TrustZone을 활용한 스택 카나리 보완 기법

박재열[1], 박성환[2], 권동현[3]
[1]부산대학교 정보컴퓨터공학부 학부생
[2]부산대학교 정보융합공학과 박사과정
[3]부산대학교 정보컴퓨터공학부 교수 (교신저자)
woduf0628@pusan.ac.kr, starjara@pusan.ac.kr, kwondh@pusan.ac.kr

A Study on Stack Canary Security Enhancement Techniques Using TrustZone

Jae-Yeol Park[1], Seong-Hwan Park[2], Dong-Hyun Kwon[3]
[1]Dept. of Computer Science and Engineering, Pusan National University
[2]Dept. of Information Convergence Engineering, Pusan National University
[3]Dept. of Computer Science and Engineering, Pusan National University

요 약

다양한 방면에서 사용되는 임베디드 시스템의 메모리 취약성에 대한 관심이 많아짐에 따라 임베디드 시스템의 메모리 보호와 관련하여 많은 연구가 진행 중이다. 스택 카나리는 효율적인 메모리 보호 기법으로써 널리 사용되지만 물리 메모리가 제한적이고 사용자 권한 분리를 지원하지 않는 임베디드 시스템에서는 기존 방식을 활용한 스택 카나리를 적용하는 것에 한계가 있다. ARM의 TrustZone은 일반 실행 환경과 신뢰 실행 환경으로 분리하여 일반 실행 환경에서 신뢰 실행 환경의 데이터나 코드에 접근하지 못 하도록 막는다. 그렇기 때문에 ARM의 TrustZone에 암호화 키를 저장하거나 보안이 중요한 동작을 TrustZone에서 실행하는 연구가 많다. 본 논문에서는 ARM의 TrustZone을 활용하여 임베디드 시스템에서 스택 카나리 기법의 한계를 보완 할 수 있는 방법을 제안한다.

1. 서론

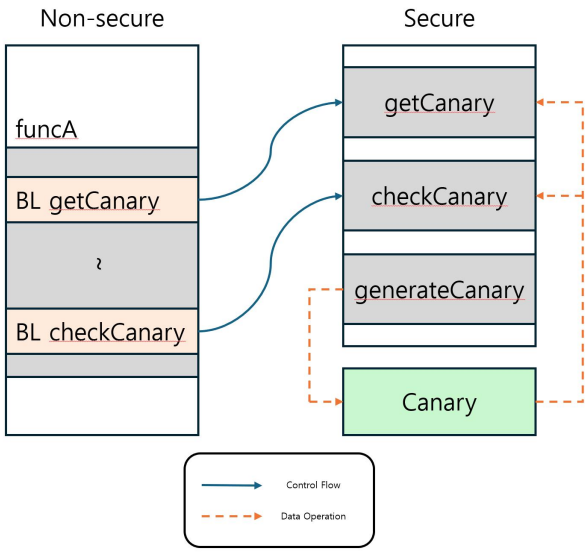
현대 사회에서는 스마트폰, 인공지능 스피커, 드론, 차량 등 다양한 용도로 마이크로 컨트롤러가 사용된다.[1] 이렇게 사용 분야가 넓어짐에 따라 ARM Cortex-M과 같은 마이크로 컨트롤러가 탑재된 임베디드 시스템에서의 소프트웨어 보안 문제도 주목받고 있다. 특히 임베디드 소프트웨어는 메모리 취약성이 있는 C/C++ 언어로 주로 작성이 되기 때문에 메모리 안전성 문제를 해결하기 위한 보안 연구가 많이 진행되고 있다.[2]

스택 카나리는 메모리 취약성을 완화하는 보호 기법 중 하나로 스택에 임의의 값을 넣어 해당 값의 변조 여부를 판단하여 공격을 탐지한다.[1] 낮은 성능 부하와 구현이 간단하다는 특징이 있어 많이 활용된다. 그러나 임베디드 시스템에서는 임베디드 시스템이 가진 한계 때문에 스택 카나리를 기존 방식대로 적용하기에 문제가 있다. 따라서 본 논문에서는 마이크로 컨트롤러가 사용된 임베디드 시스템에서 스택 카나리를 적용할 때의 문제를 확인하고, ARM에서 제공하는 TrustZone을 활용해 스택 카나리 기법을 보완 할 수 있는 방법 TZ-Canary를 제안한다.

2. 임베디드 시스템에서 스택 카나리

x86 시스템에서는 커널이 랜덤한 값을 Thread Local Storage(TLS)라는 자료 구조의 카나리 값 변수에 저장하고 fs 세그먼트 레지스터가 TLS의 주소를 가리킨다. 카나리는 fs 레지스터에 저장된 주소에서 카나리를 가져와 설정한다. 임베디드 시스템에는 카나리 값이 저장되는 특정 세그먼트 레지스터가 없기 때문에 보통 컴파일 시에 컴파일러가 임의의 값을 생성하거나 부팅 시에 랜덤한 값을 생성해서 메모리의 특정 위치에 저장하게 된다.

임베디드 시스템에서는 가상 메모리를 지원하지 않고 메모리에서의 권한 분리가 되어 있지 않기 때문에 물리 메모리에 직접 카나리 값을 저장하고 관리하게 되면 공격자가 카나리 값을 조회하거나 임의의 값으로 덮어쓰게 되어 원본 카나리 값에 대한 기밀성과 무결성을 보장할 수 없다는 문제가 있다.[1]



(그림 1) TZ-Canary Overview

3. TZ-Canary

ARM에서는 중요한 데이터, 펌웨어 등 보안이 필요한 정보를 안전하게 보호하기 위해 TrustZone이라고 하는 일반 수행 영역과 분리되어 동작하는 신뢰 실행 환경을 제공한다.[3] 스택 카나리를 이 TrustZone 안에서 생성하고 저장하게 되면 일반 수행 영역에 있는 공격자는 TrustZone 내부에 있는 카나리 값에 접근할 수가 없어 안전하게 카나리 생성, 저장 및 검증 동작을 수행할 수가 있다.

TZ-Canary는 카나리 생성, 카나리 설정, 카나리 검증 총 세 단계로 구성된다.

(1) 카나리 생성 : 부팅 시 TrustZone에 있는 안전한 부트로더에서 랜덤한 값을 만들어 TrustZone 내의 안전한 메모리 영역에 원본 카나리 값을 저장한다.

(2) 카나리 설정 : 컴파일러 수정을 통해 기존에 카나리 값을 가져와 스택에 위치시키던 프롤로그 코드 사이에 카나리 값을 가져오는 함수 호출 코드를 추가한다. 해당 코드는 신뢰 실행 환경으로 월드 스위칭을 한 후 (1) 단계에서 저장된 카나리 값을 가져온다. 함수 호출이 완료된 후 가져온 값을 스택에 위치시킨다.

(3) 카나리 검증 : 컴파일러 수정을 통해 에필로그에 있던 카나리 검증 코드 사이에 카나리를 저장했던 위치에서 값을 가져와 신뢰 실행 환경에서 값을 검증하는 함수를 호출하는 코드를 추가한다. 만약 (1) 단계에서 저장한 값과 스택에서 가져온 값이 다르다면 신뢰 실행 환경에서 동작 중인 소프트웨어가 일반 실행 환경의 소프트웨어를 강제로 종료시키며 예외를 발생시킨다.

세 단계 코드를 각각 부트로더, 함수별 프롤로그와 에필로그에 추가하면 TruztZone에서 무결성과 기밀성이 보장된 카나리 기술 사용이 가능하다.

4. 결론

널리 사용되는 메모리 보안 기술인 스택 카나리 기법이 임베디드 시스템에 사용되면서 한계를 가지게 되고 이에 따라 한계를 보완할 필요가 있다. 본 논문에서는 TrustZone을 활용하여 임베디드 시스템의 스택 카나리가 가지는 기밀성과 무결성 부족에 대한 해결 방안을 제안하였다. 추후 구현을 통해 하드웨어를 활용한 성능 및 보안 평가가 진행될 예정이며, 이를 통해 TrustZone을 활용하여 비교적 적은 성능 부하를 가지고 메모리 보호가 가능함을 보일 수 있을 것으로 기대된다. TZ-Canary의 성능 부하는 일반 실행 환경에서 신뢰 실행 환경으로의 월드 스위칭으로 인한 성능 부하가 주요 원인으로 예상된다. 월드 스위칭 부하는 TrustZone 사용으로 인해 반드시 생기기 때문에 TZ-Canary의 부하를 줄이기 위해서는 Canary 생성, 설정, 검증 단계의 코드가 최소화되도록 최적화가 진행되어야 한다.

Acknwledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2024-2020-0-01797)

참고문헌

[1] Tan, Xi, et al. "Is the Canary Dead? On the Effectiveness of Stack Canaries on Microcontroller Systems." ACM/SIGAPP Symposium On Applied Computing (SAC). 2024.

[2] Tan, Xi, et al. "Where's the" up"?! A Comprehensive (bottom-up) Study on the Security of Arm Cortex-M Systems." arXiv preprint arXiv:2401.15289 (2024).

[3] Yeo, Gisu, et al. "Efficient CFI Enforcement for Embedded Systems Using ARM TrustZone-M." IEEE Access 10 (2022): 132675-132684.