УДК 004.77

B. Hinko

O. Boretskyi

V. Marianovskyi

PRACTICAL ASPECTS OF AUTOMATED COLLECTION OF INITIAL DIAGNOSTIC INFORMATION FOR THE USER NETWORK TECHNICAL SUPPORT SERVICE

Taras Shevchenko National University of Kyiv

Анотація

У статті розглядаються неефективності ручного збору діагностичної інформації в технічній підтримці мереж університетів. Детально описано розробку та практичні аспекти автоматизованої системи, призначеної для оптимізації цього процесу. Система, створена з використанням Python, FastAPI та Celery, інтегрується з тикет-системою Zammad для автоматизації ідентифікації користувача, діагностики мережевих комутаторів через SSH та генерації звітів, що значно прискорює вирішення проблем

Ключові слова: автоматизація, діагностика мережі, технічна підтримка, збір даних, SSH, Python, FastAPI, Zammad.

Abstract

This paper addresses the inefficiencies of manual diagnostic data collection in university network technical support. It details the development and practical aspects of an automated system designed to streamline this process. The system, built with Python, FastAPI, and Celery, integrates with the Zammad ticketing system to automate user identification, network switch diagnostics via SSH, and report generation, significantly accelerating problem resolution.

Keywords: automation, technical support, network diagnostics, data collection, Python, FastAPI, Celery, Zammad, network incidents.

Introduction

Ensuring uninterrupted access to network resources and the prompt resolution of technical problems are key tasks for the information services of large organizations, particularly universities. At Taras Shevchenko National University of Kyiv, the efficiency of the technical support service directly impacts the quality of educational and scientific processes [1]. Current approaches to initial network incident diagnosis predominantly rely on manual data collection by engineers [2], leading to time delays, human error, and scaling difficulties. This underscores the urgent need for automated systems to optimize these processes [3]. This work details such a system, developed to automate the collection of initial diagnostic information. The development of a comprehensive solution was driven by the limitations of existing standalone methods for network equipment interaction.

Research results

The proposed solution is a software complex designed for integration into existing support service workflows, aligning with IT service management standards [4]. Its core is a Python-based backend service using the FastAPI asynchronous web framework [5] and the Celery distributed task queue system [6] for efficient request processing. The system automates user identification, determines the relevant network switch, and establishes a secure SSH connection to execute standardized diagnostic commands [7]. Collected data, including interface status, error counts, MAC address tables, and cable test results, undergo automatic analysis.

A structured diagnostic report is then generated and immediately transmitted to the Zammad ticketing system, supplementing the user's request and enabling faster problem resolution by first-line engineers [8].

Data collection via remote terminal (SSH) is central to the system. Python libraries like Netmiko or Paramiko [7] are utilized to programmatically establish SSH connections. Secure credential management is achieved by retrieving login information from protected runtime environments. Once connected, the system executes a sequence of CLI commands (e.g., show interface, show mac address-table) to extract diagnostic data. The raw text output is parsed, often using regular expressions, into a structured format (like JSON) for analysis and report generation in Zammad. Robust error handling manages connection timeouts, authentication failures, and unexpected outputs.

Several standalone methods for network equipment interaction present drawbacks. Direct console connection, while providing full control, is impractical for automated distributed data collection due to its physical access requirement. SNMP (Simple Network Management Protocol), an industry standard for monitoring, allows centralized data collection via MIBs. However, obtaining specific CLI-equivalent diagnostic details can be complex, and earlier SNMP versions (v1/v2c) have significant security flaws. While SNMPv3 is more secure, its configuration is more demanding, and it's not universally supported. Thus, SNMP is often insufficient for in-depth automated diagnostics. Remote terminal access via Telnet and SSH mirrors manual administration. SSH offers secure, encrypted CLI access [7] and was chosen as the system's base interaction method. However, a standalone SSH client lacks the comprehensive logic for user identification, target device determination, dynamic command formation, output parsing, data analysis, and ticketing system integration provided by the developed system. Telnet is inherently insecure due to its lack of encryption. Many devices offer a WEB UI, but these graphical interfaces are highly variable across vendors and models, making automation via web scraping unstable and maintenance-intensive due to frequent UI changes with firmware updates. The most modern approach, REST API, allows standardized programmatic interaction (usually JSON over HTTP) and is ideal for automation. However, its availability is largely limited to newer equipment from select vendors and is not yet a universal solution for diverse, existing network infrastructures [9].

Although direct SSH-based automation gathers crucial device-specific diagnostics, data from a single network switch offers an incomplete perspective on multifaceted network problems. Effective troubleshooting necessitates a broader context, achieved by augmenting switch data with intelligence from other infrastructure components. For example, integrating with monitoring tools like Grafana can reveal historical performance metrics that instantaneous CLI outputs miss. Centralized logs (from Syslog or ELK stacks) may contain correlated event data, such as device errors or security alerts. Furthermore, network discovery platforms like NetDisco offer dynamic topology and inventory details vital for assessing the scope and root cause of an issue

Conclusions

The analysis of technical support operations within the university's dormitory network identified significant inefficiencies in manual diagnostic data collection, primarily concerning time consumption and consistency, thereby validating the need for automation. A conceptual model and a flexible, modular architecture for an automated system were developed, designed for integration with existing ITSM tools. Practical implementation yielded a functional software prototype utilizing a Python, FastAPI, and Celery stack, with demonstrated Zammad API integration. Laboratory testing confirmed the prototype's operational stability and its potential for substantial acceleration of the diagnostic process compared to manual methods. The primary result is a tested software solution, exhibiting scientific novelty in its architectural adaptation to a specific environment and offering considerable practical value for optimizing support service operations. Enhancing the automated system to interface with these external data sources is a key direction for future development, aiming to provide support engineers with richer, more actionable diagnostic insights. Future development may also focus on expanding diagnostic features and enhancing analytical capabilities.

REFERENCES

1. Galup, Stuart, et al. "Information technology service management: an emerging area for academic research and pedagogical development." Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce. 2007.

2. The Practice of Network Troubleshooting [Електронний ресурс]. https://www.oreilly.com/library/view/the-practice-of/9781492033333.

3. Reference Architecture for Automated IT Incident Management [Стаття]. https://ieeexplore.ieee.org/document/9684440.

4. ISO/IEC 20000 1:2018 Information Technology – Service Management System Requirements.

5. Lubanovic, Bill. FASTAPI : Modern Python Web Development. First edition, O'Reilly Media, Inc., 2023.

6. Celery Distributed Task Queue for Micro services [Електронний ресурс]. https://link.springer.com/chapter/10.1007/978-981-15-9111-4_58.

7. Choi, Brendan. "Python Network Automation Labs: SSH in Action, paramiko and netmiko Labs." Introduction to Python Network Automation Volume II: Stepping up: Beyond the Essentials for Success. Berkeley, CA: Apress, 2024. 121-227.

8. Improving MTTR through Automated CLI Diagnostics [Електронний ресурс]. https://www.networkworld.com/article/3664147/improving-mttr-through-automated-cli-diagnostics.html.

9. Shen, Ningguo, et al. Campus Network Architectures and Technologies. CRC Press, 2021.

***Гінько Богдан**, студент 2 курсу магістратури, КІ, факультет радіофізики, електроніки та комп'ютерних систем, Київський національний університет імені Тараса Шевченка, м. Київ, e-mail: bohdan.hinko37@gmail.com*

*Науковий керівник: **Борецький Олександр**, кандидат технічних наук, асистент, кафедра комп'ютерної інженерії, факультет радіофізики, електроніки та комп'ютерних систем, Київський національний університет імені Тараса Шевченка, м. Київ, e-mail: oleksandr.boretskyi@knu.ua*

***Мар'яновський Віталій**, кандидат технічних наук, асистент, кафедра комп'ютерної інженерії, факультет радіофізики, електроніки та комп'ютерних систем, Київський національний університет імені Тараса Шевченка, м. Київ, e-mail: vitalik_m@univ.kiev.ua*

***Hinko Bohdan**, master's degree student, Faculty of Radiophysics, Electronics and Computer Systems, Taras Shevchenko National University of Kyiv, Kyiv, e-mail: bohdan.hinko37@gmail.com*

***Boretskyi Oleksandr**, Candidate of Technical Sciences, Assistant Professor, Department of Computer Engineering, Faculty of Radiophysics, Electronics and Computer Systems, Taras Shevchenko National University of Kyiv, Kyiv, e-mail: oleksandr.boretskyi@knu.ua*

***Marianovskyi Vitalii**, Candidate of Technical Sciences, Assistant Professor, Department of Computer Engineering, Faculty of Radiophysics, Electronics and Computer Systems, Taras Shevchenko National University of Kyiv, Kyiv, e-mail: vitalik_m@univ.kiev.ua*