge Base. Why do I say we must audit the so-called liquid function instead of auditing the balance off function above? Actually, it's also based on a piece of knowledge. This knowledge is that when, for example, only the owner appears, or in this scenario, I prefer to choose the business function that the user can call for liquidation for auditing, rather than using this audit. Well, this knowledge is also what I'm currently working on, which means it can also be transformed into the same data structure as the checklist just mentioned.
> Interviewer: Just like what the plugin generated just now, right?
> P1: Those are just for auxiliary purposes; in reality, one still has to acquire genuine knowledge, because the plugin only provides you with a list. What exactly you need to review still depends on your knowledge. So why do I mention the planning list? It's because during the validating process, in my understanding, there is also a so-called list and a Knowledge Base. The validating list consists of different vulnerabilities, and it involves finding ways to identify false positives. In simple terms, the purpose of validating is to reduce false positives. The way to reduce false positives is to verify them. There are roughly several methods in this regard, such as the first checklist approach,

> the so-called forced questioning or deception approach, and the third is the thinking chain approach. Currently, validating is also divided into several approaches, and the main one now is the symbolic execution approach, which involves automation. Based on the description of the vulnerability returned by reasoning, an exploit script for the vulnerability is automatically generated.
> Interviewer: Automatically generate vulnerability exploitation scripts.

1:18 ¶ 43 in P1.docx

Codes:

○ Validating: more specifically describe the three methods

Content:

> P1: The first approach is symbolic execution, which is roughly something like this. The second approach is a manual one, also using the so-called validating list to verify the corresponding vulnerability by asking questions to ChatGPT. That is, you can only ask it what you think the key to this vulnerability is, which variables have not been checked, and ChatGPT will reply with some information. Next, I will provide the entire contract to ChatGPT for you to check whether the variables in this contract violate the corresponding checks and restrictions.

2:8 ¶ 38 – 43 in P2.docx

Codes:

○ Validating: manual work: GPT results need to be manually checked

Content:

> Interviewer: Okay, for example, if you've already found this error, what would you do? Would you verify it?
> P2: Does verification refer to writing code?
> Interviewer: The question is whether it is actually an error. That is, GPT doesn't just return a result to you, and then you have to confirm whether the result it returns has any errors. You may even need to write some attack scripts or something. So, how do you use GPT in this situation?
> P2: At this point, it mostly relies on human effort, and it relies less on GPT. When it raises a question, it basically requires manual review.
> Interviewer: Will you use it when you write the report?
> P2: There is no report; everything is written manually.

5:4 ¶ 7 – 16 in P5.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: 1. auditing based on domain knowledge; 2. deceive; 3. original conversation with CoT ○ Validating: manual work: GPT results need to be manually checked

Content:

> Interviewer: Assistance with output format and content. Does it refer to the format of the final report?
> P5: Right. You can give him a template, then paste the code and describe it, and then he will give you a specific template and specific output based on the template.
> Interviewer: Just now you mentioned that you look for vulnerabilities, which means these vulnerabilities come in different types, right? Then these different types of vulnerabilities are, so to speak, some you assign to GPT to look for, while you look for others yourself. Let me rephrase my question. You just mentioned different types of issues, such as some being variables and some being function functionalities. So which ones do you think you can rely on GPT to help you find, and which ones do you prefer to find on your own?
> P5: For this, I haven't fully studied and understood how to use ChatGPT to handle these things. I can only give it a good prompt, then provide it with the code and ask it to output some content. But when it comes to having it do specific tasks like sorting out the state changes of its variables, I may not have attempted that yet.
> Interviewer: So when you use ChatGPT now, it's just equivalent to a single-turn conversation, right? For example, you input code to it, then let it directly give you an output, and then you'll ask step by step like this to guide it to find some bugs.
> P5: Right.

> Interviewer: Are you referring to the second type, or the single-round dialogue?
> P5: I'm the kind of person who, when given the code, either can switch it or, if you're not satisfied with the result, can retry. I'll just keep retrying to see if I can get something effective.
> Interviewer: You're not saying that, for example, after you give it something, it generates a result, and then after you look at this result, you continue to ask further in-depth questions based on this result, right?
> P5: Some will. If you can tell at a glance that it's a false alarm, there's no need to let it retry; just let it provide a different answer and don't dig deeper. If it's ambiguous but somewhat relevant, you may need to ask. Right, but basically, I only make a judgment when it's relatively certain. If it's ambiguous, I'll ask a bit. If the answer is still rather vague after asking, I'll directly retry and let it take a different path.

○ Validating: manual work: GPT results need to be manually checked

2 Quotations:

2:8 ¶ 38 – 43 in P2.docx

Codes:

○ Validating: manual work: GPT results need to be manually checked

Content:

> Interviewer: Okay, for example, if you've already found this error, what would you do? Would you verify it?
> P2: Does verification refer to writing code?
> Interviewer: The question is whether it is actually an error. That is, GPT doesn't just return a result to you, and then you have to confirm whether the result it returns has any errors. You may even need to write some attack scripts or something. So, how do you use GPT in this situation?
> P2: At this point, it mostly relies on human effort, and it relies less on GPT. When it raises a question, it basically requires manual review.
> Interviewer: Will you use it when you write the report?
> P2: There is no report; everything is written manually.

5:4 ¶ 7 – 16 in P5.docx

Codes:

○ Reasoning: ChatGPT prompting strategy: 1. auditing based on domain knowledge; 2. deceive; 3. original conversation with CoT ○ Validating: manual work: GPT results need to be manually checked

Content:

> Interviewer: Assistance with output format and content. Does it refer to the format of the final report?
> P5: Right. You can give him a template, then paste the code and describe it, and then he will give you a specific template and specific output based on the template.
> Interviewer: Just now you mentioned that you look for vulnerabilities, which means these vulnerabilities come in different types, right? Then these different types of vulnerabilities are, so to speak, some you assign to GPT to look for, while you look for others yourself. Let me rephrase my question. You just mentioned different types of issues, such as some being variables and some being function functionalities. So which ones do you think you can rely on GPT to help you find, and which ones do you prefer to find on your own?
> P5: For this, I haven't fully studied and understood how to use ChatGPT to handle these things. I can only give it a good prompt, then provide it with the code and ask it to output some content. But when it comes to having it do specific tasks like sorting out the state changes of its variables, I may not have attempted that yet.
> Interviewer: So when you use ChatGPT now, it's just equivalent to a single-turn conversation, right? For example, you input code to it, then let it directly give you an output, and then you'll ask step by step like this to guide it to find some bugs.
> P5: Right.
> Interviewer: Are you referring to the second type, or the single-round dialogue?
> P5: I'm the kind of person who, when given the code, either can switch it or, if you're not satisfied with the result, can retry. I'll just keep retrying to see if I can get something effective.

> Interviewer: You're not saying that, for example, after you give it something, it generates a result, and then after you look at this result, you continue to ask further in-depth questions based on this result, right?
> P5: Some will. If you can tell at a glance that it's a false alarm, there's no need to let it retry; just let it provide a different answer and don't dig deeper. If it's ambiguous but somewhat relevant, you may need to ask. Right, but basically, I only make a judgment when it's relatively certain. If it's ambiguous, I'll ask a bit. If the answer is still rather vague after asking, I'll directly retry and let it take a different path.

○ Validating: manual work: should have the knowledge base and checklist to check if the vulnerability

1 Quotations:

1:17 ¶ 38 – 42 in P1.docx

Codes:

○ Validating: manual work: should have the knowledge base and checklist to check if the vulnerability

Content:

> Interviewer: Let me recap. These three steps, planning, reasoning, and validating, basically constitute the three main steps when you conduct an audit. As for the previous steps, planning and reasoning, they should have been covered just now, and now only the validating stage is left.
> P1: When it comes to validating, there's one more thing missing. How do you say "validating"? You may actually notice that whether it's planning or reasoning, the most commonly used methods we have now are actually based on some Knowledge Base. Why do I say we must audit the so-called liquid function instead of auditing the balance off function above? Actually, it's also based on a piece of knowledge. This knowledge is that when, for example, only the owner appears, or in this scenario, I prefer to choose the business function that the user can call for liquidation for auditing, rather than using this audit. Well, this knowledge is also what I'm currently working on, which means it can also be transformed into the same data structure as the checklist just mentioned.
> Interviewer: Just like what the plugin generated just now, right?
> P1: Those are just for auxiliary purposes; in reality, one still has to acquire genuine knowledge, because the plugin only provides you with a list. What exactly you need to review still depends on your knowledge. So why do I mention the planning list? It's because during the validating process, in my understanding, there is also a so-called list and a Knowledge Base. The validating list consists of different vulnerabilities, and it involves finding ways to identify false positives. In simple terms, the purpose of validating is to reduce false positives. The way to reduce false positives is to verify them. There are roughly several methods in this regard, such as the first checklist approach, the so-called forced questioning or deception approach, and the third is the thinking chain approach. Currently, validating is also divided into several approaches, and the main one now is the symbolic execution approach, which involves automation. Based on the description of the vulnerability returned by reasoning, an exploit script for the vulnerability is automatically generated.
> Interviewer: Automatically generate vulnerability exploitation scripts.

○ Validating: more specifically describe the three methods

1 Quotations:

1:18 ¶ 43 in P1.docx

Codes:

○ Validating: more specifically describe the three methods

Content:

> P1: The first approach is symbolic execution, which is roughly something like this. The second approach is a manual one, also using the so-called validating list to verify the corresponding vulnerability by asking questions to ChatGPT. That is, you can only ask it what you think the key to this vulnerability is, which variables have not been checked, and ChatGPT will reply with some information. Next, I will provide the entire contract to ChatGPT for you to check whether the variables in this contract violate the corresponding checks and restrictions.