CS846 Week 10 Reviews

Zhaoyi Ge

Wolves in the Repository: A Software Engineering Analysis of the XZ Utils Supply Chain Attack

Problem Being Solved

This paper conducts an empirical study on the XZ Utils Attack that spanned from 2022 to 2024. The author conducts analysis of the attack by collecting a dataset of the attacker's SE activities, analyzing the timeline of the attack, categorizing the motivation behind each activity, and evaluating the impact of the attack.

New Idea

The author collected SE activities of the attacker, including mailing list correspondence, git commits, documentation maintenance. By collecting this information, we can observe many novel and sophisticated techniques in this attack. For example, from the timeline we see that this attack lasts multiple years, and the attacker gradually gained trust of the project maintainer, and eventually became the de facto manager of the project.

From the data we also see that the attacker mainly gained trust from the original maintainer by submitting low-risk contributions such as translation and documentation.

This paper further discussed the implication of this attack, especially from an OSS perspective. This attack revealed many weaknesses of the current open source community. The author proposed a few actions to strengthen the security of open source projects.

Positive Points

While most articles on the XZ Utils attack focuses on the technical details of the attack, this paper studies the activities of the attacker and reveals surprising facts about this attack, which is how the attacker mostly contributed to non-code related commits.

XZ Utils attack is one of the most impactful software attacks in recent years. While the backdoor itself did not cause any real damage due to prompt action from the community, many aspects of this attack such as the level of sophistication and timespan shocked the programming community, causing a lot of interest in this attack. This paper studies this interesting attack from a novel, SE perspective.

Negative Points

The paper failed to fully explore deeper factors of this attack. One of things this paper missed is how the attacker used scripts only found in release files, but not in git repositories to inject malicious code. This way the malicious build scripts would not go through the code review process. This is a very interesting SE/security topic, to talk about how we can manage project assets in a secure way. The attacker also exploited automated build tools such as CMake and M4 to plant the backdoor, but the authors did not give any insights on the role of such tools in this attack.

The discussion points of this paper, which covers several topics, are shallow and off the point. Many of them are just common sense and do not provide much insight.

Future Work

There should be future work on analyzing the role of different techniques involved in this attack from a SE perspective. Such techniques include placing malicious build files in release tarballs, hiding obfuscated malicious code in binary test files.

Rating

3/5. This paper studies an interesting topic but fails to fully explore it.

Discussion Points

What is the implication of this attack? What did you learn?

A student pointed out that attacks on open source software is not uncommon nor hard. An example of this is researcher from University of Minnesota deliberately made malicious commits to the linux kernel and the commit is accepted. The professor explained that modern SE practices evolve constantly and were never designed to resist multi-year adversarial manipulation.

Students noted that the attacker exploited social trust rather than software flaws, drawing parallels to insider-threat models in industry. In the industry, there are sometimes

leaks from employees who have bad intentions when they join the company, but later on decide to be malicious. The students discussed if this is a possibility. This raised questions about how intent should be evaluated in long-term open-source participation. The presenter suggested this possibility is low, because the presenter uses an anonymous name, with no internet record of anything related to that name, and with very little open source contributions prior to the attacker starting to contribute to the XZ Utils attack. The presenter also suggested that it is likely that an organization is behind this attack due to the complexity of this attack. The presenter explained that national agencies such as the NSA are known to place backdoors in cryptographic schemes.