JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 5 : 1003~1008 May. 2017

ARMv7 Thumb Architecture 취약성 분석

김시완 · 성기택*

Vulnerability analysis on the ARMv7 Thumb Architecture

Si-Wan Kim · Ki-Taek Seong*

Department of Information Security, Tongmyong University, Pusan 48520, Korea

요 약

최근 몇 년간 사물인터넷은 중요한 연구적 관심을 끌어왔다. 새로운 IoT 기술이 널리 이용되기 위해서는 정보의 신뢰성과 보호가 전적으로 요구된다. IoT 시스템은 그 특성상 직접적인 접근이 쉬우므로 이로 인한 물리적인 보안에 매우 취약하다. SoC 기술의 발달과 함께 운영체제에 대한 기술도 많이 이루어졌으며 많은 새로운 운영체제가 소개되고 있다. 본 연구에서는 ARMv7 Thumb Architecture 하드웨어 플랫폼에서 동작하는 운영체제에 대한 취약성분석 결과에 대하여 기술하였다. 최근에 소개된 “Windows 10 IoT Core” 운영체제에 대하여, 연구를 통하여 개발된 침투코드를 특정 IoT 시스템에 이식시켜 Zero-Day Attack을 구현하였다. 결과의 침투코드에 대한 바이러스 검출 여부를 “virustotal” 사이트에 의뢰하여 유효성을 입증하였다.

ABSTRACT

The Internet of Things has attracted considerable research attention in recent years. In order for the new IoT technology to be widely used, the reliability and protection of information is required. IoT systems are very vulnerable to physical security due to their easy accessibility. Along with the development of SoC technology, many operating systems have been developed and many new operating systems have been introduced. In this paper, we describe the vulnerability analysis results for operating systems running on the ARMv7 Thumb Architecture hardware platform. For the recently introduced “Windows 10 IoT Core” operating system, I implemented the Zero-Day Attack by implanting the penetration code developed through the research into a specific IoT system. The virus detection test for the resulting penetration code was validated by referral to the “virustotal” site

키워드 : 사물인터넷 시스템, ARMv7 Thumb 구조, 침투테스트, 제로 데이 공격, 로컬 시스템 해킹

Key word : IoT Systems, ARMv7 Thumb Architecture, Penetration Testing, Zero-Day Attack, Local System Hacking

Received 17 February 2017, **Revised** 14 March 2017, **Accepted** 27 March 2017

* **Corresponding Author** Ki-Taek Seong(E-mail:ktseong@tu.ac.kr, Tel:+82-51-629-1282)
Department of Information Security, Tongmyong University, Busan 48520, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.5.1003 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 현대 사회의 인터넷 사용 확산에 따른 각종 정보보안의 사건·사고가 빈번히 발생하고 있다. 이에 따라, 정부·공공기관 및 기업들은 '정보보안'이라는 사회적 이슈에 발맞춰 '조직의 자산'을 보호하기 위해 여러 방면으로 노력하고 있다[1,2].

이러한 정보보안 사건·사고의 유형에는 공통된 유사점이 있다. 바로, 조직 내의 '내부자(사람)'에 의해 정보보안 사건·사고가 발생한다는 점이다. 즉, 조직 내부에 불순한 의도를 가지고 접근하는 '악의적 내부자'와 조직 구성원의 실수와 방심에 의한 '부주의적 내부자' 같은 유형으로 나타나며, 전체 정보보안 사건·사고 중 80%이상이 이러한 유형의 '내부자'에 의해 발생된다는 점이 매우 흥미롭다. 특히, 내부자에 의한 정보보안 사건·사고는 E-Mail Phishing 또는 Spear Phishing과 같은 'Social Engineering' 공격기법으로 인간의 심리적 성향과 신뢰성 관계를 공격하므로, 목표대상이 된 조직의 내부자는 속수무책으로 당하고 있는 것이 현실이다. 대표적인 사건으로 2011년 4월 12일에 발생한 '농협 전산망 마비 사태'와 2016년 7월 28일에 발생한 '㈜인터파크 기업의 개인정보 유출사건'은 조직 내부자에 대한 직접적 공격(로컬 시스템 공격)의 심각성을 보여주는 단편적인 지표이다. 우리는 여기서 한발 더 앞서 나아가, 최근 차세대 IT 환경으로서 각광받고 자리매김 하고 있는 'IoT(Internet of Things) System'은 과연 '내부자에 의한 공격과 같은 로컬 시스템 공격의 위협으로부터 자유로울 수 있을까'라는 의문을 제기할 수 있다. 왜냐하면, IoT System의 특성상 IoT Device (혹은 Sensor)는 우리의 일상생활에 쉽게 노출(또는 장착)되어있고, IoT Device에 대한 직접적인 물리적 접근이 쉬우므로, 이로 인한 '물리적 보안'이 매우 취약하기 때문이다.

본 연구에서는 최근에 발표된 IoT System ARMv7 Thumb Architecture를 대상으로 취약성 분석 연구를 수행하였다. ARMv7 Thumb Architecture를 대상으로 직접적인 로컬 시스템 취약성을 공격하여 IoT System 속으로 침투하는 Zero-Day Attack을 시연하였다. 시연을 위하여 개발된 expolit code에 대하여 백신사이트를 통한 검증을 통하여 유효성을 확인하였다.

Ⅱ. 관련 연구

2.1. IoT System 운영체제 소개

IoT System에서 구동되어지고 사용되는 운영체제들은 ARM Architecture로 널리 포팅 되어 사용되고 있다. IoT System 환경 특성상, 소형화, 휴대성이 강조되는데 임베디드 환경에 특화되어 저전력을 사용하도록 설계된 ARM Architecture는 IoT System 운영체제에 최적화되어 있기 때문이다. IoT System의 운영체제는 보편적으로 'Windows계열'과 'Linux계열'의 두 분류로 구분할 수 있다. 'Windows계열'로 우리가 흔히 쉽게 접할 수 있는 Microsoft사의 Windows 10을 기반으로 IoT System 환경에 최적화된 운영체제로 개발된 'Windows 10 IoT Core'를 상용화하였다(그림 1 참조).

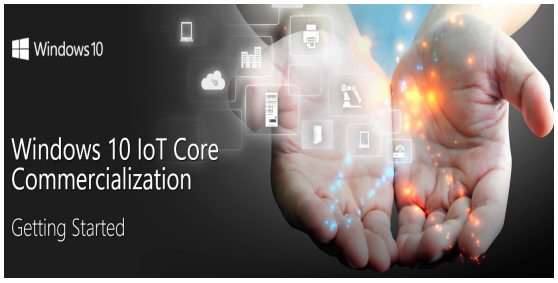

Fig. 1 Windows 10 IoT Core in Commercialization Phase

Windows 10 IoT Core 운영체제는 Microsoft사에서 전 세계의 IoT System 환경에 대한 상용 촉진 목적으로 라이선스 비용을 무료하여 배포하고 있다. 또한, Microsoft 홈페이지의 다양한 포럼을 통해 IoT System 관련 어플리케이션 개발에 대한 지원과 피드백을 받을 수 있다. 다양한 컨텐츠와 Visual Studio 2015 같은 개발자도구를 적극 지원하므로 앞으로의 IoT System 시장에서 IoT System 운영체제로서 주요 역할을 하고자 할 것이다[3]. 그 다음으로, 서버환경에서 자주 쓰이는 'Linux계열'의 IoT System 운영체제들이다. 기본적으로 Linux 운영체제는 추구하는 이념과 같이 오픈소스로 무료로 이용가능하고, 전 세계적으로 다양하게 개발되고 있는 점이 큰 장점이다. 또한, 일반적인 PC환경에서 사용되고 있는 Linux 운영체제와 비슷한 인터페이스와 구동방식으로 개발되어 거부감이 접근할 수 있다.

다만, IoT 디바이스의 특성상 플랫폼이 ARM Architecture인 점이 일반적인 PC환경의 Linux 운영체제와 가장 큰 차이점이라고 볼 수 있다. Cent OS, Fedora, Ubuntu 등 대부분의 Linux 개발팀에서 ARM Architecture로 포팅 하여 배포하고 있기 때문에 원하는 환경에서 요구되는 상황에 맞게 IoT System을 이용할 수 있다. 이와 같이 Linux 계열 IoT 운영체제는 ARMv7 Architecture 기반 베이스로 포팅 되어 개발되고, Windows 계열인 Windows 10 IoT Core은 ARMv7 Thumb Architecture로 포팅 되어 개발되었다.

기술적으로 ARMv7 Architecture에 대한 침투테스트 관련 자료는 간혹 존재하나, ARMv7 Thumb Architecture에 대한 침투테스트 관련 자료는 없다[4,5].

본 연구에서는 ARMv7 Thumb Architecture 기반인 Windows 10 IoT Core를 대상으로, 로컬 시스템 취약점을 분석하여 Exploit 코드를 작성하고 대상 IoT System 속으로 침투하는 Zero-Day Attack의 침투테스트를 구현하여 취약점을 확인하였다.

2.2. Windows 10 IoT Core 취약점 분석

Windows 10 IoT Core은 Universal Windows Platform(이하, UWP) 인터페이스를 지원한다[6]. 그림 2에서는 마이크로소프트의 윈도우 플랫폼에서 UMP의 위치와 역할을 보여주고 있다.

Fig. 2 Universal Windows Platform configuration diagram

UWP은 서로 다른 다양한 장치에서도 동일한 어플리케이션을 실행시킬 수 있도록 개발된 기술이다. 즉, 모바일에서 실행 가능한 어플리케이션을 컴퓨터에서도 실행 시킬 수 있다. 이러한 UWP 기술의 핵심은 바로 UWP API의 System Call이 지원되는 것이다. PC, Moblie, IoT, Surface Hub에 해당하는 각각의 Windows 10 프로젝트 산하 운영체제 모두에서 Windows API를 호출 받아 사용할 수 있다.

본 연구에서는 IoT System에서도 Windows API 핸들링이 가능하다는 점을 활용하여 대상 IoT System의 침투를 구현하였다.

Ⅲ. 침투코드 작성 및 취약점 분석

3.1. Exploit Code 작성 및 구현

그림 3은 Reverse_Tcp Connection을 설명하는 것으로 일반적으로는 InBound에 대하여 방화벽과 같은 다양한 보안 방법이 있으나 이미 시스템 내에 코드가 내재되어 내부에서 외부로 나가는 OutBound(예를 들면 Reverse_Tcp 연결)에 대해서는 방어가 어렵다.

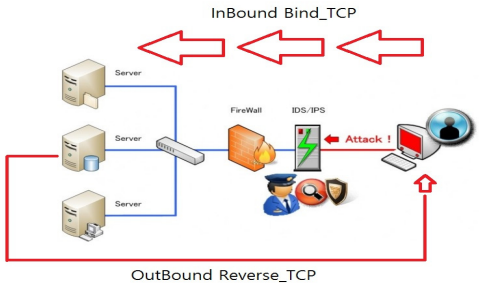

Fig. 3 Comparison of Session Establishment Differences

이를 위하여 바이러스 진단 프로그램 등이 사용되나 이러한 진단 프로그램이 잡아내지 못하면 시스템에 치명적인 보안허점을 갖게 되는 것이다.

전술한 취약점 분석을 기반으로 Exploit Code를 작성하여 대상 IoT System의 로컬 시스템 속으로 침투 할 수 있다.

Exploit Code는 Session을 수립시키기 위해 TCP/IP 인터페이스를 지원하는 Socket 클래스 Windows Sockets 2 API를 핸들링 하여 IoT System으로부터 역으로 세션을 수립시키는 Reverse_Tcp Connection을 유도한다.

대상 IoT System의 Exploit에 사용되는 코드를 분석하면, 먼저 'winsock2.h' 헤더파일이 사용된다. 이 winsock2.h 헤더파일은 Windows 소켓 헤더 파일로, 다양한 Instructions 구조체를 포함하고 있다[7]. winsock2.h 헤더를 사용하기 위해선 'ws2_32.lib' 라이

브러리 파일을 이용하고 또한, 'SOCKADDR_IN' 구조체도 사용되며 이는 Windows 소켓이 소켓을 연결할 로컬 또는 원격의 끝점 주소를 지정하는 데 사용된다. Winsock2를 사용하기 위해선, 먼저 'WS2_32.DLL'을 초기화 시켜줘야 하는데, 관련 Library를 초기화 시켜주기 위해 WSAStartup 함수가 사용된다. 또한, WSAStartup 함수 사용에 유의점으로는 프로그램의 마지막부분에 Winsock 관련 Library를 해제할 때 사용하는 'WSACleanup' 함수를 WSAStartup 함수의 호출 수와 동일한 개수로 반드시 같이 써줘야 한다는 점이다. 더불어, 트랜스포트 서비스 Provider에 바인드 된 소켓을 생성하는 WSASocket과 숫자와 점으로 이루어진 IP 문자열을 Unsigned long(데이터 형식 범위)형의 숫자 IP 주소로 변환시켜주는 inet_addr 함수, Host System에서 Network로 Short 형의 메모리 값 데이터를 보낼 때 Host Byte 순서에서 Network Byte 순서로 Byte Order를 바꿔주는 Htons 함수, 다른 어플리케이션의 소켓으로 접속을 시도하는 WSAConnect 함수, 대상 시스템으로부터의 Cmd Shell Process를 Return 받기 위해 사용하는 Create Process 함수, 이 모두를 종합하여 대상 IoT System으로부터 Cmd Shell을 획득할 수 있는 Exploit Code를 작성한다.

3.2. Cross Compile 구현

작성된 Exploit Code를 IoT System에서 동작 가능하도록 Exploit Backdoor로 컴파일하면 된다. 침투테스트 하게 되는 대상인 IoT System은 ARMv7 Thumb Architecture 이기 때문에, 작성된 Exploit Code를 일반적인 컴파일러로 컴파일 하게 되면 Exploit Backdoor가 IoT System에서 동작하지 않는다. 따라서, 컴파일러가 실행되는 플랫폼이 아닌 다른 플랫폼에서 실행 가능한 코드를 생성할 수 있는 컴파일러인 Cross Compiler를 사용하여, 작성된 Exploit Code가 ARMv7 Thumb Architecture에 포팅 되도록 컴파일 한다. 그림4는 작성된 코드에 대하여 Cross Compiler과정이다

3.3. Session Handler 설정

Cross Compiler로 컴파일 된 Exploit Backdoor로부터 역방향으로 요청되어지는 Reverse_tcp Session을 수립해주고 컨트롤하기 위한 Session Handler를 준비한다.

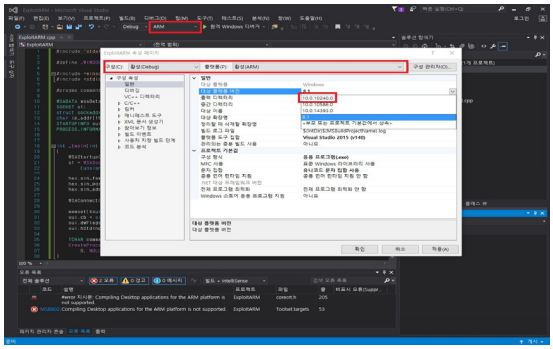

Fig. 4 Cross-compile implementation

먼저 첫 번째 방법으로, 윈도우 운영체제와 Linux 운영체제 모두에서 사용할 수 있는 Netcat Handler를 이용하는 방법이다. Netcat Handler를 Listen 모드 상태로 설정하는 -l 옵션과 세션을 수립하기 위한 포트를 지정해주는 -p 옵션을 사용하여 연결대기를 한다. 두 번째 방법으로는 다중 Session 수립을 지원하는 Linux 환경의 Multi Handler Framework를 이용하는 방법이다. Multi Handler Framework는 Resource Script File을 작성하여 Handler 옵션 값을 미리 Setting 할 수 있으며, Session 수립뿐만 아니라 Framework로서의 이후 Post Exploit 단계에 대한 다양한 침투테스트 기능을 제공한다.

3.4. IoT System Exploit

앞서 제작한 Exploit Backdoor를 통하여 IoT System의 로컬 시스템 상으로 직접적으로 침투하는 과정이다. 본 연구에서는 공격자측 환경으로 Linux 운영체제를 사용하여 Exploit을 진행하였다. 먼저, Session 수립과 컨트롤을 위한 Netcat Handler 또는 Linux 환경의 Multi Handler를 구동하여 Listening 대기시킨다.

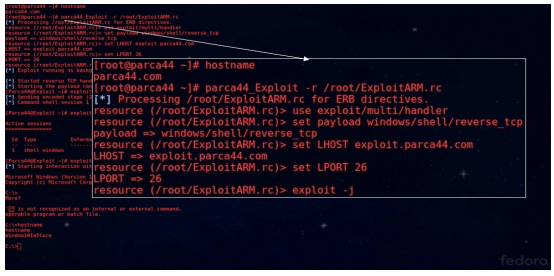

Fig. 5 Linux Multi Handler Framework Setting

그림 5는 공격자 측 환경(Multi Handler를 구동하여 Listening 대기)이 구축된 결과를 보이고 있다.

Cross Compiler로 ARMv7 Thumb Architecture에 맞게 포팅 시킨 Exploit Backdoor를 침투대상인 IoT System의 Windows 10 IoT Core에 Upload 시킨다. 그림 6은 윈도우 운영체제와 Linux 운영체제 모두에서 사용할 수 있는 Netcat Handler는 Listen 모드 상태로 설정하는 -l 옵션과 세션을 수립하기 위한 포트를 지정해주는 -p 옵션을 사용하여 연결대기 상태를 보여주고 있다.

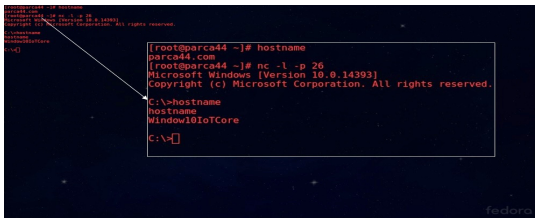

Fig. 6 Exploit implementation through Netcat Handler

Upload 된 Exploit Backdoor를 Windows 10 IoT Core에서 실행시키면, Exploit Code가 정상작동 되고, IoT System 측에서 공격자 측으로 Reverse_Tcp Connection Session 수립요청을 하게 된다. 이후, Listening 대기 중인 공격자 측 Handler는 IoT System 측에서 요청한 Session 수립을 정상적으로 공격자 측으로 수립시킨다. 위의 Exploit Code 작성단계에서 Session 수립 중간과정에 Exploit Code의 CreateProcess 함수에 의해 Cmd Shell을 실행 받아 오도록 코딩하였으므로, 이로서 공격자 측은 IoT System의 Cmd Shell을 획득하게 되고 IoT System 로컬 시스템에 대한 Zero-Day Attack Exploit 침투테스트는 성공적으로 완료된다. 그림 7은 침투된 시스템으로부터 파일이 유출된 화면을 보이고 있다.

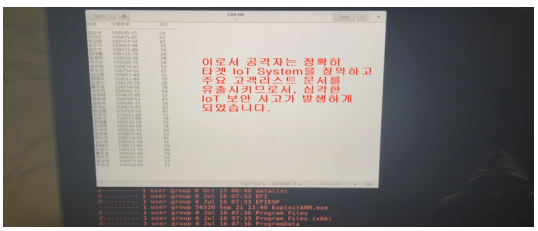

Fig. 7 Display monitor of the penetrated system

3.5. Zero-Day Attack Exploit Reporting

Zero-Day 공격에 사용된 Exploit Backdoor에 대한 악성코드 탐지여부를 조사하였다. 검증에는 전 세계의 각기 다른 백신프로그램을 사용하여 파일의 악성 여부를 분석해주는 VirusTotal 사이트를 이용하였다[8]. 그 결과, 본 연구에서 작성한 Exploit Backdoor의 악성 탐지 비율은 전 세계의 주요 백신 프로그램의 46개 중 단 하나의 백신 프로그램도 악성 탐지를 하지 못한 수치인 '악성 시그니처 탐지비율 0%'를 기록하였다. 이로서, IoT System의 ARMv7 Thumb Architecture의 Windows 10 IoT Core에 대한 취약점 존재 사실이 검증되었다. 그림 8은 개발된 코드를 바이러스 탐지여부를 확인한 결과 화면이다.

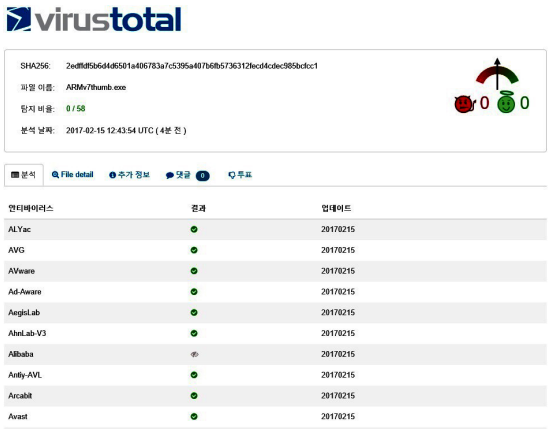

Fig. 8 'VirusTotal' malicious detection rate 0% recorded

Ⅵ. 결 론

IoT System의 운영체제인 Windows 10 IoT Core에 대한 취약점 분석을 통하여 Exploit Code를 작성하여, Zero-Day Exploit 공격으로 Cmd Shell을 획득하여, 대상 IoT System의 로컬 시스템 속으로 침투를 구현 하였다. 개발된 코드에 대한 바이러스 탐지여부를 확인한 결과 Windows 10 IoT Core의 취약점 존재를 확인할 수 있었다. IoT System 자체가 외부환경에 취약하므로 물리적 보안성을 강화하는 것이 필요할 것이다. 또한 본 연구 결과에서 확인한 취약점은 반드시 운영체제 코드패치가 이루어져야 할 것이다.

향 후 연구과제로는 Windows 10 IoT Core용 백신프로그램을 개발하거나 본 연구에서 작성된 Exploit Code에 대한 Signature를 탐지할 수 있는 보안솔루션을 개발하는 것이다.

ACKNOWLEDGMENTS

This Research was supported by the Tongmyong University Research Grants 2016(2016A011).

REFERENCES

[1] M. S. Smith, "Protecting Privacy in an IoT-Connected World," *Information Management Journal*, vol. 49, Issue 6, pp. 36-39, Nov./Dec. 2015.

[2] R. H. Weber, "Internet of things: Privacy issues revisited," *Computer Law & Security Review*, Vol. 31, Issue 5, pp. 618 - 627, Oct. 2015.

[3] Windows 10 IoT Core [Internet]. Available : https://developer.microsoft.com/ko-kr/windows/iot.

[4] The penetration testing standard [Internet]. Available : http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines.

[5] ARM architecture penetration information [Internet]. Available : https://www.defcon.org/html/links/dc-archives/dc-18-archive.html.

[6] Microsoft Windows platform [Internet]. Available : https://docs.microsoft.com/ko-kr/windows/uwp/get-started/universal-application-platform-guide.

[7] Microsoft Windsock code [Internet]. Available : https://msdn.microsoft.com/ko-kr/library/windows/desktop/ms737593(v=vs.85).aspx.

[8] Analyzes malicious contents by online antivirus engines [Internet]. Available : https://www.virustotal.com/.

김시완(Si-Wan Kim)

동명대학교 정보보호학과 학사
※관심분야 : System 보안, IoT 보안, Penetration Testing, Reverse Engineering

성기택(Ki-Taek Seong)

1998년 : 국방과학연구소(ADD) 선임연구원
2006년 : 동명대학 조교수
2006년 ~ 현재 : 동명대학교 정보보호학과 부교수
※관심분야 : 네트워크 보안, 임베디드시스템 보안