Project Proposal: Understanding and Detecting Bugs in Network Systems

Penghui Li 1155137827
phli@cse.cuhk.edu.hk

ABSTRACT

Network systems connect the Internet world. Bugs in network systems can lead to denial-of-service, performance issues, *etc.* Understanding network system bugs can benefit both the system developers and security analysts.

However, to the best of our knowledge, the bugs in network systems have not been well understood. The potential security impact has not been well investigated as well. In this project, we aim to understand the bugs in network systems. We hope to learn lessons to further guide future bug detection in network systems.

1 NETWORK SYSTEM BUGS

Network systems are the hardware and software components from which these networks are built [7]. They are commonly applied to perform data analytics, e-commerce, storage back, *etc.* [3]. The network architectures have been involved rapidly along with the advance of new network applications, *e.g.,* video applications and video streaming.

Bugs commonly exist in network systems. For example, more than 200 bugs were found in the Linux IP stack [1]. Such bugs can cause severe security and reliability problems like crashes, denial-of-service, and failures [3].

Understanding the root causes of the bugs in network systems is important. In particular, first, the empirical knowledge can benefit system developers to avoid making similar mistakes. Second, security analysts are thus possible to design new approaches to identifying bugs. The network systems thus can be more secure and reliable. The user experience can be significantly improved as well [2].

2 RELATED WORK AND RESEARCH PROBLEMS

There are substantial works on characterizing different types of bugs in many systems (*e.g.,* regular-expression denial-of-service bugs in web applications [4–6], performance bugs in C/C++ compilers [8, 9]). Such studies greatly advance the improvement of security analysis and bug detection techniques. For example, Yang *et al.*, guided by their empirical study, developed a rule-based static system to effectively and scalably detect new bugs in C/C++ compilers.

Toward understanding bugs in network systems, Yin *et al.* studied the bugs in open-sourced router software [10]. They concluded the many bugs could lead to reliability issues and were hard to patch. Gill *etc.* investigated bugs in data center networks and demonstrated that the load balancers were highly buggy [3]. However, to the best of our knowledge, these works either fall short of characterizing the root causes of the bugs or have not included newer network software. Furthermore, how such knowledge can benefit the security analysis has not been demonstrated yet.

In this project, we plan to conduct a comprehensive study on existing bugs in several network systems. Specifically, we aim to answer the following research questions:

- What are the main categories of the bugs?
- What are the main factors that cause bugs?
- How widespread are the bugs in network systems?
- How severe are the bugs in network systems?

Second, we hope to summarize the general patterns of the bugs in network systems and the common practices in patching network system bugs.

Note that, there is no guarantee that we are able to design a better tool to detect network system bugs or not as this largely depends on what we find in the empirical study, which is unknown for now. We do not promise we will be able to identify new bugs within this project. Nevertheless, the empirical knowledge on network system bugs can definitely benefit the community to develop more secure and reliable network systems.

3 LOGISTICS

Regarding the time of *6 weeks* for this project, we plan to use *3 weeks* on understanding existing bugs in network systems; we then spend *2 weeks* on summarizing the bugs and implement necessary tools (if available) for more practical bug detection; last, we use *1 week* to present our findings in the final reports. The time schedule is tentative and might change accordingly.

As required, we currently host the project content on a GitHub page[1]. We also create on a GitHub repository[2] and will maintain the necessary project artifact (*e.g.,* source code (if any), dataset) there.

CSCI '5570, Spring 2021, CUHK
2021.

[1] https://peng-hui.github.io/csci5570-project.html
[2] https://github.com/peng-hui/csci5570-project

REFERENCES

[1] . 2021. Kernel.org Bugzilla-Bug List. https://bugzilla.kernel.org/.

[2] . 2021. Modern network bugs require fast customer-centric solutions, not patches. https://undo.io/resources/modern-network-bugs-require-solutions-not-patches/.

[3] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. 2011. Understanding network failures in data centers: measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 Conference.*

[4] Guoliang Jin, Linhai Song, Xiaoming Shi, Joel Scherpelz, and Shan Lu. 2012. Understanding and detecting real-world performance bugs. In *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. Beijing, China.

[5] Adrian Nistor, Po-Chun Chang, Cosmin Radoi, and Shan Lu. 2015. Caramel: Detecting and fixing performance problems that have non-intrusive fixes. In *Proceedings of the 37th International Conference on Software Engineering (ICSE)*. Florence, Italy.

[6] Adrian Nistor, Linhai Song, Darko Marinov, and Shan Lu. 2013. Toddler: Detecting performance problems via similar memory-access patterns. In *Proceedings of the 35th International Conference on Software Engineering (ICSE)*. San Francisco, CA.

[7] Dimitrios Serpanos and Tilman Wolf. 2011. *Architecture of network systems.* Elsevier.

[8] Chengnian Sun, Vu Le, Qirun Zhang, and Zhendong Su. 2016. Toward understanding compiler bugs in GCC and LLVM. In *Proceedings of the 25th International Symposium on Software Testing and Analysis (ISSTA)*. Saarbrücken, Germany, 294–305.

[9] Xuejun Yang, Yang Chen, Eric Eide, and John Regehr. 2011. Finding and understanding bugs in C compilers. In *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. San Jose, CA.

[10] Zuoning Yin, Matthew Caesar, and Yuanyuan Zhou. 2010. Towards understanding bugs in open source router software. *ACM SIGCOMM Computer Communication Review* (2010).