https://kiisc.or.kr

2022년 한국정보보호학회 동계학술대회

CISC-W'22

Conference on Information Security and Cryptography-Winter 2022

2022년 11월 26일 (토) | 국민대학교

- 학술대회 등록대: 미래관 자율주행 스튜디오 (4층)
- 개회식/정기총회: 본부관 (1층)

Proceedings

주최 한국정보보호학회 Korea Institute of Information Security & Cryptology

주관 KMU 국민대학교 KOOKMIN UNIVERSITY

후원 국가정보원 NATIONAL INTELLIGENCE SERVICE, 과학기술정보통신부, 행정안전부, KISA 한국인터넷진흥원, ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute, NSR 국가보안기술연구소 National Security Research Institute, KiSTi 한국과학기술정보연구원 Korea Institute of Science and Technology Information www.kisti.re.kr, 지란지교시큐리티, PILAB

한국정보보호학회
Korea Institute of Information Security & Cryptology

임베디드 시스템 심층 분석을 위한 클라우드 기반 인스펙션 오케스트레이션 시스템 연구

이지수[1], 박예원[1], 안성규[2]. 박기웅[3‡]

[1,2]세종대학교 시스템보안연구실(학부생, 대학원생)

[3‡] 세종대학교 정보보호학과(교수)

Research of Cloud-based inspection orchestration system for in-depth analysis of embedded systems

Jisoo Lee[1], Yewon Park[1], Sung-Kyu Ahn[2], Ki-Woong Park[3‡]

[1,2]SysCore Lab, Sejong University(Undergraduate Student, Graduate Student)

[3‡] Department of Computer and Information Security, Sejong University(Professor)

요 약

최근 사물인터넷 기술이 산업 분야를 넘어 생활공간, 자동차, 의료 분야 등으로 활용 영역을 넓혀감에 따라 임베디드 시스템의 활용 범위가 기하급수적으로 증가하고 있다. 일반 레거시 시스템과 달리 임베디드 시스템은 저마다의 성능이 다르며 한 가지의 뚜렷한 목적을 가지고 설계되는 특징이 있다. 이로 인하여 다양한 형태의 펌웨어와 디바이스 드라이버가 필요하지만, 각 제조사 하드웨어 스펙 및 장착된 내부 기기, 센서의 종류와 조합이 다양하여 일관된 시스템 분석 도구를 활용하여 시스템 인스펙션 업무를 수행하는데 어려움이 따른다. 본 논문에서는 이러한 어려움을 해결하기 위해 임베디드 펌웨어, 디바이스 드라이버 환경에 활용되고 있는 다양한 시스템 인스펙션 도구 및 시스템 모니터링 소프트웨어를 분석하고, 이를 클라우드에 탑재시키기 위한 분석을 수행하여 클라우드 기반 인스펙션 오케스트레이션 시스템을 제안하고자 한다.

I. 서론

사물인터넷 기술이 산업 분야를 넘어 생활공간, 자동차, 의료 분야 등으로 활용 영역을 넓혀감에 따라 임베디드 시스템의 활용 범위가 기하급수적으로 증가하고 있다. 일반 레거시 시스템과 달리 임베디드 시스템은 초기에 단순 제어 역할만 수행하였지만, 최근 사물인터넷 개념과 융합되어 실시간 모니터링 및 실시간 제어 변경 기능 등 다양한 기능을 보유하고 있다[1]. 하지만 이러한 임베디드 소프트웨어의 기능을 활용하려면 펌웨어나 드라이버 제작과정에서의 시스템 목적, 구성하고 있는 센서 및 회로 등 다양한 환경을 고려해야 한다. 이를 위해 임베디드 시스템의 펌웨어나 디바이스 드라이버의 제작 과정에서 디버깅 기능이 필수적으로 필요하지만, 제조사 별 하드웨어 스펙 및 장착된 내부 기기, 센서의 종류와 조합이 다양하여 일관된 시스템 분석 도구를 활용해 시스템 인스펙션 업무를 수행하는데 어려움이 따른다. 본 논문에서는 이러한 어려움을 해결하기 위해 임베디드 펌웨어, 디바이스 드라이버 환경에 활용되고 있는 다양한 시스템 인스펙션 도구 및 시스템 모니터링 소

‡ 교신저자

본 연구는 IITP 정보보호국제공동연구과제의 지원(RS-2022-00165794, 60%) 및 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업(IITP-2022-2021-0-01816, 20%)및 2020년도 한국연구재단(NRF) 연구과제의 지원(NRF-2020R1A2C4002737, 20%) 을 받아 수행된 연구임.

프트웨어를 분석하고 이를 클라우드에 탑재시키기 위한 분석을 수행하여 클라우드 기반 인스펙션 오케스트레이션 시스템을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 시스템 인스펙션 도구 및 시스템 모니터링 소프트웨어를 분석하고 3장에서는 클라우드 기반의 인스펙션 오케스트레이션 시스템을 제안한다. 이를 바탕으로 4장에서 결론을 도출한다.

II. 시스템 인스펙션 도구 및 시스템 모니터링 소프트웨어 분석

대부분의 임베디드 시스템에서 사용되는 운영체제는 임베디드 리눅스이다[2]. 임베디드 리눅스의 배포판 종류에는 Android SDK, BuildRoot, OpenWRT, ubuntu 등이 있으며 높은 보안성, 안정된 구조, 유지보수의 용이성, 다양한 하드웨어 이식성 등의 강점을 보인다. 본 논문은 임베디드 리눅스 환경에서 사용할 수 있는 디버깅 유틸리티인 gdb와 커널 성능 모니터링 도구인 perf, 하드웨어 디버깅 장비인 JTAG를 분석하고자 한다.

2.1 gdb(GNU Project Debugger)

gdb는 GNU 소프트웨어 시스템용 디버거 유틸리티로, 유닉스 기반의 여러 시스템에서 동작이 가능하다. 임베디드 리눅스에서 사용하기 위해서는 포팅과 셋업 과정이 필요하며, 패키지에 내장되어 있는 gdbserver를 사용해 원격으로 디버깅할 수 있다. 호스트 시스템에서 gdb를 실행시키고 타겟 시스템에서 gdbserver를 실행시키면, 상대적으로 메모리가 부족한 타겟 시스템에서의 원격 디버깅이 가능해진다[3]. gdb는 장애 발생 원인을 찾기 위한 메모리 덤프 생성, 유지보수를 위한 실시간 메모리 덤프 분석에 사용된다. 또한 프로세스 테스트를 위한 컴파일에도 활용되며, 내부에 브레이크 포인트를 설정하고 스택을 확인하는 등 전반적인 프로세스의 실행과 디버깅 사용된다. gdb는 메모리 덤프, 레지스터 검사, 스택 분석 등의 기능을 아래의 [표 1]과 같이 지원한다.

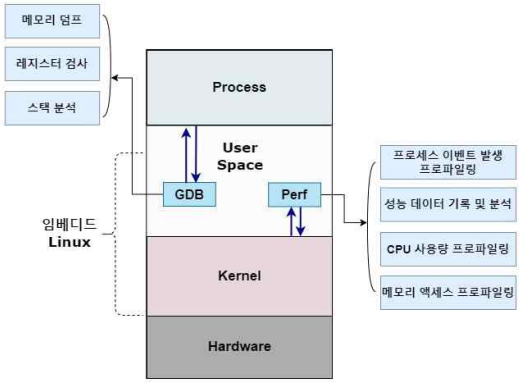
그러나 임베디드 시스템에서 gdb를 사용하려면 매번 원격 연결이 필요하며, 타겟 시스템이 고장난다면 수리하기 전까지 개발을 진행할 수 없다.

2.2 perf

perf는 리눅스 커널 성능 측정 도구이다. 커널에서 perf 명령어를 사용하여 시스템 이벤트를 수집할 수 있고, 시스템 전반의 성능을 분석하여 실시간 분석 리포트를 얻을 수 있다[4]. perf는 CPU 사용량 프로파일링, 메모리 액세스 프로파일링, 프로세스 이벤트 발생 프로파일링, 성능 데이터 기록 및 분석 등의 기능을 아래의 [표 1]과 같이 지원한다.

[표 1] gdb, perf 기능 및 내용[5]

종류	기능	내용
gdb	메모리 덤프	메모리 덤프 생성 통한 메모리 누수 원인 파악 가능
	레지스터 검사	레지스터 값 확인 및 변경
	스택 분석	콜스택 추적을 통해 프로세스 비정상 종료 원인 파악
perf	CPU 사용량 프로파일링	- perf top 명령어로 실시간 CPU 사용량 프로파일링 가능 - 사용하고 있는 프로그램, 라이브러리 이름, 사용 중 CPU 백분율 출력 - 커널 영역에서의 실행 여부 [K] 기호로 식별
	메모리 액세스 프로파일링	- perf kmem 명령어로 시스템의 메모리 액세스 프로파일링 가능 - 수집된 샘플 백분율 출력 - 메모리 샘플 개수, 액세스 대기 시간, 메모리 액세스 유형, 버스 트랜잭션, TLB 메모리 액세스, 메모리의 Lock 여부 출력
	프로세스 이벤트 발생 프로파일링	perf stat 명령어로 프로세스 이벤트 발생 수의 전반적 통계 출력
	성능 데이터 기록 및 분석	- perf record 명령어로 커널 내의 전반적인 성능 데이터 기록 - perf report 명령어로 기록된 내용을 분석



[그림 1] gdb 및 perf의 분석 영역 비교

위 [그림1]은 gdb와 perf의 기능을 분석 영역에 매핑한 그림이다. gdb는 프로세스 영역에 위치한 소프트웨어를 디버깅하며, perf는 커널 영역의 성능을 모니터링한다.

2.3 JTAG(Joint Test Action Group)

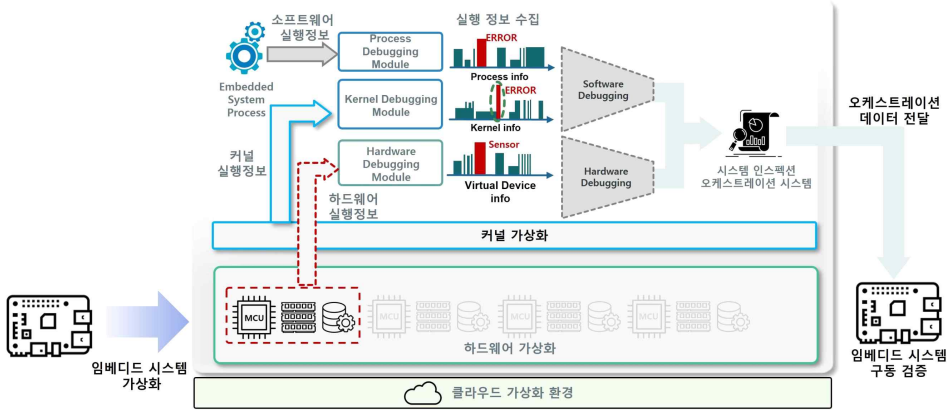
JTAG는 임베디드 시스템 개발에 사용되는 하드웨어 테스트 및 소프트웨어 디버깅 장비이다[6]. 디바이스의 외부 핀을 구동시키고 값을 읽어들일 수 있으므로 wifi 통신 모듈이 없는 임베디드 시스템에서도 사용할 수 있다. 임베디드 시스템에 JTAG을 연결한 뒤 OpenOCD를 사용하면 실시간 디버깅 수행이 가능해지며, 개발자가 CPU를 직접 조절하며 디버깅할 수 있다. 또한 타겟 시스템의 자원을 소모하지 않아 디버깅으로 인한 자원 소모가 없다.

그러나 디버깅 소프트웨어를 사용하지 않은 체 JTAG만으로 디버깅한다면, 핀을 조작해 디바이스 자체에서 출력값을 확인할 수밖에 없다. 또한 JTAG의 데이터 관찰 및 전송을 담당하는 boundary scan 기능을 사용하면 프로세서의 내장 펌웨어를 탈취할 수 있다[7].

III. 클라우드 기반 시스템 인스펙션 오케스트레이션 시스템

현재 임베디드 시스템은 소프트웨어, 하드웨어 각각의 디버깅 툴을 이용해야 하며 이는 개발 환경의 효율성을 저하시키고 있다. 장치마다 사용할 수 있는 모듈을 개발하고 수정해야 하는데, 이 과정에서 매번 디버거를 원격 연결한 뒤 디버깅해야 하는 번거로움이 있기 때문이다. 이러한 문제에 더불어 2장에서 언급한 디버거의 한계를 해소하기 위해 스택과 레지스터를 단계적으로 분석할 수 있는 gdb의 기능, 프로세스 이벤트 발생 프로파일링 및 커널 내의 전반적인 성능을 모니터링하는 perf의 기능, 하드웨어를 디버깅할 수 있는 JTAG의 기능이 조합된 클라우드 기반의 새로운 디버깅 시뮬레이션 시스템을 제안하고자 한다.

클라우드 기반 인스펙션 오케스트레이션 시스템의 구조는 아래의 [그림2]와 같다. 사용자는 클라우드 가상화 환경에서 임베디드 시스템을 가상화한다. 위 환경에서 하드웨어 실행정보는 하드웨어 디버깅 모듈을 사용하고, 커널 실행정보는 커널 디버깅 모듈을 사용하며, 소프트웨어 실행정보는 프로세스 디버깅 모듈을 사용해 분석 및 수정이 가능하다. 디버깅이 완료된 시스템 오케스트레이션 정보를 임베디드 시스템에



[그림 2] 클라우드 기반 인스펙션 오케스트레이션 시스템 구조

업로드하고 구동을 검증할 수 있다.

클라우드 환경에서 소프트웨어와 커널, 하드웨어를 디버깅하는 서비스를 제공한다면 공간에 제약받지 않고 디버깅할 수 있으며, 하드웨어와 소프트웨어 분석 상황을 동시에 모니터링을 할 수 있다. 이는 하드웨어와 소프트웨어를 동시에 모니터링하고 디버깅하는 것이 불가능한 기존 방식에 비하여 개발자의 편리성을 증진할 수 있다. 또한 임베디드 가상화 환경을 구축하여 위 시스템과 함께 활용한다면, 매번 원격 연결을 하지 않아도 되며 임베디드 시스템에 펌웨어를 넣지 않고 클라우드 기반 시스템 인스펙션 오케스트레이션 시스템을 사용해 테스트할 수 있으므로 개발 기간을 단축할 수 있으며 번거로움을 줄일 수 있다. 본 논문에서 제안하는 시스템은 QEMU에서 지원하는 가상화 하드웨어인 bamboo, mpc8544ds, ppce500, ref405ep, sam460ex, virtex-ml507 보드에 활용할 수 있다[8]. 또한 이러한 환경을 통해 시뮬레이션 구동을 하여 임베디드 개발 환경의 디버깅이 가능하기에 개발 환경의 편리성과 분석 효율성을 높일 수 있으며, 단일 디버거 사용시의 한계점을 최소화할 수 있다.

IV.결론

본 논문에서는 임베디드 펌웨어, 디바이스 드라이버 환경에 활용되고 있는 시스템 인스펙션 도구 및 시스템 모니터링 소프트웨어를 분석하고, 앞서 조사한 기능을 통합하여 임베디드 시스템에 활용할 수 있는 클라우드 기반의 시스템 인스펙션 오케스트레이션 시스템을 제안하였다. 스택 및 레지스터 단계적 분석 기능, 프로세스 이벤트 발생 프로파일링 및 커널 내의 전반적인 성능 모니터링 기능, 하드웨어 디버깅 기능을 통합한 소프트웨어를 클라우드 서비스로 제공한다. 이에 더불어 임베디드 가상화 환경을 구축하여 클라우드 내에서 펌웨어 및 디바이스 드라이버를 제작하도록 한다.

현재 QEMU에서 가상화 환경으로 제공하는 보드는 시스템에 활용 가능하지만, 하드웨어 모듈의 디버깅은 불가능하며 부품 자체에서 발생된 물리적 손상을 클라우드 환경에서 확인할 수 없다. 향후 연구에서는 하드웨어 모듈을 가상화하고, 실제 하드웨어 모듈의 물리적 손상 상황을 가상화 환경에 반영할 수 있는 방법을 모색해야 한다.

[참고문헌]

[1] 김병철, et al. "임베디드 소프트웨어 개발을 위한 JTAG 기반의 디버깅 도구." 한국정보과학회 학술발표논문집 31(1A) (2004): 943-945.

[2] 이형석, 정영준. "임베디드 운영체제 커널 기술 동향." [ETRI] 전자통신동향분석, 21(1) (2006): 0-0.

[3] 심현철, 강용혁, 엄영익. "리눅스 환경에서의 다중 프로세스 응용에 대한 원격 디버깅 기법." 정보과학회논문지: 컴퓨팅의 실제 및 레터 8(6) (2002): 630-638.

[4] 박진영. "PMU 기능이 없는 임베디드 리눅스 시스템에서 샘플링 기반 프로파일러의 성능 평가 연구." 국내석사학위논문 한양대학교 대학원, 2018. 서울

[5] Red Hat Customer Portal: https://access.redhat.com/documentation/ko-kr/red_hat_enterprise_linux/8/html/monitoring_and_managing_system_status_and_performance/common-perf-commands_getting-started-with-perf

[6] 이건하. "임베디드 시스템에서 안전성 확보를 위한 JTAG의 특성 분석 및 실험." 국내석사학위논문 한양대학교 대학원, 2022. 서울

[7] 이건하, 공원배, 정혜민, 전지원, 김동규. JTAG 신호분석을 이용한 상용 MCU 해킹 취약성 연구. 전자공학회논문지, 58(12) (2021): 19-26.

[8] QEMU Embedded family boards: https://www.qemu.org/docs/master/system/ppc/embedded.html