아두이노 임시파일을 이용한 메모리 초기화에 관한 연구

이우호*, 강성민**, 임채상**, 노봉남**
*전남대학교 정보보안협동과정
**순천대학교 정보통신공학과
e-mail: leeouho@naver.com

Research on Memory Initialization through Using Arduino Temporary Files

Woo-Ho Lee*, Sung-Min Kang**, Chae-Sang Lim**, Bong-Nam Noh**
*Dept of Computer Science, ChonNam University
**Dept of Computer Engineering, *Sunchoen University

요 약

사물인터넷은 기존의 여러 ICT기술과 유,무선 장비의 네트워크 및 다양한 통신 기술들이 적용된 것을 의미한다. 최근 다양한 사물 인터넷에 대한 발전은 경량화, 소형화 되어가며 OSHW(Open Source HardWare)을 기반으로 점차 다양화 되었다. 이에 따라 사물인터넷 다양한 디바이스와 펌웨어, 암호학에 대한 연구는 활발하게 진행되고 있다. 하지만 특정 하드웨어의 센서 디바이스에 대한 연구는 부족하다. 본 논문에서는 OSHW 중 하나인 AVR기반의 아두이노 개발도구에 대한 임시 파일에 대해 분석하고 메모리 초기화 방법에 대해 제안한다. 또한 임시파일을 이용한 메모리 초기화 방법을 이용하여 사용자정보와 메모리 공격에 대한 데이터 유출을 방지할 수 있다.

1. 서론

최근 사물인터넷은 일상적인 생활을 편리하게 할 뿐 만 아니라 홈, 자동차, 의료등의 다양한 분야에서 적용되고 있으며, 점차 그 분야는 넓어지고 많은 기기들의 결합이 진행되고 있다. 국내뿐만 아니라 국외기업들의 사물인터넷에 대한 많은 투자와 기술 연구가 증가되고 있으며 사물인터넷 분야는 사용자의 전자제품과 편의시설에 대한 직접적 접근(e.g. 냉장고, 자동차, 의료등)이 가능하여 공격자에 의한 보안에 관한 위험성은 높아지고 있다. 사물인터넷에서 가장 큰 문제는 사물인터넷을 이용한 기기의 오작동을 일으킬 수 있는 공격이다.

하지만 사물인터넷의 기기별 특성은 다양하고 그만큼 보호해야할 기기는 다양하다. 기기별 특성에 따라 암호화, 저전력화, 연결성이 강조되고 있다. 기존의 한계를 나타내는 기술 중 암호화와 네트워크에 관한 기술은 현재 많은 연구가 진행되고 있으나 기초연구인 하드웨어(센서, 모터)를 직접적으로 제어하고 각각의 센서 디바이스에 대한 연구는 부족하다.[1]

OSHW(Open Source HardWare)가 증가함에 따라 센서 디바이스에 대한 종류와 저 사양 기기 사용이 늘어나므로써, 저 사양 하드웨어에 대한 보안 적용과 기기별 보안에 대한 업데이트 또한 어려워지고 있다.

본 논문에서는 OSHW의 공통적으로 들어가는 SRAM, 플래쉬 메모리에 대한 아두이노의 소프트웨어를 이용한 메모리를 초기화하는 방식에 대해 제안한다.

2. 아두이노 하드웨어 분석

아두이노(Arduino)는 MCU(Micro Controller Unit)의 종류 중 하나인 AVR은 아트멜(Atmel)에서 개발되었다. 아두이노의 종류는 MCU에 따라 수십 가지에 이른다. 다음 (그림1)은 아두이노의 종류와 MCU에 따른 분류이다.

	MCU	Arduino Board
AVR	ATmega168	pro(168),Mini(168),LilyPad(168v)
	ATmega328	Uno,ETHERNET,Flo.Nano,Pro(328),Pro Mini, LilyPad/Simple(328v)
	ATmega2560	Mega 2560, Mega ADK
	ATmega32U4	YUN, Leonardo, Esplora, Micro, LilyPad USB, ROBOT(Control,Motor)
	ATtiny85	GEMMA
ARM	Cortex-M0+	Zero, Zero Pro, M0,M0 PRO
	Cortex-M3	Due

(그림 1) 아두이노 보드

아두이노의 AVR계열 MCU 종류는 크게 AVR Tiny와 MEGA AVR, 그리고 AT90 AVR로 나눌 수 있다. 그 중 대표적인 디바이스 중 하나인 아두이노 보드는 (그림 1)에서 보는 것과 같이 주로 MEGA AVR를 위주로 사용한다.

MCU	Arduino Board	EEPROM	SRAM	FlashMemory
Atmega328	UNO, Uno Ethernet, Menta, Boarduino	1Kbyte	2Kbyte	32Kbyte
Atmega32U4	Leonardo, Flora, Micro, Teensy, 32U4 Breakout	1Kbyte	2.5Kbyte	32Kbyte
Atmega2560	Mega, MegaADK	4Kbyte	8Kbyte	267Kbyte

(그림 2) 아두이노 보드 별 메모리 구성[2]

위의 (그림2)는 아두이노 보드의 메모리에 대한 분류이며, EEPROM, SRAM, 플래쉬 메모리로 구성된다. SRAM은 휘발성 메모리이며, 전원 공급이 끊기게 되면 데이터는 삭

제된다. 메모리를 분석한 결과 아두이노의 SRAM은 플래쉬 메모리와 데이터를 공유하며, Arduino Studio IDE를 이용한 프로그램 빌드 시 각종 필요한 변수와 버퍼 등이 생성될 때 저장되는 공간이다.

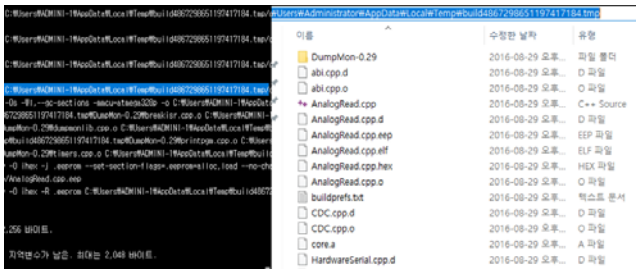
아두이노 보드(Board)는 Reset 스위치를 이용한 메모리 초기화를 지원한다. 다음 (그림 3)은 하드웨어 디바이스 초기화에 따른 메모리 상태이다.

(그림 3) 디바이스 데이터 초기화 전/ 후

하드웨어 초기화 버튼을 이용한 초기화는 아두이노 메모리에 대한 데이터를 초기화하며, EEPROM의 빌드 파일 역시 초기화되어 다시 빌드 해야 된다. 한편 하드웨어 초기화를 진행하여도 초기화 되지 않은 데이터가 남는 것을 확인할 수 있다. 본 논문에서는 직접적인 하드웨어 메모리 초기화 방법의 해결방안으로 소프트웨어에서 빌드파일을 이용한 초기화 방법을 제시한다. 실험환경은 windows7 운영체제와 Arduino Studio IDE v1.4.2 환경에서 실험하였으며 제시하는 초기화 방법은 다음과 같다.

3. 아두이노 임시파일을 이용한 메모리 초기화

아두이노는 빌드과정에서 임시파일을 이용하여 디바이스에 소스코드가 업로드 된다. 임시파일은 windows 사용자 임시 디렉터리(C:\Users\사용자명\AppData\Local\Temp)에 저장된다. 임시 디렉터리 아래 프로젝트 명칭의 알파벳명칭+임시번호.tmp 디렉터리로 저장된다. 임시디렉터리에 저장된 hex파일을 직접 ICSP 업로드 장치를 통해 업로드 할 수 있다. (그림 4)의 왼쪽 그림과 같이 아두이노의 임시파일경로는 쉽게 획득할 수 있다. 파일 구성은 (그림 4) 오른쪽 그림과 같은 형태를 가진다.



(그림 4) 왼쪽/오른쪽 빌드경로, 임시파일 구성

임시 디렉터리 속의 buildprfs.txst 파일 안에는 빌드를 위한 소스코드와 개인정보(Wi-Fi ID/PW, H/W ID), 빌드정보가 저장되어 있으며, 메모리 덤프를 이용해 쉽게 획득할 수 있다. 또한 공격자는 악성코드를 이용하여 하드웨어 대한 메모리를 조작할 수 있다.

본 논문에서는 생성한 프로그램을 이용해 임시파일을 변조하여 공격자의 메모리 덤프를 막기 위해 플래쉬 메모리를 특정 문자로 데이터를 초기화 하였다. 초기화 한 메모리를 소프트웨어 메모리 덤프 프로그램(Dumpmon)을 이용하여 메모리 덤프 된 내용을 확인하였으며, 마지막으로 플래쉬 메모리에서 빌드파일에 대한 정보와 사용자 정보 및 개인정보, 하드웨어 ID에 대한 누출을 막을 수 있었다.
(그림 5)는 데이터 파일 수정 전/ 후의 메모리 데이터를 확인한 결과이다.

(그림 5) 업로드 후 메모리 확인 수정 전/후

4.결론

본 연구에서는 아두이노의 임시파일을 이용하여 소프트웨어를 이용한 메모리 초기화 방법을 제안했다. 이 방법은 하드웨어 초기화 버튼을 이용한 초기화에 비해 소스코드에 대한 정보를 유지할 수 있으며 메모리덤프를 이용한 개인정보(Wi-Fi ID/PW, H/W ID), 빌드정보의 누출을 막을 수 있다.

OSHW(Open Source HardWare)가 증가함에 따라 저 사양 기기에 대한 보안이 중요해지고 있다. 대부분의 저 사양 기기는 헤드리스(headless devices) 기기로써 소량의 코드만으로 쉽게 감염될 수 있기 때문이다[3]. 본 연구에서 제안한 소프트웨어를 이용한 메모리 초기화 방법이 그 방어방법 중 하나로 적용될 것으로 기대된다.

저 사양 기기는 OTA(Over-The-Air)를 통해 업데이트 된다. 업데이트 과정에서 공격자에 의한 공격에 대해 대비하여 사물인터넷 전송구간 암호화에 대한 많은 기초연구가 필요하다.

참고문헌

[1] 류호석 "IoT환경에서 안전한 스마트홈 체계 구축을 위한 보안 아키텍처에 관한 연구" 3rd Ed. McGraw Hill

[2] 양대엽 "메모리 초기화를 이용한 사용자 데이터 유출 방지에 관한 연구" 전자 공학회 논문지 .2012

[3] http://playground.arduino.cc/Learning/Memory

[4] Alberca, Carlos, et al. "Security Analysis and Exploitation of Arduino devices in the Internet of Things." Proceedings of the ACM International Conference on Computing Frontiers. ACM, 2016.