Network Fuzzing Framework for Communication Protocol*

Hyeon Park, SongHa Ryu and TaeGuen Kim†
Korea University, Seoul, Republic of Korea
{rainbow00000, wasabi060725, tg_kim}@korea.ac.kr

Abstract

This study proposes a customized network-fuzzing framework grounded in protocol specifications and message-format specifications used in network communications. The framework applies three targeted strategies—protocol-specification violations, message-format-specification violations, and boundary/type-error value injection—within a closed-network testbed to generate and analyze manipulated message behaviors. In addition, we incorporate a lightweight LLM to assist automated generation and diversification of mutation rules. Experimental results show that manipulated messages frequently disrupted session continuity, causing session termination and re-creation, thereby revealing implementation-level weaknesses in input handling. Future work will deepen the fuzzing coverage and improve reproduction techniques to identify concrete vulnerabilities and develop practical mitigations.
Keyword: Network protocol fuzzing, Protocol vulnerability, Syntax mutation

1 Introduction

In modern ICT environments, communication protocols between clients and servers are essential but are increasingly exposed to security vulnerabilities that threaten the stability of entire networks. These weaknesses can lead to large-scale service outages, data breaches, and exploitation by botnet-based attacks such as Mirai. In this study, we propose a customized network-fuzzing framework—based on protocol specifications and message-format rules—that models input structures and intentionally generates message-format–violating mutations to expose protocol parsing and implementation flaws. The framework focuses on identifying and analyzing protocol vulnerabilities and reproducing real-world attack scenarios to develop appropriate countermeasures. Through this approach, we aim to strengthen the security of network communications and contribute to the construction of a safer network environment. One of the key contributions of this study is the use of an LLM-based rule generation mechanism to produce more intelligent and automated mutated messages. By leveraging LLMs to analyze protocol specifications, message formats, and real traffic, the approach can automatically generate sophisticated mutation rules at scale, enabling detection of vulnerabilities that are difficult to uncover with manually defined rule sets.

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. P-62, December 16-18, 2025, Sapporo, Japan.
† Corresponding author

2 Proposed Framework

The proposed framework is built on a closed-network testbed (Fig. 1) and is designed to observe, manipulate, and analyze traffic of network communication protocols in an environment that emulates real operational deployments. The core components are a server and services, a fuzzing module (traffic interception and mutation), an access gateway, and user devices; all communications between the server and user devices are relayed through the fuzzing module.

The fuzzing module is augmented with streaming capture and message-hooking capabilities to record and inspect message bodies in real time, and scripts that define fuzzing strategies and procedures apply deliberate mutations such as protocol-specification violations, message-format violations, and boundary-value manipulations. To ensure experimental reproducibility and enable root-cause analysis, session management and comprehensive logging (original messages, mutation history, etc.) are maintained; mutated inputs are analyzed quantitatively and qualitatively for parser exceptions, session transitions, and error responses.

The architecture is intentionally isolated within a closed network to safely reproduce attack scenarios and empirically assess implementation weaknesses in input validation and state management.

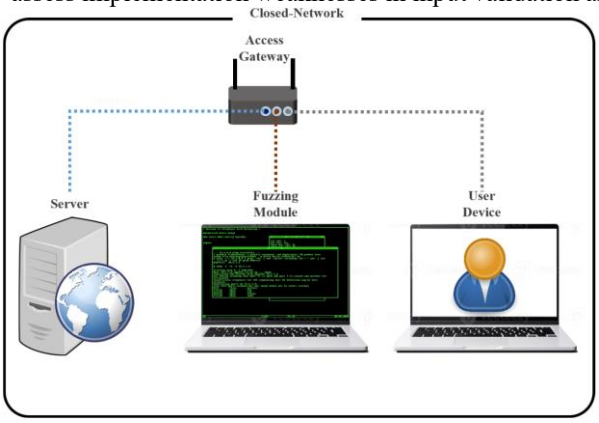

Figure 1 Closed-Network Testbed for Fuzzing

3 Conclusion

We pre-defined fuzzing rules for each message type and configured the fuzzing module to randomly select and apply a rule from the corresponding set whenever a message was detected. As a result, we observed that delivery of certain mutated messages led to connection termination and session re-creation. This suggests that inputs violating protocol specifications and message formats can disrupt normal protocol processing and state management. However, this study did not identify definitive vulnerabilities such as crashes or memory corruption. Therefore, future work will diversify and deepen mutation strategies to enable more sophisticated testing and will introduce dynamic instrumentation and runtime monitoring to facilitate the detection of hidden vulnerabilities.

Acknowledgment

This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2022-00164800)