


## 캡스톤디자인 면담 확인서

팀원	2013104053 김동준, 2013104129 한선우, 2016110312 송미희, 2014104110 신기성		
주제	IoT 디바이스 펌웨어 취약점 탐지 방법의 비교 분석		
면담일시	2019. 5. 27.	지도교수	조 진 성 
면 담 내 용	<ul style="list-style-type: none"> <li>● 팀원들이 각자 연구한 내용을 하나로 통합하는 작업이 필요함</li> <li>● 코드 취약점 부분에서는 정적 분석의 경우 다른 툴들에 비해 상대적으로 분석에 요구되는 부하가 적다는 점을 잘 활용할 필요가 있음. 그래서 다른 툴들과의 비교를 위해 동일한 재구성 코드를 대상으로 하더라도 재구성한 코드 외에 전체 코드에 대하여 분석을 하는 것 또한 가능할 것임. 전체 데모 시나리오상으로는 최초의 분석을 정적 분석으로 하게 될 것이고 분석 결과가 퍼저에게 범위를 축소할 힌트를 줄 수 있어야 함. 일반적 대형 프로그램 분석에서는 정적 분석의 오탐이 충분히 발생할 수 있으므로 데모 시나리오를 구성할 때 그러한 차이를 잘 보일 수 있도록 해야 함.</li> <li>● LibFuzzer를 이용하기 위하여 퍼저가 만들어주는 퍼징 데이터를 의미 있게 활용할 필요가 있음. 특히 퍼징 데이터를 바이트단위로 만들기 때문에, 해당 데이터를 일정 바이트 단위로 나눠 확인하면 의미 있는 데이터가 나올 가능성이 있고, 실제로 그러한 결과를 알 수 있었음. addressSanitizer 옵션을 이용하는 것과 이용하지 않는 것의 차이를 분명히 해야 할 것이며, 결과값으로 도출하는 로그를 정확히 이해할 필요가 있음.</li> <li>● 퍼징의 인풋 데이터의 범위를 줄 수 있는 옵션이 있는지 찾아보고 인풋에 유의미한 데이터를 삽입할 시 결과에 차이가 보이는지 연구하고 좀 더 의미 있게 활용하고자 함. libfuzzer와 afl 퍼저 간의 비교, 퍼저와 정적 분석 도구 간의 비교로 차이를 파악하고 비교가 될 수 있는 코드를 구현하고 비교 분석을 할 필요가 있음.</li> <li>● 기호 실행의 단점으로 인해 취약점 탐지가 어려운 점을 고려하여, 퍼징의 단점인 코드 커버리지를 보완하는 방식으로 기호 실행을 활용할 방법을 찾아보는 것이 요구됨.</li> </ul>		