Seminar Feedback Report (Week 01)

2021315385 이건

Practical Software Diversity

Practical software diversity is a part of the moving target defense (MTD) approach, which alters software data to complicate attacks. Techniques such as canaries, non-executable (NX) memory, data execution prevention (DEP), and address space layout randomization (ASLR) have defended code from several attacks. However, attackers have found ways to bypass these defenses through code reuse attacks. This problem has highlighted the need for a more adaptive defense mechanism, which has led to the development of compiler-assisted code randomization—a form of practical software diversity.

Compiler-assisted code randomization actively modifies the layout of the code by rearranging functions and basic blocks. This technique relies on compiler-rewriter cooperation. They consolidate and embed metadata, such as basic blocks and layouts, into a master binary, which software vendors then distribute through legacy distribution channels. Once end users receive the master binary, they generate variants locally. This approach improves the reliability, transparency, compatibility, and cost-effectiveness of the distribution process.

In summary, compiler-assisted code randomization represents a practical application of software diversity as it makes attacks more complex. This topic is an excellent icebreaker for those interested in software security.