

공격자의 딜레마: 왜 불변 특징은 쉽게 사라지지 않는가?

1. 기능 유지의 제약 (Functional Constraints)

악성 행위(정보 탈취, C2 통신 등)의 목표는 동일하므로, 핵심 기능을 구현하는 코드 구조나 데이터는 필연적으로 남게 됩니다.



2. 회피를 위한 막대한 비용 (Prohibitive Evasion Cost)

Invariant를 제거하려면 아래와 같은 고비용 기술이 필요합니다.

- **메타모픽(Metamorphic) 엔진**: 실행 시마다 코드를 재조합. 개발 난이도가 극도로 높고 성능 저하 및 안정성 문제를 유발합니다.
- **실시간 암/복호화**: 필요한 코드만 잠시 복호화 후 다시 암호화. 심각한 CPU 오버헤드를 발생시킵니다.
- **가상머신(VM) 기반 실행**: 커스텀 인터프리터 방식. VM 개발/유지보수 비용이 크고, 인터프리터 자체가 새로운 탐지 시그니처가 됩니다.

Invariant 제거는 공격자에게 득보다 실이 큽니다.

특히 MaaS 형태의 악성코드는 개발 효율성을 위해 코어 로직을 재사용하는 경향이 강합니다.