사물 인터넷의 경량 펌웨어 업데이트를 위한 부분 퍼징 기법

김나현 [1], 이진민 [2], 이일구 [3]
[1] 성신여자대학교 융합보안공학과 학부생
[2] 성신여자대학교 미래융합기술공학과 박사과정
[3] 성신여자대학교 융합보안공학과/미래융합기술공학과 교수

20221079@sungshin.ac.kr, csewa56579@gmail.com, iglee@sungshin.ac.kr

Partial Fuzzing Technique for Lightweight Firmware Update of Internet of Things

Na-Hyun Kim[1], Jin-Min Lee[2], Il-Gu Lee[1,2]
[1]Dept. of Convergence Security Engineering, Sungshin Women's University
[2]Dept. of Future Convergence Technology Engineering, Sungshin Women's University

요 약

IoT(Internet of Things) 기기의 수가 급격히 증가하면서 무선 네트워크로 펌웨어와 데이터를 다운받아 업데이트하는 FOTA(Firmware Over-The-Air) 기술이 중요해지고 있다. 그러나, 종래 퍼징 기술은 펌웨어 취약점을 탐지할 때 요구되는 컴퓨팅 파워와 메모리가 커서 한정적인 자원을 지닌 IoT 기기에 적합하지 않다. 따라서 본 연구에서는 펌웨어 업데이트 파일에서 기존에 검증된 부분을 제외하고 업데이트된 부분만을 퍼징하는 부분 퍼징(Partial fuzzing) 기법을 제안한다. 실험 결과에 따르면 제안한 부분 퍼징 기법이 종래의 기법 대비 3분 더 빨리 11개의 크래시를 찾았고, 10분의 퍼징 시간 동안 평균 1,044 (2 unique) 크래시를 추가로 발견했으며 평균 메모리 사용량을 232 (KIB) 줄일 수 있었다.

1. 서론

IoT(Internet of Things) 기기가 널리 보급되면서 일상과 산업 전반에 활용되면서 펌웨어를 자동 업데이트하기 위해 무선 네트워크를 이용한 FOTA(Firmware Over-The-Air) 기술이 중요해지고 있다. 그러나 업데이트 빈도와 내용이 많아질수록 컴퓨팅 자원이 한정적인 IoT에서 펌웨어 취약성 분석 수행이 어려워지는 문제가 있다. 따라서 본 연구에서는 펌웨어의 업데이트된 부분만 퍼징하는 부분 퍼징(Partial fuzzing)을 제안하고 종래의 전체 펌웨어 퍼징 방식과 탐지 크래시 수, 메모리 사용량, 시간 복잡도를 비교한다.

본 논문의 주요 기여점은 다음과 같다.

- 경량 IoT 장치를 고려한 펌웨어 부분 퍼징 기법을 제안했다.
- 기존의 퍼징 기법과 제안하는 부분 퍼징 기법의 평균 소요 시간, 메모리 사용량, 크래시 수를 비교 평가하는 프레임워크를 제안했다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 펌웨어 업데이트 과정에서의 취약점 분석 선행연구를 분석한다. 3장에서 부분 퍼징 기법을 제안하고 4장에서 제안 방식과 종래 방식을 비교 및 분석한다. 그리고 5장에서 결론을 맺는다.

2. 관련연구

Hans Chandra[1]는 저전력 IoT를 고려하여 LWMesh (Lightweigh Mesh) 기반 P2PMesh(peer-to-peer mesh) 아키텍처를 활용한 무선 펌웨어 업데이트를 제안한다. 그러나 보안 및 인증 측면에서의 한계와 Hub Node와의 통신량 증가로 인한 전력 소모 증가 문제가 있다.

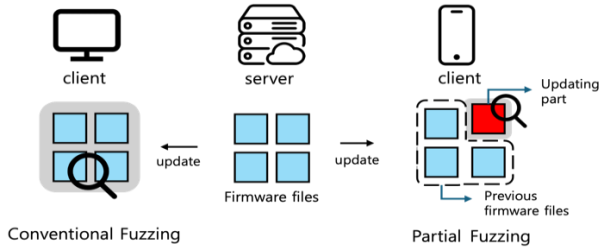
Pengfei[2]는 PATCHECKO라는 딥러닝과 동적 바이너리 분석을 결합한 기법으로 펌웨어 취약점과 패치된 함수 사이의 유사성을 비교하여 취약점을 정확하게 탐지한다. 그러나 PATCHECKO는 복잡도가 커서 경량 사물 인터넷 기기에 활용되기 어렵다.

Noy Hadar [3]는 경량 보안 장치를 사용하여 IoT 장치의 알려진 취약점을 효과적으로 차단하는 클라우드 기반의 취약점 완화 프레임워크를 제안한다. 그러나,

제안한 방식은 새로운 CVE 취약점 발견하기 어렵고, 보안 장치 적용까지 시간이 오래 걸린다.

3. 부분 퍼징 기법

종래 퍼징 기법은 펌웨어를 업데이트할 때 기존에 취약점을 분석했던 부분을 다시 분석하여 경량 IoT 장치에 비효율적이다. 이러한 문제를 해결하기 위해 본 연구에서는 그림 1과 같이 부분 퍼징 기법을 제안한다. 제안하는 부분 퍼징은 펌웨어 업데이트가 진행될 때 기존 펌웨어 파일과 비교하여 업데이트된 부분만을 퍼징한다.



(그림 1) 부분 퍼징의 구조

4. 실험 및 평가

본 실험에서는 종래 펌웨어 퍼징 기법과 제안하는 부분 퍼징 기법의 성능을 평가하기 위해 프로그램에 무작위로 데이터를 입력하여 버그 및 취약점을 찾아주는 자동화 툴인 AFL(American fuzzy lop)을 사용했다.

표 1은 10분 동안 크래시 발생 횟수와 메모리 사용량의 평균값에 대해 기존 방식과 제안 방식을 비교한 표이다. 10분 동안 찾은 총 크래시 수를 비교하면 제안 방식이 기존 방식 대비 평균적으로 1,044(2 unique)를 더 찾았으며, 기존 방식 대비 메모리를 232 (KIB) 만큼 적게 사용하는 것을 확인할 수 있었다. 이와 같이 부분 퍼징 기법을 사용하여 새롭게 업데이트된 부분만 검사함으로써 이미 알고 있는 부분까지 분석하는 기존 퍼징 방법보다 같은 시간 동안 더 많은 크래시를 찾을 수 있었다.

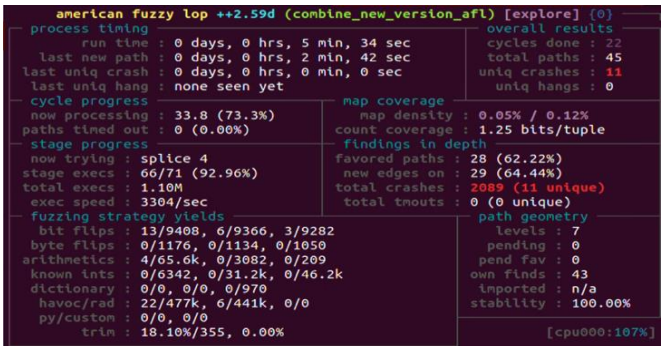
<표 1> 10분안에 찾은 전체 크래시 수와 메모리 사용량

Category	Total crashes (unique)	Memory usage (RES (KIB))
Conventional model	1,257 (8 unique)	4,412 KIB
Partial fuzzing	2,301 (10 unique)	4,180 KIB

표 2는 기존 방식과 제안 방식에서 11개의 고유한 크래시를 찾는 데까지 걸린 시간에 대해 비교한 표이다. 실험 결과에 따르면 기존 방식 대비 부분 퍼징 기법은 퍼징하는데 걸린 시간을 약 3분 1초 단축한 것을 확인할 수 있었다. 그림 2는 부분 퍼징 기법을 AFL에 적용했을 때, 예기치 않은 고유한 크래시(unique crashes)를 11개 찾을 때까지 걸린 시간을 측정한 실험 결과이다. 실행 시간은 그림 2의 runtime을 의미하는데, 총 5분 34초가 나온 것을 확인할 수 있었다.

<표 2> 11개의 크래시 수를 찾기 위한 시간

Category	Number of times
Conventional model	8min 35sec
Partial fuzzing	5min 34sec



(그림 2) 부분 퍼징 AFL

5. 결론

종래 퍼징 기법은 업데이트가 수행될 때마다 업데이트 내용이 많아져서 컴퓨터 자원이 한정적인 경량 IoT 장치에 적합하지 않았다. 이를 해결하기 위해 본 연구에서는 기존에 분석을 수행한 부분을 제외하고 업데이트된 부분만을 퍼징하는 부분 퍼징 기법을 제안했다. 실험 결과에 따르면 제안하는 퍼징 기법이 종래 퍼징 기법보다 평균 1,044 (2 unique) 크래시를 더 발견하고 메모리 사용량을 232 (KIB) 줄임으로 취약점 탐지 성능이 기존 대비 우수했다. 또한, 특정 크래시 수를 찾기까지 걸린 퍼징 시간도 3분 단축했다. 향후 연구에서는 제안한 부분 퍼징 기법을 검증하기 위한 테스트베드를 구축하고, 실증할 계획이다. 또한, 제안하는 퍼징 기법에서 에러가 발생하는 문제를 해결하기 위한 방안을 제안할 것이다.

Acknowledgement

본 논문은 2024 년도 정부재원(과학기술정보통신부 여대학원생 공학연구팀제 지원사업)으로 과학기술정보통신부와 한국여성과학기술인육성재단의 지원(WISET 계약 제 2024-138 호), 산업통상자원부 및 한국산업기술진흥원의 지원(No. RS-2024-00415520)과 과학기술정보통신부 및 정보통신기획평가원의 지원 (No. IITP-2022-RS-2022-00156310)을 받은 연구결과로 수행되었음.

참고문헌

[1] Chandra, H., Anggadjaja, E., Setya Wijaya, P., Gunawan, E. “Internet of Things: Over-the-Air (OTA) firmware update in Lightweight mesh network protocol for smart urban development,” *2016 22nd Asia-Pacific Conference on Communications (APCC),* Yogyakarta, Indonesia, 2016, pp.115-118, doi: 10.1109/APCC.2016.7581459.

[2] “Hybrid Firmware Analysis for Known Mobile and IoT Security Vulnerabilities,”*2022 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DNS),* Valencia, Spain, 2022, pp. 373-384, doi: 10.1109/DSN48063.2020.00053.

[3] Hadar, N., Shachar, S., Elovici, y. “A lightweight Vulnerability Migitation Framework for IoT Devices,” *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, TX, USA, Nov 3 2017, 71-75