Automated network access control with LLMs

Jonas Wessner, Tobias Meuser, Frank Kargl

Angaben zur Veröffentlichung / Publication details:

Wessner, Jonas, Tobias Meuser, and Frank Kargl. 2026. "Automated network access control with LLMs." In *7th KuVS Fachgespräch on Machine Learning in Networking (MaLeNe), University of Augsburg, Augsburg, Germany, March 19-20, 2026: proceedings*, edited by Michael Seufert, Andreas Blenk, and Björn Richerzhagen. Augsburg: Universität Augsburg.

7TH KUVS FACHGESPRÄCH ON MACHINE LEARNING IN NETWORKING (MaLeNe)

PROCEEDINGS

MARCH 19-20
2026

UNIVERSITY OF AUGSBURG, AUGSBURG, GERMANY

Automated Network Access Control with LLMs

Jonas Wessner
Ulm University
Email: jonas.wessner@uni-ulm.de

Tobias Meuser
Technical University Darmstadt
Email: tobias.meuser@kom.tu-darmstadt.de

Frank Kargl
Ulm University
Email: frank.kargl@uni-ulm.de

***Abstract*—Access control policies provide an essential layer of security for organizations' systems and data by defining rules for allowed and disallowed accesses, thereby preventing unauthorized access to resources. Access policies are inherently dynamic, as they must adapt to evolving organizational requirements such as infrastructural, operational, or personnel changes. As networks grow larger, the complexity of access policies and the frequency of required updates increase, traditional manual approaches to access control configuration no longer scale. In this paper, we explore the potential of large language models (LLMs) for automating network access control by automatically translating help desk tickets into policy changes. We summarize the challenges of using LLMs for network access control automation and present insights from preliminary experiments. Our early experiments show that LLMs are capable of understanding user intent and applying configuration changes in isolated examples when relevant background information is provided within the model's context. At the same time, we identify integration into production systems and reliability as major challenges to be addressed in future work.**

***Index Terms*—access control, intent-based networking, autonomous networking, LLM, natural language**

I. Introduction

Network access policies are a central building block of security architectures, defining rules for allowed and disallowed accesses to resources such as data, applications, and physical infrastructure. Requirements for accessibility of resources are usually highly dynamic. For instance, applications may be moved from testing to production environments, or user permissions may change as they switch departments or teams. In traditional workflows, users can request access control changes through help desk tickets. However, as systems grow in the number of users and resources, and as the landscape of access control mechanisms becomes increasingly complex and heterogeneous, manual implementation of continuous access control changes no longer scales. With IT help desks becoming a bottleneck in access control configuration, prolonged queuing of user tickets can hinder company productivity and potentially lead to slow reactions to system vulnerabilities. These challenges highlight the need for more agile and automated approaches to access control configuration.

The rise of Large Language Models (LLMs) provides a new angle to this topic, as their strengths in natural language understanding and reasoning can be helpful for automating help desk ticket processing. LLMs can be used to automatically understand user tickets, interpret them, and generate policy updates, as illustrated in Figure 1. However, LLMs are prone to hallucinations, which can be fatal if they result in incorrect policies [1]. Furthermore, user requests may be vague or lack essential technical details, or even propose nonsensical updates, making it difficult to synthesize precise network access control policies directly from user requests. In this paper, we identify key challenges in automating access control configuration and share insights from preliminary experiments. Our results highlight the potential of state-of-the-art LLMs for understanding user requests and constructing structured policies, while also identifying several challenges for their use in real-world systems. Based on these insights, we suggest future work to further explore LLM-based systems for access control management to serve the dynamic requirements of current and future secure IT infrastructure.

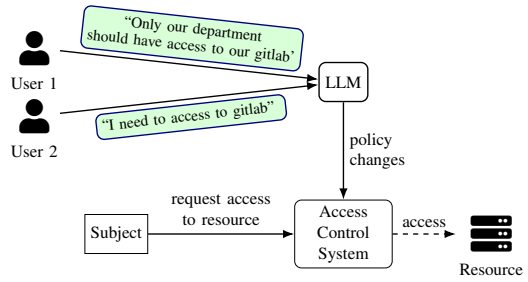

Fig. 1. Overview of an LLM-based access control system. On the control plane, an LLM is used to translate help desk tickets into access policies. On the data plane, these policies are enforced by the access control system.

II. Key Challenges in LLM-Based Access Control

In this section, we discuss central challenges for automated network access control policy synthesis from natural language.

A. Vague and Incomplete User Requests

Typically, users who report access control changes via help desks do not come from a technical background. Thus, such help desk tickets are inherently vague and lack technical details. For instance, the message "I need access to gitlab" cannot be directly translated into a low-level policy for two reasons: First, entities mentioned in the text (e.g., "gitlab") are ambiguous without further context (e.g., could refer to one of multiple existing gitlab servers). Second, users refer to high-level entities without specifying or knowing technical details, such as IP addresses and ports, that are necessary for implementing the policies. Additionally, users might even completely omit important information, which suggests that an LLM-based access control system should be able to query the user for additional information in such cases.

B. Heterogeneous Landscape of Access Control Mechanisms

In recent years, access control has become increasingly complex and heterogeneous, with trends such as zero trust [2] and micro-segmentation [3] pushing towards more layered and fine-grained access control. Specifically, access control policies should not only be enforced at a single point (such as a perimeter firewall), but at as many security layers as possible, including, for example, SDN switches, host firewalls, and application-level logins. This complex landscape of access control mechanisms poses challenges not only for manual implementation of access control policies but also for LLM-based approaches. For instance, it is known from previous works that LLMs struggle with large contexts and nuanced distinctions between software versions, both of which are relevant for correctly managing large-scale and heterogeneous infrastructure [4], [5].

C. User Intent Conflicts

An access control system that serves change requests from end users, similar to how a help desk works, is a multi-user system. Thus, requests from one user can interfere with or be contradictory to requests from other users. This calls for a conflict resolution approach, possibly through defining precedence among requests or an authorization scheme.

III. Lessons from Early Exploration

We obtained a small set of real-world help-desk tickets from a university computing center, containing user messages related to access control policy changes. Using insights from this dataset, we conducted preliminary experiments to explore the potential as well as revealing challenges of using LLMs for automating access control management. In the following, we share lessons learned from our experiments, which may support future research in this field.

A. Potential of LLMs for Understanding User Intent

As a case study, we presented state-of-the-art LLMs with help desk tickets and manually selected necessary background information that help desk staff needed to correctly process such tickets, providing this information as part of the prompt. This contextual information included (1) the identity of the user and related information such as previous requests or their relationship to other relevant parties, (2) relevant excerpts of the access control configuration files, and (3) any additional background information about the organization relevant to this ticket. We then asked an LLM to construct an access policy in a given JSON format and compared the result with the expected policy outcome. We found that current LLMs can often accurately understand user intent and can adjust access control configuration accordingly. Furthermore, we find that current models benefit from in-context examples [6]: When we illustrate the task using the example of another ticket as part of the prompt, the model behavior is closer to our expectations. In production systems, tickets previously processed by humans could be utilized to provide such examples to the LLM.

B. Challenges of Full Automation in Real-World Settings

There remains a gap between isolated use cases and complex production environments. Real-world use cases with large, unstructured code bases of configuration files and large organizations pose challenges to the problem understanding of LLMs [5]. We identify the automated selection of relevant context information and navigation of configuration code bases to identify relevant sections as core challenges. To address these challenges, we are exploring viable approaches. To select contextual information, embedding-based similarity search [7], similar to that used in Retrieval Augmented Generation (RAG) [8], might be useful. For navigating the code base, approaches based on LLM agents [9] seem promising, allowing for iterative exploration of code bases [10], [11]. Agentic approaches might also be suitable for asking follow-up questions to the user if their request is missing essential information.

IV. Reliability of LLM Decisions

While we find that LLMs can often accurately understand user intent, we also identify cases in which LLMs respond to user tickets with incorrect solutions. Especially when the user message is not written clearly or is missing important information, we find that the LLM tends to output wrong answers instead of suggesting to ask the user for clarification. Such inaccuracies highlight that, while LLMs are a powerful tool for network access control automation, they must be integrated into production systems with caution. We suggest that future work create datasets that allow measuring the accuracy of specific LLM-based designs, which could help estimate the utility and risk of such systems. Additionally, certain security checks, such as rule-based checks and human reviews, should be performed before deploying LLM-generated policy changes to avoid misconfiguration. At the same time, human intervention should be kept minimal to not degrade the efficiency of the system, highlighting the trade-off between automation and human supervision.

V. Conclusion

In this paper, we presented the concept of LLM-based network access control management. We highlighted the need for automated access control solutions to handle dynamic and increasingly complex IT infrastructure and how LLMs emerge as a new tool for automating this task. We further shared insights from preliminary experiments. On the one hand, we found that state-of-the-art LLMs can often correctly understand user intent and edit access control configurations accordingly, while on the other hand, we identified challenges in scaling to complex real-world use cases and ensuring correctness of LLM outputs in production settings. Our work provides evidence that LLM-based access control is a promising field of research and worth further investigation. Open questions to be addressed are how to integrate LLMs into real-world systems and how to prove or estimate the reliability of LLM-based access control systems.

REFERENCES

[1] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *Transactions on Information Systems*, 2025.

[2] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero Trust Architecture (ZTA): A Comprehensive Survey," *IEEE Access*, 2022.

[3] N. Sheikh, M. Pawar, and V. Lawrence, "Zero Trust Using Network Micro Segmentation," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021.

[4] T. Wu, W. Wu, X. Wang, K. Xu, S. Ma, B. Jiang, P. Yang, Z. Xing, Y.-F. Li, and G. Haffari, "Versicode: Towards Version-Controllable Code Generation," *arXiv preprint arXiv:2406.07411*, 2024.

[5] T. Li, G. Zhang, Q. D. Do, X. Yue, and W. Chen, "Long-Context LLMs Struggle with Long In-Context Learning," *arXiv preprint arXiv:2404.02060*, 2024.

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Curran Associates, Inc., 2020.

[7] N. Reimers and I. Gurevych, "Sentence-bert: Sentence Embeddings Using Siamese BERT-Networks," *arXiv preprint arXiv:1908.10084*, 2019.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2020.

[9] X. Dong, X. Zhang, W. Bu, D. Zhang, and F. Cao, "A Survey of LLM-based Agents: Theories, Technologies, Applications and Suggestions," in *International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*, 2024.

[10] T. Gupta, L. Weihs, and A. Kembhavi, "CodeNav: Beyond Tool-Use to Using Real-World Codebases with LLM Agents," *arXiv preprint arXiv:2406.12276*, 2024.

[11] K. Zhang, J. Li, G. Li, X. Shi, and Z. Jin, "Codeagent: Enhancing Code Generation with Tool-Integrated Agent Systems for Real-World Repo-Level Coding Challenges," *arXiv preprint arXiv:2401.07339*, 2024.