IoT 펌웨어 취약점 데이터베이스 구축 방안 연구

이경석[O], 조호묵[*]
[O]KAIST 사이버보안연구센터,
[*]KAIST 사이버보안연구센터
e-mail: {kslee[O], chmook79[*]}@kaist.ac.kr

Construction of Database for IoT Firmware Exploit

Kyeong Seok Lee[O], Ho Mook Cho[*]
[O]KAIST Cyber Security Research Center,
[*]KAIST Cyber Security Research Center

● 요 약 ●

본 논문에서는 IoT 취약점 탐지 시스템과 취약점 데이터베이스 구축 방안을 제안한다. 동적 웹페이지 제어기술 기반의 크롤링 기법으로 펌웨어를 수집한 후, 패킹된 펌웨어 파일을 Binwalk, FMK를 활용하여 추출하고 Qemu 에뮬레이팅 기반의 실제 서비스를 실행시키는 시스템을 구현하여 펌웨어 취약점을 탐지할 수 있는 환경을 구축한다. 구축된 시스템을 통하여 수집, 추출, 에뮬레이팅 과정에서 생성된 데이터와 연계되어 탐지된 취약점 정보를 저장할 수 있는 데이터베이스를 제안한다. 제안된 시스템과 데이터베이스를 통하여 IoT 기기 펌웨어의 취약점을 탐지하고 예방을 할 수 있을 것이라 기대한다.

키워드: 사물인터넷(Internet of Things), 펌웨어(Firmware), 에뮬레이팅(Emulating)

I. Introduction

IoT(Internet of Things) 기술의 발전에 따라 수많은 IoT 제품이 가정이나 회사 등 일상생활에서 사용되고 있다. 가트너에 따르면 2009년 9억 대에서 2020년에는 약 260억대에 이를 것으로 전망했다. 하지만 IoT 기기 대부분은 소형화 기기 임베디드 소프트웨어 특성에 따라 보호 기능이 부족하여 위험에 노출되어 있다 보고하였다[1]. 2016년 말, 아마존, 페이팔, 넷플릭스 등 많은 온라인 서비스에 분산 서비스 공격(DDoS) 공격을 가한 미라이 악성코드는 수천 대의 IoT 장비들의 보안취약점을 악용하여 DNS 호스팅 업체에 과도한 트래픽을 유발시켰고[2], 이후에도 에이드라, 가프지트, 와이패츠 등 미라이 기반 악성코드가 지속적으로 발견되었다. 또한, IBM에 따르면 2019년에 IoT 기기의 취약점은 5년 전과 비교해 5배 증가하였다[3]. 기하급수적으로 증가하는 IoT 기기는 보안에 취약한 상태이며, 취약점 탐지를 통한 사고 예방이 필요로 하다. IoT 기기와 같이 소형화 제품에 사용되는 임베디드 소프트웨어는 특정 하드웨어를 구동하기 위해 전용화되어 내장되는 소프트웨어이다[4]. 임베디드 소프트웨어의 보안취약점을 분석하기 위하여 일반 소프트웨어 분석과 같이 정적/동적분석 방법을 사용한다. 정적분석은 소스코드를 스캔하여 분석하는데 오픈소스는 가능하지만, 상용화된 제품은 소스코드를 공개하지 않아 제한점이 있다. 동적분석은 소프트웨어를 동작시키고 테스트를 통해 분석한다. 기기에 특화되어있는 임베디드 소프트웨어는 모든 기기를 사용하여 직접 테스트하기 어려워 가상 환경을 구축하여 분석한다. 본 논문에서는 제조사에서 배포되는 IoT 기기의 임베디드 리눅스 기반 펌웨어를 수집하고 이를 분석할 수 있도록 파일 언패킹, 가상화된 환경으로 에뮬레이팅 과정을 거쳐 취약점을 탐지할 수 있는 시스템과 이를 기반으로 수집된 IoT 취약점 데이터베이스 구축 방안을 제시하고자 한다.

II. Related Works

1. 동적 웹페이지 제어

일반적으로 펌웨어는 제조사의 웹사이트나 ftp를 통하여 수집한다. 펌웨어를 수집하기 위하여 크롤링 기법을 활용하는데 크롤링 기법은 크게 두가지로 나눈다. 정적 크롤링은 Request 방식의 통신을 이용하여 웹서버의 응답값을 토대로 구문 분석하여 원하는 데이터를 추출한다. 하지만 다수의 웹사이트는 Javascript, Flash 등 동적 웹페이지로 구현되어 있어 정적 크롤링과 같이 소스코드를 직접 실행시키지 않으면 소스코드만으로는 원하는 데이터를 추출하기 어렵다. 동적 크롤링은 실제 브라우저를 이용하여 Javascript와 같은 동적 웹페이지의 소스코드를 실행시켜 원하는 데이터를 가져올 수 있으며, 필요에

따라 클릭 이벤트, 텍스트 입력 등 사용자 행위를 자동으로 수행할 수 있도록 커스터마이징이 가능하다[3]. 이러한 동적 크롤링 구현을 위해 사용되는 Selenium은 웹 어플리케이션 테스트를 위한 프레임워크로서 Chrome, IE, Firefox 등 웹 브라우저를 자동 제어하기 위한 오픈소스이다[5].

2. 펌웨어 파일 시스템 추출

임베디드 리눅스 기반의 기기에서 사용되는 펌웨어는 LZMA, TRX, SquashFS, CramFS 등 다양한 방법으로 압축되어 있어, 커널, 파일 시스템, 실행 파일 등 다양한 콘텐츠를 구조화된 형태로 추출하는 기술이 필요하다. Binwalk[6]는 펌웨어 이미지 파일을 분석할 수 있는 오픈소스로서, 커널, 파일 시스템, 시그니처 정보 등을 추출할 수 있을 뿐 아니라 펌웨어 내 압축된 데이터를 재귀적으로 추출할 수 있다. Binwalk

기반의 펌웨어 이미지 분석 도구인 FMK(FirmwareMod Kit)[7]은 Binwalk에 비하여 더 많은 펌웨어 유형을 지원하며 자동 파일 시스템 추출이 가능하다. 하지만, 위 도구들은 일부 암호화된 펌웨어에 대한 파일 시스템을 추출할 수 없는 경우가 발생한다는 제약이 있다.

3. 펌웨어 에뮬레이팅

IoT 기기의 취약점을 분석하기에 가장 좋은 방법은 실제 기기상의 실행된 어플리케이션이나 서비스를 분석하는 것이다. 하지만 무수히 많은 IoT 기기를 분석하는 것은 불가능하고, 실제 사용되는 IoT 기기의 가용성 문제로 인해 직접 분석은 어려움이 있다. 따라서 다수의 연구에서는 가상화 에뮬레이팅 환경을 구성하여 각 기기에 서비스되는 소프트웨어를 분석하는 방법을 사용한다. 이러한 에뮬레이팅 기술 중의 하나인 Qemu는 x86, MIPS, ARM 등 다양한 아키텍처 기반의 임베디드 리눅스를 실행할 수 있는 하드웨어 가상화 CPU 에뮬레이터이다[8]. MIPS, ARM 등 펌웨어 아키텍처에 맞춰 커널과 이미지를 부팅시킨 후 펌웨어 루트 파일 시스템 업로드 및 권한 부여, 서비스 실행 등을 수행하여 펌웨어가 정상적으로 동작할 수 있도록 가상화 에뮬레이팅한다.

III. The Proposed Scheme

본 논문에서는 IoT 취약점 탐지를 위하여 펌웨어 수집 및 분석 시스템과 분석과정에서 발생된 데이터와 취약점 정보를 저장할 수 있는 취약점 데이터베이스를 제안한다. 이 시스템은 펌웨어 수집, 펌웨어 이미지 분석, 펌웨어 에뮬레이팅 총 3단계 시스템으로 구축한다. 각 단계별로 수집되는 펌웨어 분석 정보는 취약점 데이터베이스에 저장한다. 시스템 구성도는 Fig. 1.과 같다.

1. 펌웨어 수집

본 논문에서는 상용화된 제품에서 실제 사용되는 펌웨어를 수집하기 위하여 국내외 IoT 기기 제조사 홈페이지를 조사하였고, 각 홈페이지별로 IoT 기기 제품 탐색 모듈 및 제품의 다운로드 링크 수집, 펌웨어 다운로드를 수행하는 크롤러를 구현하였다. 로그인이나 특정 제품군 검색을 위한 클릭이벤트, Javascript 제어 등 사용자의 개입 행위가 필요한 웹페이지를 위하여 Selenium 기반의 동적 크롤러를 구현하였다. 최종적으로 공유기, IP카메라, TV 등 3종, 약 5,500개의 펌웨어를 수집하였고, 수집결과는 Table 1과 같다. 펌웨어 수집 크롤러는 각 펌웨어의 파일명, 버전, 다운로드 URL, 게시날짜, Hash 등 Metadata를 취약점 데이터베이스에 저장하여 기초자료로써 활용할 수 있도록 하였다. 또한, 다운로드 링크와 파일 Hash 등을 조합하여 Transaction Hash를 생성하여 펌웨어 수집 때 중복 수집을 예방하였다.

2. 펌웨어 이미지 분석

수집된 펌웨어를 에뮬레이팅할 수 있도록 파일 추출 및 아키텍처 분석 도구를 구현하였다. BIN, ZIP, RUI 파일 등의 펌웨어가 수집시 자동으로 분석될 수 있도록 시스템을 구현하였다. 파일 다운로드가 완료되는 시점으로부터 해당 스토리지 경로에 있는 파일을 불러와 아키텍처를 분석한 후, 패킹된 펌웨어 파일을 FMK와 Binwalk를 활용하여 파일 시스템으로 추출하였다. 추출과정에서 httpd, rc 등 중요 서비스 위치를 파악하여 경로 정보를 데이터베이스에 저장하였다. 각 펌웨어는 Hash를 기준으로 데이터를 저장하여 차후 분석에

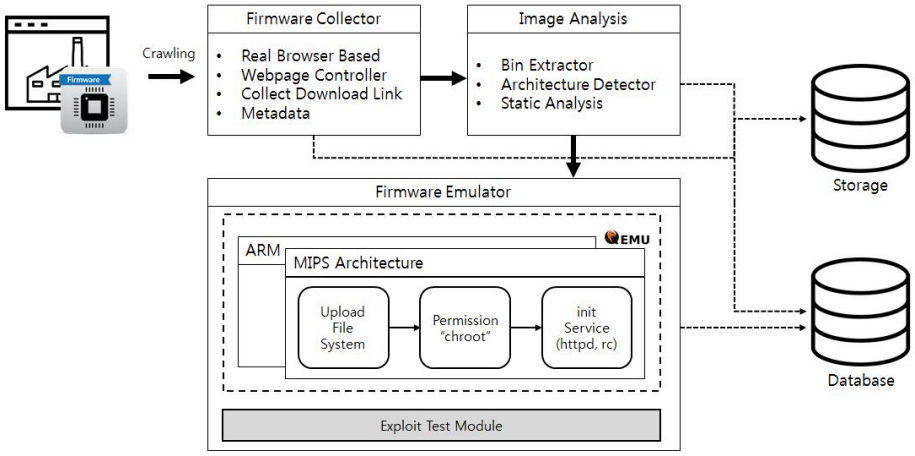

Fig. 1. System Architecture

Table 1. Result of Firmware Collection

	제조사	제품수	파일수
공유기	Iptime	185	3872
	Asus	84	460
	Tplink	17	17
	Netgear	9	120
	Humax	10	15
	Eznet	14	14
	Dlink	36	205
	Netis	43	369
IP카메라	Hanhwa	230	232
	Webgate	24	24
	Idis	62	42
TV	LG	44	99
	Samsung	75	4
	Lucoms	8	9
총		841	5482

용이하도록 하였다. 수집된 펌웨어를 분석한 결과, 아키텍처는 MIPS,ARM 중 MIPS의 비중이 높았으며, SSL 등 인코딩되어 파일 추출을 수행할 수 없는 경우에는 분석대상에서 제외하였다.

3. 펌웨어 에뮬레이팅

추출된 분석대상 펌웨어는 Qemu를 통하여 각각 MIPS, ARM 아키텍처에 맞게 에뮬레이팅하였다. 펌웨어 내 루트 파일 시스템을 가상환경에 업로드한 후, 서비스를 정상적으로 실행시키기 위하여 해당 경로에 루트권한을 부여하였다. 펌웨어 이미지 추출과정에서 분석된 httpd, rc 등 펌웨어 서비스 위치를 참고하여 펌웨어 서비스를 시작하는 프로세스를 실행시켰다. 공유기 제어 페이지와 같은 웹서비스에 접근하여 펌웨어 서비스 실행 여부를 확인하여 취약점 테스트를 진행하기 위한 환경을 구성하였다.

4. IoT 펌웨어 취약점 데이터베이스

본 논문에서는 펌웨어 수집/이미지분석/에뮬레이팅 등 3단계의 펌웨어 에뮬레이팅을 통한 취약점 탐지 환경을 구현하였고 이 과정에서 생성된 펌웨어 관련 데이터 기반 펌웨어 취약점 데이터베이스를 구축하였다. IoT 기기 제조사 정보 테이블, 수집된 펌웨어 Metadata 테이블, 펌웨어 정적 분석 테이블, 에뮬레이팅 테이블, 모의해킹 테이블로 구성하였다. 각 테이블에 따른 콘텐츠 정보는 Table 2.와 같다. 현재는 에뮬레이팅 과정까지의 펌웨어 정보를 데이터베이스에 저장하였고 향후에는 Nmap, Routersplit과 같은 모의해킹 도구들을 활용하여 취약점 테스트를 진행하고 펌웨어 취약점 정보를 추가하도록 하고자 한다.

Table 2. Contents of IoT Firmware Exploit Database

컬럼명	내용	비고
제조사 Table		
Name	제조사명	
Category	기기분류	Wifi/IPcam/...
URL	제조사 링크	펌웨어 페이지
crwDate	크롤링 날짜	
downDate	다운로드 날짜	
펌웨어 기본 정보 Table		
productName	제품명	
fileName	파일명	
Hash	파일 해쉬	Sha256
tranHash	트랜젝션 해쉬	제품명/해쉬/날짜
Version	버전	
downURL	다운로드 링크	
uploadDate	게시 날짜	
downDate	수집 날짜	
filePath	스토리지 경로	
펌웨어 정적 분석 Table		
Hash	파일 해쉬	
Architecture	아키텍쳐 정보	MIPS/ARM/...
extractFlag	파일 시스템 추출	성공여부
extractPath	파일 시스템 경로	스토리지 경로
webPath	웹서비스 경로	
rcPath	rc 파일 경로	
busyboxPath	busybox 경로	
에뮬레이팅 정보 Table		
Hash	파일 해쉬	
Architecture	아키텍쳐 정보	MIPS/ARM/...
emulFlag	에뮬레이팅	성공여부
webFlag	서비스 실행	성공여부
rcFlag	서비스 실행	성공여부
모의해킹 Table		
nmapFlag	네트워크 스캔	실행여부
portNum	열린 포트 정보	
serviceName	열린 서비스	
routerFlag	routersploit	실행여부
routerResult	탐색 결과	

IV. Conclusions

기하급수적으로 증가하는 IoT 기기의 보안 취약성을 분석하기 위하여 IoT 기기 펌웨어의 취약점을 탐지하기 위한 펌웨어 수집 및 분석 시스템과 IoT 펌웨어 취약점 데이터베이스를 제안하였다. 사용자 개입이 필요한 웹페이지를 제어하기 위한 동적 크롤링 기반 펌웨어 수집 시스템을 구현하였고, 수집된 펌웨어를 Binwalk, FMK를 활용하여 파일 시스템 추출 및 아키텍처 탐지를 수행하였다. Qemu 기반 가상환경 에뮬레이팅을 통하여 펌웨어 동적 분석 환경을 구축하였다. 이 과정에서 발생하는 펌웨어 수집 및 분석 정보를 저장하는 데이터베이스를 구축하였다. 향후에는 Nmap, Routersplit과 같은 모의해킹 도구를 활용하여 펌웨어 취약점 탐지 정보를 확장하고자 한다. 이를 기반으로 실제 운용중인 IoT 기기의 위험성을 선제적으로 탐지하여 예방할 수 있을 것이라 기대한다.

REFERENCES

[1] Mark H, “Leading the IoT”, https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

[2] B.Herzberg, D.Bekerman, and I.Zeifman, “Breaking Down Mirai: An IoT DDoS Botnet Analysis,” https://www.incapsula.com/blog/malware-analysismirai-ddos-botnet.html, Nov. 2017.

[3] Cho, H. M, Lee, K. S. and Lee, J. H., “High-speed Dynamic Crawler Implementation with Resource Occupancy Optimization,” Korea Computer Conference 2019, pp. 1138-1140, 2019.

[4] Heo,J. Park,T.H., and Kim.J.M. "Research on Embedded/Intelligent Software Market Value Survey", https://spri.kr/posts/view/22924?code=research. 2020.

[5] Selenium, *https://www.selenium.dev/*

[6] Lee, S. and Cho, E. S., “Toward Firmware-Type Analysis Using machine Learning Techniques,” IEEE Computer Software and Applications Conference, Vol. 1, pp. 977-978, 2018.

[7] Firmware Modification Kit, *https://code.google.com/archive/p/firmware-mod-kit/*

[8] Chen,D. Woo,M., Brumley,D., and Egele.M. "Towards Automated Dynamic Analysis for Linux-based Embedded Firmware." NDSS. Vol. 16. 2016.