Contents

[Chapter 4: Analysis of Results and Discussion 2](#_Toc207473051)

[4.1 Introduction of the Analysis and Discussion 2](#_Toc207473052)

[4.2 Presentation of Findings 2](#_Toc207473053)

[4.2.1 Test Case 1: Broad Prompt 2](#_Toc207473054)

[4.2.2 Test case 2: Mid prompt 4](#_Toc207473055)

[4.2.3 Test Case 3: Specific Prompt 5](#_Toc207473056)

[4.3 Cross-Reference Analysis with Other Studies 6](#_Toc207473057)

[4.4 Overall Test Cases Assessment 8](#_Toc207473058)

[References 10](#_Toc207473059)

# Chapter 4: Analysis of Results and Discussion

## 4.1 Introduction of the Analysis and Discussion

This research hypothesised that large language models (LLMs) could be systematically evaluated for their ability to detect security misconfigurations in Cisco IOS configurations when assessed against the CIS Benchmarks. By employing a structured evaluation framework built on controlled datasets and carefully engineered prompts, the study aimed to measure both the accuracy and reliability of GPT-4o in identifying benchmark-aligned errors as well as common configuration mistakes. The primary objective was to determine whether prompt engineering could be used to guide the models toward more consistent and accurate compliance assessments.

In addition, the research sought to analyse how different levels of prompt specificity, Broad, Mid and Specific and how it shaped the model’s performance, and to compare results across multiple protocol domains, including AAA, EIGRP, OSPF and RIP. This study also aimed to capture qualitative patterns in the model’s responses, such as recurring misclassifications or hallucinations, to better understand their interpretative strengths and limitations. In the following sections, the analysis will present the result of these evaluations, compare model performance across prompt types and protocols, and critically discuss the implications of these findings for the integration of LLMs into professional network auditing and compliance workflows.

## 4.2 Presentation of Findings

### 4.2.1 Test Case 1: Broad Prompt

The Broad prompt was designed as the baseline test, requiring the model to review configurations without explicit reference to the CIS Benchmarks. Its purpose was to measure GPT’s ability to apply general networking knowledge when identifying misconfigurations and Mistype errors. The results indicated limited effectiveness in detecting benchmark aligned issues, with stronger but still inconsistent performance in identifying simple typographical errors.

Chart 4.1: Broad Prompt Results

Chart 4.1 summarises the results across all four protocols. Out of 120 benchmark-related errors, only 34 were detected, resulting in a total PP Score of 28%. Accuracy varied significantly by protocol. OSPF achieved the highest PP Score of 57%, followed by AAA at 40%, while RIP and especially EIGRP performed poorly, with scores of 10% and 3% respectively. By contrast, Mistype errors were detected resulting in a total PP Score of 75%. Accuracy varied between protocol with EIGRP, OSPF and RIP all having PP Scores of 80% while AAA scored a PP Score of 60%. This indicated that GPT was more reliable at identifying basic syntactic mistakes than at enforcing benchmark compliance.

In terms of qualitative patterns, the model frequently produced short and list-like responses rather than detailed checklists. While it often flagged certain recurring issues, such as incorrectly labelling no shutdown commands as misconfigurations, it consistently failed to detect critical requirements in protocols such as EIGRP, including missing “ip authentication key-chain eigrp” and related commands. For OSPF, the model intermittently detected the absence of “ip ospf message-digest-key”, though not with consistency. Another limitation was the tendency to omit a conclusive statement at the end of some outputs, despite being instructed to explicitly state whether the configuration was secure.

Overall, the Broad prompt demonstrated the weakness of unguided prompting. While capable of identifying obvious typos and occasional misconfigurations, its detection of benchmark-aligned errors was inconsistent and unreliable across protocols. These findings establish a baseline for comparison with the Mid and Specific prompts, where addition guidance was introduced.

### 4.2.2 Test case 2: Mid prompt

The Mid prompt introduced explicit reference to the CIS Benchmarks but did not provide excerpts of the standards themselves. Instead, the model was instructed to review the configurations “according to CIS Benchmarks”, requiring them to rely on any embedded knowledge of the framework gained during training. This test case therefore measured whether mentioning the CIS Benchmarks within the prompt alone improved detection accuracy compared to a purely unguided assessment.

Chart 4.2: Mid Prompt Results

The results showed moderate gains in benchmark-related error detection. Out of 120 misconfigurations, 39 were identified, resulting in a total PP Score of 33%. Accuracy differed by protocol, with OSPF achieving the highest PP Score of 50% followed by AAA at 37% and EIGRP at 33%. RIP remained the weakest domain with only 17% of errors detected. Mistype errors were handled more effectively than in the Broad test, with a PP Score of 85% across all protocols. Protocol-specific mistype scores ranged from 80% in AAA, OSPF and RIP to 100% in EIGRP, demonstrating strong consistency in identifying basic syntactic issues. A per-protocol breakdown of PP Scores is presented in Chart 4.2.

Qualitative inspection of the outputs revealed that responses were generally more detailed than in the Broad test, though still presented in a list-like style. Importantly, the answers appeared more compliance-oriented, with the model often attempting to frame its findings against implicit CIS rules. Nonetheless, recurring detection gaps were observed across all domains. For example, in AAA the absence of “aaa accounting system” was sometimes detected but not consistently, in EIGRP, missing “key” statements were rarely identified, in OSPF, the omission of “router ospf” was occasionally overlooked, and in RIP, the lack of “ip rip authentication mode md5” was frequently missed. Additionally, the model still did not always provide a final binary statement of whether the configuration was secure, despite being explicitly instructed to do so.

### 4.2.3 Test Case 3: Specific Prompt

The Specific prompt represented the most constrained test case, limiting the model’s evaluation to a single protocol and attaching the corresponding section of the CIS Benchmark for reference. By providing explicit standards alongside the configuration, this test case measured the model’s ability to apply authoritative compliance rules within a defined scope.

Chart 4.3: Specific Prompt Results

The results demonstrated considerably higher accuracy than the previous test cases. Out of 120 benchmark-related errors, 73 were detected, giving a total PP Score of 61%. Protocol-level performance varied, with OSPF achieving the highest accuracy of 77%, followed by EIGRP at 73%, AAA at 50% and RIP at 43%. Mistype detection was strong overall, with a combined PP Score of 85%, though individual protocol scores ranged from 60% in RIP to 100% in both EIGRP and OSPF. A per-protocol breakdown of PP Scores is illustrated in Chart 4.3

Closer analysis revealed that certain critical errors were consistently detected across all runs. For example, “aaa authentication enable” was always flagged in AAA, “ip authentication key-chain eigrp” in EIGRP, “router ospf” in OSPF and “key-string” in RIP. However, other key requirements were occasionally missed. In AAA, the absence of “aaa authorization exec” was not always identified, in OSPF, some instances of missing “ip ospf message-digest-keyt md5” were overlooked, and in RIP, the “router rip” statement was occasionally missed. These inconsistencies highlight limitations in strict rule adherence, even when benchmark excerpts were available.

Qualitatively, the outputs were notably more structured and checklist-like than in the other test cases, with longer and more precise explanations. Responses were more direct in referencing benchmark rules, and the model adhered closely to the requirement of concluding with a Yes/No statement of security compliance. This produced outputs that were more consistent with the expectations of a professional security audit.

## 4.3 Cross-Reference Analysis with Other Studies

The findings of this study align closely with those reported by Sare and Debono [21], particularly in the effect that benchmark guidance has on LLM performance. In this research, the Broad prompt produced limited accuracy, with an overall PP Score of 28% for misconfiguration detection, whereas accuracy increased substantially when benchmark excerpts were introduced in the Specific case, reaching 61%. A similar trend was documented by Sare and Debono, who reported that “GPT-4’s response accuracy rate is 75%. Compared to GPT-4’s performance in Table 4.7 from Test case 2 without CIS benchmark document provided, which has an accuracy rate of 40.8%, this shows an improvement of 34.2%” [21]. Her work also demonstrated that baseline zero-shot prompting without CIS reference produced accuracy rated of only 40.8% for GPT-4 and 26.3% for GPT-3.5, highlighting the same challenge observed in this study, where unguided prompts struggled to consistently detect benchmark-aligned errors. Furthermore, Sare and Debono observed that when prompts explicitly referenced CIS benchmarks, “There is a 10% improvement in the ‘Cross-reference with CIS Benchmarks’ category” [21], which is consistent with the improvements noted here when using benchmark-guided prompts. Together, these parallels reinforce that while LLMs demonstrate some capacity for detecting vulnerabilities under general prompts, their accuracy is markedly enhanced when grounded in explicit compliance standards.

In addition to Sare’s study, the results reported by Cao et al. [16] offer a closely related pattern regarding the effect of prompt design on LLM performance. They show that a basic, minimally guided prompt led ChatGPT to miss a meaningful portion of true faults and sometimes emphasize non-critical issues, whereas providing clearer task intention and context substantially improved performance and shifted the model toward more functionally relevant detections. These observations are constant with this dissertation’s results, unguided, broad prompts tended to produce shorter, list-like outputs and occasional false positives, while benchmark-guided, protocol-scoped prompting yielded more structured compliance reasoning and higher accuracy in identifying materially important configuration issues. Cao et al. further note that supplying explicit intent and objectives in the prompt improves fault localization, and that interactive follow-ups can raise repair success yet remain vulnerable to misinterpretation [16], like this study’s finding that even with authoritative CIS excerpts attached, certain critical omissions could be overlooked.

The variability and inconsistencies observed in this study are mirrored in the findings by Sobania et al. [4]. In the evaluation stages, GPT models often failed to consistently detect critical errors, with some misconfigurations like missing OSPF authentication keys overlooked in multiple test cases despite their clear inclusion in the CIS Benchmarks. Similarly, the bug-fixing study noted that “benchmark problems are often only solved in one or two runs … So ChatGPT seems to have a relatively high variance when fixing bugs” [4], indicating comparable instability in accuracy across domains. Both investigations also relied on strict evaluation criteria. While in this work outputs were judged manuals for correctness against benchmark requirements, the bug-fixing study emphasized that “we are very strict in our evaluation and consider only patches as correct if the bug … is actually identified and corrected” [4]. Furthermore, the improvement in accuracy observed in this dissertation when CIS benchmarks were explicitly provided parallels the performance gains seen when additional guidance was given in bug-fixing tasks, where “adding a hint of ChatGPT vastly improves its performance, with 31 out of 40 problems solved” [4]. These parallels reinforce that across distinct application areas (network configuration auditing and automatic program repair) the effectiveness of LLMs is closely tied to the specificity and clarity of guidance in the prompt.

## 4.4 Overall Test Cases Assessment

The evaluation of all three test cases demonstrates that the level of specificity within the prompt had a direct and measurable impact on model performance. The Broad case, which provided no reference to external standards, consistently underperformed, showing that unguided reasoning alone is insufficient for reliable compliance validation. By contract, the Mid and Specific prompts, which progressively added references to CIS Benchmarks and direct benchmark excerpts, achieved higher accuracy and produced more structured responses. This progression highlights that guiding an LLM with explicit and authoritative standards substantially improves its ability to detect configuration issues. Nevertheless, even with benchmark excerpts attached, the models demonstrated inconsistencies that prevent them from being considered fully dependable for automated compliance auditing.

When considered across all protocols, certain patterns became evident. OSPF emerged as the strongest area for misconfiguration detection, whereas EIGRP was consistently the weakest, with frequent failures to identify missing authentication commands. For mistypes, the models achieved their highest accuracy in the EIGRP domain, while AAA and RIP were the weakest, reflecting inconsistency in detecting simple human errors across different protocol contexts. Across all test cases, the models performed more strongly on Mistype errors than on benchmark-related misconfigurations, confirming that LLMs are more effective at spotting surface-level mistakes than deeply embedded compliance violations.

Recurring strengths and weaknesses were also observed in relation to individual commands. Commands such as “service password-encryption”, “aaa authorization exec”, “area authentication message-digest” and “ip rip authentication key-chain” were reliably identified across tests. In contrast, critical commands like “router rip”, “ip authentication key-chain eigrp”, “ip ospf message-digest-key md5” and “aaa accounting system” were inconsistently flagged, representing the most persistent blind spots in the evaluations. False positives further undermined performance, with the models frequently misclassifying the valid “no shutdown” command as an error. The quality of responses also varied with prompt specificity, becoming more checklist-like, structured and precise under the Specific prompt, whereas the Broad and Mid prompts often produced short, vague or incomplete outputs.

Overall, these findings reinforce the conclusion that while prompt specificity improves reliability, LLMs remain inconsistent in detecting benchmark-driven misconfigurations and often generate false positives. The requirement for the model to state whether a configuration was secure was followed more consistently under the Specific prompt, but remained inconsistent in earlier test cases. From an operational standpoint, the models cannot yet be recommended for use in production environments, as their output requires substantial human verification. Furthermore, limitations such as restricted context memory, sensitivity to prompt wording, and the inability to process full benchmark documents in a single pass presents significant obstacles. These factors highlight that while LLMs offer promise as an assistive tool in network security auditing, they are not yet mature enough to replace traditional compliance mechanisms or expert human review.

## References

[4] D. Sobania, M. Briesch, C. Hanna, and J. Petke, “An Analysis of the Automatic Bug Fixing Performance of ChatGPT,” in *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)*, May 2023, pp. 23–30. doi: [10.1109/APR59189.2023.00012](https://doi.org/10.1109/APR59189.2023.00012).

[16] J. Cao, M. Li, M. Wen, and S. Cheung, ‘A study on Prompt Design, Advantages and Limitations of ChatGPT for Deep Learning Program Repair’, Apr. 17, 2023, *arXiv*: arXiv:2304.08191. doi: [10.48550/arXiv.2304.08191](https://doi.org/10.48550/arXiv.2304.08191).

[21] A. Sare and D. Debono, “The Dual-Edged Sword: The Impact of Large Language Models in Network Infrastructure Security,” Institute of Information and Communication Technology, Malta College of Arts, Science and Technology, 2025. Accessed: Aug. 30, 2025. [Online]. Available: <https://www.scitepress.org/PersonProfile.aspx?PersonAccountID=SPMqmL3tsi8=&t=1>

‌