Seminar Feedback Report (Week01)

2021312738 소프트웨어학과 김서환

1주차 세미나의 내용은 소프트웨어의 다양성에 대한 것이다. 소프트웨어 다양성은 같은 의미의 코드이지만, 외형적으로 다른 형태의 코드를 의미한다. 또한, 간단하게 이동 목표 방어는 공격자의 관점에서 방어자가 위치를 변경하여 공격을 회피하는 방어 수법이라는 것이 설명되었다. 이러한 공격과 방어에 대해 자세히 살펴보자면, 코드 주입 공격은 공격자가 의도한 payload를 통해 스택의 반환주소를 공격자가 지정한 주소로 변경하여 공격자가 의도한 코드(악성코드)를 실행하는 방식으로 공격이 이루어진다. 코드 주입 공격을 방어하는 수단으로는 Canary, NX(비실행 메모리를 이용한 방법)이 있다. 또, 세그먼트의 base 주소를 무작위화시키는 과정 즉, 주소공간의 배치 무작위화를 통해 이동 목표 방어를 수행할 수 있다. 하지만, 더 나은 방안은 코드 그 자체를 다양화(무작위화)하는 것이다. 구체적인 방안으로 컴파일러 지원 코드 무작위화에 대해서 소개되었다. 컴파일러는 코드의 layout, 기본적인 block 등을 포함한 코드 구조를 이해하고, 메타데이터를 생성할 수 있다. 이러한 메타데이터를 master binary에 저장하면 소프트웨어 공급자가 앱 실행 파일을 배포할 수 있고, 필요에 따라 변형 버전을 생성할 수 있게 된다. 이렇게 하면, 단일 master binary로 여러 곳에서 호환성을 유지할 수 있게 되어 일관성이 높아지게 된다. 컴파일 시에 메타데이터가 수집되며, 링크시에 메타데이터를 통합하고 조정한다. 클라이언트 쪽에서 master binary를 수신하면, 무작위화된 코드로 재작성할 수 있으며, 이러한 과정 속에서 메타데이터가 통합되어 layout을 재구성한다. SPEC 2006 벤치마크 평가에서 대부분의 실행파일에서 랜덤화가 발생한 함수 및 기본 블록 수준에서 성능 손실이 거의 없었다. 1주차 세미나를 들으면서 소프트웨어 다양성과 공격자의 의도된 공격을 어떤 매커니즘으로 방어하는지, 이를 위해 어떤 형태로 코드가 다양화되는지 알 수 있어서 유익했다. 또한, 소프트웨어 다양성을 위한 코드 무작위화를 위해서는 메타데이터의 역할이 중요하다는 것을 알 수 있었다.