2023 Spring CNS Reading Critique #2

Reflections on Trusting Trust VS Reflections on trusting distributed trust

R11921A16 何秉學

I. Summary of Reflections on Trusting Trust

It attempts to convey the concept that there are vulnerabilities in the compiler that are not easy to find. The main concept is that if the compiler is tampered with maliciously, the whole system will become vulnerable. In this paper, the author tried to demo how his perspective and actual attack concept. In addition, this kind of attack is untraceable, which means it's hard to defend. The conclusion is before we trust the computing system, we must take compiler security seriously and try to find an algorithm or method to detect if the compiler is tampered with or not.

II. Summary of Reflections on trusting distributed trust

Nowadays, the authors have seen an explosion of academic and industrial cryptographic systems built on distributed trust, including secure multi-party computation applications and blockchains. These systems have great potential for improving security and privacy, but face a significant hurdle on the path to deployment. This paper attempt to establish a system that easy to set up a distributed-trust application without expensive, cross-organization coordination. For now, bootstrapping without cross-organization coordination can enable small organizations to securely deploy distributed-trust systems such as end-to-end encrypted messaging applications that could use the distributed trust to establish a public-key infrastructure or backup secret keys.

III. Comparison

In "Reflections on Trusting Trust", talked about the computing system security in compiler exploitation with uneasy to detect property. And the other one talked about how to construct a trusting system nowadays in which the global system connects and how to deploy the system easily. So, though they talked about "trusting", the previous one focused on attacking and detecting potential crises, and the other one focused on establishing trust and maintaining it within the distributed system. The solution of these two papers first proposed a solution that the compiler should add some protection to the source code to prevent the injection of malicious code. The other one proposed a method such as distributed hash value to establish a trusting connection and use it to check the code that runs in the trust domain.

IV. Reflection

As I mentioned above, these papers proposed a few perspectives on different system articles. They used a different way to prove how important the problem is, and also proposed some concept or an actual way to address it. If I were the author of the first paper, I'll attempt to use some program analysis or runtime monitoring, to detect and prevent such attacks automatically. In addition, I'll extend the trusting trust concept to hardware attacks such as microprocessors or firmware backdoors, etc. In the second paper, I'll try to extend the distributed system trusting issue to blockchain technology which may have some vulnerabilities that can solve by the concept of this paper.