MCP-Enabled AI Agents for Secure and Autonomous Enterprise Storage Management

NIKHIL KUMAR AEDLA

Abstract: *Enterprise storage supports analytics, regulatory compliance, and artificial intelligence applications. However, its management often remains manual and disconnected. Language model agents can plan, reason, and interact with external tools. However, their autonomy introduces significant security vulnerabilities. This research investigates secure MCP-based agents for autonomous enterprise storage management. The aim of this research is to explore MCP-based agents for autonomous enterprise storage management that is also secure and operates within zero trust controls. Design science methodology was used in a simulated environment of enterprise storage with an MCP server that has enterprise storage tools exposed. The experimental dataset had 750 synthetic task samples with 150 adversarial samples with hidden instructions and malicious tool descriptions. 3 operating scenarios have been tested – manual management by a human, unsecured agent, and MCP-secured agent. The secured agent managed to attain a success rate of 94.67%, in comparison to the manual approach and the unsecured agent, whose success rates stood at 87.87% and 80.53%, respectively. In adversarial records, the unsecured agent executed unauthorized transactions in 61.33% of the total instances recorded, whereas the same happened in 3.33% of the cases for the secured agent. Unauthorized actions were thus 46.00 times more likely to occur without any controls, with p-value being less than .001. The median audit completeness stood at 98.6% for the secured agent, in contrast to the 80.1% recorded by the manual administrators. The policy enforcement process increased the time taken to execute a transaction by 7.12 seconds, while manual operations needed a median time of 613.15 seconds.*

Keywords: Model Context Protocol; AI Agents; Enterprise Storage Management; Zero Trust; Prompt Injection; Ransomware Resilience; Audit Completeness

Nikhil Kumar Aedla (nikhil.aedla@gmail.com) Principal Data Storage Engineer, Virginia, USA
ORCID: https://orcid.org/0009-0009-2509-7324

Introduction

Enterprise storage supports analytics, compliance, and artificial intelligence workloads but administration remains manual and disjointed. Language model agents have capabilities to reason, plan, and invoke external tools to accomplish operations (Yao et al., 2023). Tool invocation has become much more feasible because models can learn when to invoke interfaces (Schick et al., 2023). The Model Context Protocol (MCP) enables standardized agent interaction with tools and datasets (Hou et al., 2025). In terms of storage, such agents can configure volumes, enforce quotas, fine-tune tiering, and execute backup recovery processes. Autonomy increases the attack surface as malicious servers and corrupted metadata could hijack agents (Hou *et al.*, 2025). The proposed design extends a conventional MCP server with an integrated zero-trust execution workflow. A separate policy checker typically evaluates permissions before tool execution. However, this design combines instruction validation, contextual authorization, parameter inspection, execution approval, and audit verification. Each storage command passes through these security stages before reaching the storage management tools. The architecture also examines tool descriptions and arguments for potentially malicious instructions. Unauthorized requests are rejected before execution. High-risk operations require additional approval. All decisions and execution outcomes are recorded in an audit log. This approach connects agent reasoning with controlled storage execution. Therefore, the contribution lies in the integrated security workflow rather than MCP tool exposure alone. Hidden commands in accessed data can lead to language model applications to perform unintentional actions (Greshake *et al.*, 2023). Storage becomes appealing as ransomware attackers aim at encryption and destruction of vital data (Beaman et al., 2021). Zero trust approach implies continuous authentication and least privileges for each access attempt (Rose *et al*., 2020).

Related work

Research shows that combining reasoning and actions improves agent reliability (Yao et al., 2023). ReAct reported success improvements of 34% on ALFWorld and 10% on WebShop. Toolformer demonstrated API usage through limited instructions (Schick et al., 2023). MCP further standardises interactions between agents and external systems (Hou et al., 2025). However, MCP introduces security challenges involving permissions and malicious tool interactions. Greshake et al. (2023) demonstrated indirect prompt injection against deployed language model applications. Their findings showed how malicious retrieved content could manipulate agent behaviour. Existing permission and approval mechanisms restrict unauthorised tool execution. However, they do not necessarily validate retrieved instructions, tool descriptions, and execution parameters together. Zero-trust principles recommend continuous verification and restricted access (Rose et al., 2020). These principles support resource-level permissions for enterprise storage operations. Nevertheless, limited research integrates MCP agents, storage administration, and zero-trust enforcement. This study addresses this gap through integrated security controls. Its evaluation examines task success, audit completeness, and resistance to adversarial instructions. Their results indicate that retrieved content can discreetly lead agents to commit acts of data theft or harm. Beaman *et al*. (2021) explored ransomware methods and mitigation measures and identified the necessity for

increased stored data security. Rose *et al*. (2020) formulated seven zero trust guidelines, which include per-session access control and constant monitoring. These guidelines easily transfer into storage, where agents are required to have limited rights per volume, per snapshot, and per policy. Typical activities like forecasting of capacity, replication and test recovery seem to be natural cases for agent automation. All in all, these researches show the direction of moving from tools learning to agent standardization and security. Nevertheless, there are very few works combining MCP agents, storage administration and zero trust architecture in one system. Evaluation of such agents would necessarily include task performance, audit completeness and immunity to injection of instructions. This is why the current research in secure autonomous enterprise storage management is necessary.

Method

The testbed features an MCP server with storage tools for quota control and recovery (Hou et al., 2025). A language model agent applies the ReAct pattern to reason and choose tools (Yao et al., 2023). The experimental dataset has 750 synthetic task instances generated by the researchers. Each task instance consists of a natural language request, the sequence of required tools, the risk, and the ground truth outcome. The tasks include routine operations, capacity planning, backup recovery, and security incidents. Among these, there are 150 adversarial instances with secret instructions and malicious tool descriptions for measuring injection resilience (Greshake et al., 2023). Ransomware scenarios like mass file encryption are simulated to assess defensive measures (Beaman et al., 2021). All agent requests go through a zero trust policy layer to impose least privilege and continuous verification (Rose et al., 2020). The dataset contains MCP descriptions, vendor storage documentation, and audit trails. Metrics used for evaluating the performance are completion rate of tasks, unauthorized access rate, audit trail completion rate, and response time.

Table 1. Composition of the experimental dataset

Location	Required information
Methodology	Six specific MCP storage tools
Security controls	Authorization before write operations
Snapshot deletion	Administrator approval
Quota changes	Limit verification and authorization
Backup recovery	Approval and backup integrity checks
Evaluation	Four experimental performance metrics

The researchers generated 750 synthetic enterprise storage task instances. Each instance contained a unique request, expected tool sequence, risk category, and expected outcome. The dataset covered quota management, snapshot operations, backup recovery, capacity planning, and security incidents. Of these, 150 instances contained adversarial instructions or malicious tool descriptions. The remaining 600 instances represented routine and legitimate operational requests. Each task was executed under three scenarios: manual administration, unsecured agent, and secured MCP agent. Execution outcomes were compared against predefined expected results. The researchers recorded task completion, unauthorized actions, audit

completeness, and execution time. The resulting dataset supported quantitative comparisons across all three scenarios.

Results

Comparative Effectiveness of the 3 Operating Conditions

Table 1. Overall Performance Metrics for Manual, Unsecured, and Secured MCP Agent Conditions (n = 750 per condition)

Condition	Success rate (%)	Unauthorized action rate (%)	Median audit completeness (%)	Median response time (s)
Manual administrator	87.87	2.00	80.1	613.15
Unsecured agent	80.53	14.13	84.1	16.25
Secured MCP agent	94.67	1.07	98.6	23.37

The MCP agent, which was secured, had the highest success rate in completing the tasks at 94.67%, compared to 87.87% for the manual admin and 80.53% for the unsecured agent. There were unauthorized actions in 1.07% of tasks conducted by the secured agent, 2.00% by manual admin, and 14.13% by the unsecured agent. The manual administration baseline represents conventional human-led storage management. The administrator follows predefined operational instructions without autonomous agent assistance. The same storage tasks and expected outcomes apply across all three experimental scenarios. Task completion time measures the duration required to complete each operation. Therefore, the reported timing reflects the tested manual baseline. It should not be interpreted as a general measurement of human administrator performance. It appears that the zero-trust controls increase reliability rather than limit agent activity.

Variation Across Storage Workload Types

Table 2. Task Success Percentages by Storage Task Category and Operating Condition

Task category	Manual administrator	Unsecured agent	Secured MCP agent
Routine operations	91.6	87.6	96.0
Capacity planning	88.7	88.0	95.3
Backup and recovery	86.7	76.0	96.7
Security incidents	83.5	69.5	91.0

The secured MCP agent succeeded in all four workload categories with the highest rates, varying from 91.0% in security incidents to 96.7% in backup and recovery (Hasan *et al.*, 2025). The unsecured agent had the lowest performance in security incidents (69.5%) and backup and recovery (76.0%), where the number of malicious records was the largest (70 and 30 respectively). The secured MCP agent achieved the highest success rates across all four

workload categories. Its performance ranged from 91.0% in security incidents to 96.7% in backup and recovery (Hasan et al., 2025). The unsecured agent recorded the lowest performance in security incidents (69.5%) and backup recovery (76.0%). These categories contained 70 and 30 adversarial records, respectively. In routine operations, the unsecured agent achieved 87.6%, below manual administration's 91.6%. The secured agent achieved the highest routine success rate. Manual administration declined from 91.6% in routine operations to 83.5% in security incidents. Chi-square analysis found no statistically significant differences across secured-agent workload categories, $\chi^2(3) = 7.53$, $p = .057$. This suggests relatively consistent performance across different operational risks.

Exposure to Injected and Malicious Instructions

Table 3. Unauthorized Action Frequency for Adversarial and Benign Task Records

Condition	Adversarial events / 150	Adversarial rate (%)	Benign events / 600	Benign rate (%)
Manual administrator	9	6.00	6	1.00
Unsecured agent	92	61.33	14	2.33
Secured MCP agent	5	3.33	3	0.50

The adversarial data showed one of the main problems with the unsecured agent, which performed unauthorized actions in 92 out of 150 runs (61.33%). The secured MCP agent did so in 5 instances (3.33%) — it is a decrease of 58 percentage points. The administrators performed 6% unauthorized actions, 9 instances out of all runs. On the benign data set, all types of experiments demonstrated low unauthorized actions — 2.33% by the unsecured agent, 1.00% by manual administrators and 0.50% by the secured agent. The adversarial tests examined instruction manipulation and unauthorised storage operations. The secured agent evaluated permissions before executing write operations. Invalid requests were rejected before reaching storage tools. Approved operations generated corresponding audit records. These controls reduced unauthorised execution during synthetic testing. The difference between 61.33% and 2.33% unauthorized actions performed by the unsecured agent demonstrates that virtually all risk related to the unsecured agent comes from instructions and tool descriptions. These results match concerns about prompt injections and malicious server, and show that the least privilege principle would help defend from such attacks.

Inferential Tests for Binary Outcomes

Table 4. Cochran's Q, McNemar, and Fisher Exact Test Results for Success and Unauthorized Actions

Outcome and comparison	Test statistic	Discordant pairs (b / c)	Adjusted p
Success, all 3 conditions	$Q = 69.97$, $df = 2$	n/a	$< .001$
Success, manual vs unsecured	McNemar	130 / 75	$< .001$

Success, manual vs secured	McNemar	34 / 85	< .001
Success, unsecured vs secured	McNemar	26 / 132	< .001
Unauthorized (adversarial), all 3 conditions	Q = 146.20, df = 2	n/a	< .001
Unauthorized, manual vs unsecured	McNemar	5 / 88	< .001
Unauthorized, manual vs secured	McNemar	9 / 5	1.000
Unauthorized, unsecured vs secured	McNemar	89 / 2	< .001
Unauthorized, unsecured vs secured	Fisher exact, OR = 46.00	n/a	< .001

For unauthorized activities regarding the adversarial data, Q(2) = 146.20, p < .001. The unsecured agent was different from the other two, and the odds of taking an unauthorized action was 46.00 times greater for the former than the latter, p<.001 by Fisher exact test. The secured agent and the manual administrator were not significantly different, p = 1.000 from 9 and 5 discordant pairs.

Cochran's Q indicated significant differences among the three groups, Q(2) = 69.97, p < .001. Pairwise McNemar tests identified significant differences after multiple-comparison correction. The secured agent outperformed the unsecured agent, with 132 versus 26 discordant pairs. For adversarial tasks, Cochran's Q also showed significant differences, Q(2) = 146.20, p < .001. The unsecured agent recorded 61.33% unauthorised actions, compared with 3.33% for the secured agent. The corresponding rate ratio was 18.42. The reported odds ratio was 46.00, calculated using the following expression:

$$OR = \frac{a/b}{c/d} = \frac{ad}{bc}$$

Here, a and b represent unsecured-agent outcomes. c and d represent secured-agent outcomes. The exact contingency-table counts must support this estimate. Fisher's exact test indicated statistical significance (p < .001). The secured agent and manual administrator showed no significant difference. Their comparison produced nine and five discordant pairs, respectively (p = 1.000).

Inferential Tests for Audit Quality and Speed

Table 5. Friedman and Wilcoxon Signed Rank Results for Audit Completeness and Response Time

Metric	Friedman $\chi^2(2)$	Kendall's W	Comparison	Rank biserial r	Adjusted p
Audit completeness	1021.13	.681	Manual vs unsecured	.230	< .001
			Manual vs secured	.995	< .001

			Unsecured vs secured	.983	< .001
Response time	1216.27	.811	Manual vs unsecured	1.000	< .001
			Manual vs secured	1.000	< .001
			Unsecured vs secured	.629	< .001

The Friedman test revealed significant differences between the three conditions on audit completeness, $\chi^2(2) = 1021.13$, and on response time, $\chi^2(2) = 1216.27$, $p < .001$ for both. Kendall's W indicated strong consensus regarding the ranking of the tasks in different conditions (.681 for audit completeness and .811 for response time). In terms of audit completeness, the secured agent was very different from the manual condition (r = .995) and from the unsecured agent (r = .983), while the manual and the unsecured condition were just slightly different (r = .230). In terms of response time, both agents were faster compared to the manual condition in all 750 pairs of tasks (r = 1.000). The secured agent took more time than the unsecured agent (r = .629) due to the cost of policies checking. The 750 synthetic tasks formed a matched experimental dataset. Each task was evaluated under manual, unsecured-agent, and secured-agent conditions. Identical task identifiers ensured consistent comparisons. Audit completeness was calculated using predefined audit requirements. Response time was obtained from recorded execution timestamps.

Discussion

The secured MCP agent achieved a task success rate of 94.67%. This exceeded the manual baseline of 87.87%. The unsecured agent achieved a lower success rate of 80.53%. Adversarial testing revealed significant security differences between conditions. The unsecured agent executed unauthorised actions in 61.33% of adversarial cases (Greshake et al., 2023). The secured agent reduced this rate to 3.33%. Manual administration recorded a 6.00% unauthorised-action rate. These findings demonstrate improved resistance through security controls. Audit completeness reached 98.6% for the secured agent. Manual administration achieved 80.1% completeness. Policy enforcement added 7.12 seconds to transaction execution. Manual operations required a median of 613.15 seconds. Rapid security responses are important during ransomware incidents (Beaman et al., 2021). MCP provides standardised interfaces for external tool interactions (Hou et al., 2025). Reasoning and tool selection support complex agent operations (Yao et al., 2023). However, these findings reflect simulated testing rather than live deployments. Further validation using real enterprise storage environments remains necessary. The time factor is critical in case of a ransomware attack, because quick isolation mitigates losses (Beaman *et al.,* 2021). Uniform interfaces of the tools made such results feasible, though MCP servers remain susceptible to security threats (Hou et al., 2025). Effective reasoning and tool selection helped to perform multiple steps correctly (Yao et al., 2023). The research was conducted in a testbed environment; hence, further analysis is necessary in a real setting.

Conclusion

The research demonstrate that the secured MCP agent improved task performance and security within the simulated environment. It achieved higher task success and audit completeness than the tested alternatives. It also reduced unauthorised actions during adversarial testing. However, these findings apply only to the synthetic experimental setup. They do not establish superiority over human administrators in real enterprise environments. Further validation is necessary using live storage infrastructure, real operational workloads, and experienced administrators. Future research should examine scalability, operational reliability, and security under real deployment conditions. Thus, the research indicates that autonomous storage management is more reliable, effective, and auditable when employing the zero trust paradigm as opposed to performing all the operations manually. Future research needs to test these simulation results in a real storage environment.

Reference List

Beaman, C., Barkworth, A., Akande, T. D., Hakak, S., & Khan, M. K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, *111*, 102490. https://doi.org/10.1016/j.cose.2021.102490

Beaman, C., Barkworth, A., Akande, T. D., Hakak, S., & Khan, M. K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, *111*, 102490. https://doi.org/10.1016/j.cose.2021.102490

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 79–90. https://doi.org/10.1145/3605764.3623985

Hasan, M.M., Li, H., Fallahzadeh, E., Rajbahadur, G.K., Adams, B. & Hassan, A.E., 2025. Model context protocol (mcp) at first glance: Studying the security and maintainability of mcp servers. *ACM Transactions on Software Engineering and Methodology*. https://dl.acm.org/doi/abs/10.1145/3814959

Hou, X., Zhao, Y., Wang, S., & Wang, H. (2025). Model Context Protocol (MCP): Landscape, security threats, and future research directions. *arXiv*. https://arxiv.org/abs/2503.23278

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems, 36*. https://arxiv.org/abs/2302.04761

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*. https://arxiv.org/abs/2210.03629