Journal of KIIT. Vol. 21, No. 11, pp. 163-169, Nov. 30, 2023. pISSN 1598-8619. eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2023.21.11.163

버퍼 오버플로 취약점 및 변조 펌웨어를 통한 유무선 공유기 사용자 데이터 탈취

남윤수*, 최선오**

Stealing Wireless Router User Data Through Buffer Overflow Vulnerability and Tampering Firmware

Younsu Nam*, Sunoh Choi**

이 논문은 2023년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임
(No. RS-2023-00237159)

요 약

2010년대 이후 스마트 디바이스가 보급됨에 따라 무선 네트워크가 널리 설치되고, 특히 비용 절감 및 편의를 위해 유무선 공유기가 사용되고 있다. 하지만 유무선 공유기가 많이 보급됨에 따라 공격자가 수많은 공격 수단으로 정상적인 동작을 방해 후 데이터 탈취를 시도하는데, 특히 비정상 동작을 수행하여 임의 코드 수행이 가능한 버퍼 오버플로 공격이 활성화되고 있다. 본 논문에서는 버퍼 오버플로 취약점을 통해 사용자 동의 없이 변조된 펌웨어로 부팅 후 사용자 데이터를 탈취하는 방법을 시도한다. 시도하는 방법은 어려운 방법을 요구하지 않고, 사용자 동의 없이 펌웨어 변경을 통한 데이터 탈취가 가능하다. 이러한 방법을 통해 공유기를 사용하는 사용자 및 제조사가 보안에 신경 써야 함을 알 수 있었다.

Abstract

As smart devices have become widespread since the 2010s, wireless networks have been widely installed, and wired and wireless routers are especially used for cost reduction and convenience. However, as it becomes more widespread, attackers use numerous attack methods to disrupt normal operations and then attempt to steal data. In particular, buffer overflow attacks, which allow arbitrary code execution by performing abnormal operations, are becoming more active. In this paper, we attempt to steal user data after booting with firmware that has been modified without user consent through a buffer overflow vulnerability. The method being attempted does not require difficult methods and allows data theft through firmware change without user consent. Through this method, we learned that users and manufacturers of routers need to pay attention to security.

Keywords

wireless router, heap-buffer overflow attack, vulnerability analyze, OpenWRT, tcpdump, net-usb

* 전북대학교 소프트웨어공학과 학사과정
- ORCID: https://orcid.org/0009-0006-0753-8235
** 전북대학교 소프트웨어공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0002-0654-7109

· Received: Jul. 27, 2023, Revised: Sep. 04, 2023, Accepted: Sep. 07, 2023
· Corresponding Author: Sunoh Choi
Dept. of Software Engineering, Jeonbuk National University Jeonju, Korea
Tel.: +82-63-270-4784, Email: suno7@jbnu.ac.kr

Ⅰ. 서 론

2010년대 이후 스마트 디바이스가 보급됨에 따라 무선 네트워크(Wireless network)가 널리 보급되고 있다. 사람들이 가장 많이 활용하는 스마트폰부터 몸에 착용하는 웨어러블 디바이스, 사물을 인터넷으로 제어할 수 있는 사물 인터넷 등 각 계층별로 인터넷을 통한 상호작용을 할 수 있다. 특히 비용의 절감과 관리의 용이성을 위해 기존 무선 네트워크 기기와 유선 네트워크 기기를 통합한 유무선 공유기가 가정 및 소규모 사무실을 대상으로 널리 보급되고 있다[1].

하지만 널리 보급되고 있는 만큼 많은 보안 취약점과 침해 사고가 일어나고 있다. 공유기에 대한 보안 공격이란 악의를 가진 공격자가 공유기의 정상적인 동작을 방해하거나, 송수신 되는 데이터를 엿듣거나 변조하여 피해를 입히거나, 내장된 펌웨어의 취약점을 통해 의도되지 않은 동작을 수행할 수 있다. 근래에는 보안 공격을 손쉽게 실행할 수 있는 오픈소스 소프트웨어나 프로젝트가 공개되어 있어, 만약 취약점이 공개되어 있다면 크래킹에 대한 지식이 없어도 사용법에 따라 실행하면 피해를 줄 수 있는 보안 공격을 실행할 수 있다[1][2].

본 논문에서는 위에서 언급한 방안을 통해 특정 공유기 제조사 펌웨어를 분석하여 비정상 동작을 수행할 수 있는 버퍼 오버플로 취약점을 찾아 시도하며 공격을 통한 관리자 권한을 획득할 것이다. 시도하는 방법에서 공격자는 취약점을 통해 관리자 권한을 획득하여 악의적 공격을 할 수 있는데, 특히 본 연구진은 취약점이 성공하면 펌웨어를 자동으로 변조된 펌웨어로 변경하는 방안도 시도하였다. 이러한 취약점은 어려운 방법을 요구하지 않아 위험성이 더 크고, 변조된 펌웨어를 사용자가 관리자가 인지하기가 어렵다[2][3].

이를 통해 공유기를 사용하는 사용자가 보안에 더 많은 관심을 가지며 제품을 선택해야 하고, 각 제조사는 취약점이 존재하면 이를 대비하고, 펌웨어 보안이 더욱 강화되어야 함을 보였다.

Ⅱ. 관련 연구

2.1 유무선 공유기

유무선 공유기(이하 공유기)란 기존 무선 네트워크 기기 및 유선 네트워크 기기를 통합하여 가정이나 소규모 사무실 등에서 사용하는 소용량의 라우터를 의미한다[1]. NAT(Network Address Translation) 기능을 통해서 사설 IP를 할당하여 통신을 할 수 있는 기능이 포함되어 있으며, 최근 출시되는 제품들의 경우에는 USB, 클라우드 등을 연결하여 단순히 홈 네트워크 구성뿐만이 아닌 장치 허브의 역할을 수행하기도 한다.

2.2 버퍼 오버플로우 공격

버퍼 오버플로우 공격(Buffer overflow attack) 이란 데이터의 길이에 대한 불명확한 정의를 악용하여, 공격자가 임의의 코드를 덮어쓰는 것을 의미한다[4].

프로그램은 수많은 함수로 구성되어 있다. 함수가 호출되면 지역 변수 및 복귀 주소가 스택에 저장된다. 이 복귀주소는 함수가 실행되고 난 다음의 행동에 대해 지정해 둔 주소인데, 만약 함수의 지역 변수 혹은 다른 변수에 의해 복귀주소가 침범 당하면 전혀 다른 명령을 실행시킬 수 있다[5].

2.3 기존 공격방법과의 차이점

기존의 공유기 공격 방법은 크게 두 가지로 나눌 수 있다. 공유기의 웹 취약점을 이용한 공격과 DoS를 통한 사용자가 사용할 수 없도록 만드는 공격으로 나눌 수 있다.

웹 취약점을 이용한 공격은 XSS(Cross Site Script)나 CSRF(Cross Site Request Forgery)등을 이용하여 공유기의 DNS를 조작하여 파밍(Pharming) 사이트로 유인하는 유형의 공격 등이 있다[1].

DoS를 통한 공격은 공격자가 패킷 변조등을 통한 서비스 거부 혹은 과도한 트래픽 흐름을 유발하여 사용자가 접속을 할 수 없도록 하거나 이용할 수 없을 정도의 네트워크 효율을 저하시킨다[4].

정적 분석과정을 통한 버퍼 오버플로우 공격은 공유기가 기본적으로 제공하는 관리 웹 페이지의

취약점 혹은 실제 기기와의 통신을 진행하는 백엔드(Back-end) 바이너리의 취약점을 응용하거나[5], 오버플로 취약성을 검출할 수 있는 스크립트 및 프로그램을 사용하여 바이너리 디버깅을 할 수 있다[6].

본 논문에서는 위와 같은 정적 분석 및 디버깅을 통한 취약점 분석뿐만이 아닌, 공유기 펌웨어에서 사용하는 라이브러리 혹은 시스템의 취약점을 분석한다.

이는 다양한 기능을 제공하는 공유기일수록 다양한 라이브러리를 사용하기 때문에 취약점 분석이 수월하며, 특히 외부 라이브러리 취약점일 경우 외부 제조사에서 수정하기 전까지는 지속적으로 응용할 수 있는 취약점이기 때문에 대응하기에 어렵다.

특히 취약점을 통한 펌웨어 변조 과정에서 사용자의 동의 없이 데이터 탈취가 가능한 펌웨어로 변조 할 수 있으며, 30초 정도의 짧은 시간 이후에는 공격자가 데이터 탈취를 손쉽게 할 수 있다.

III. 설 계

3.1 실험 시스템 구성

일반적으로 공격자가 공격을 수행하면 공격 대상에 대한 시스템 분석을 수행한다. 보통 무선 취약점의 경우 같은 프로토콜을 사용하여 많은 무선 AP를 대상으로 수행할 수 있지만 본 연구진이 수행하려 하는 방법에서는 기기 내부의 취약점을 이용하기 때문에, 특정 대상에 대한 정밀한 방법을 기반으로 접근해야 한다.

본 논문에서 구성하고 실험한 시스템에서는 TP-Link사의 제품군 중 하나인 C7을 기반으로 하였다. 선정한 이유는 가장 많이 사용되는 공유기이며, 개발 자료가 풍부하게 공개되어 있기 때문이다.

‘뉴욕타임스’에서는 여러개의 공유기 제조사를 상대로 수백 시간의 테스트를 거쳐, 가격대비 성능이 가장 뛰어난 공유기로 C7을 선정하였다. 또한 그림 1과 같이 인터넷 판매 쇼핑몰 ‘Amazon’ 등에서 “다른 사람에게 추천할 수 있는 공유기”의 상위 등수에 오르기도 한 제품이기도 하다.

그리고 오픈소스 라우터 OS 중 하나인 “OpenWRT” 에서 가장 많은 사용자가 사용하고, 관련 자료 및 연구가 풍부하게 진행되었기 때문에 이 공유기를 선정하였다.

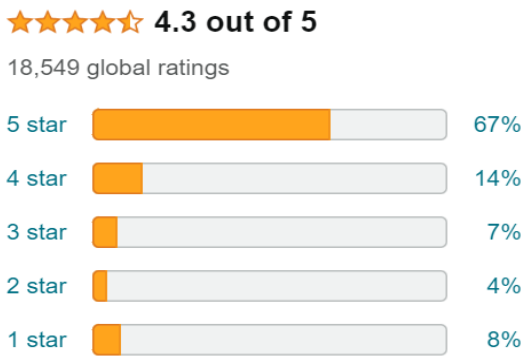

그림 1. 선정한 시스템의 아마존 상품 평가 및 추천 점수
Fig. 1. Amazon product evaluation and recommendation scores for the selected system

최종 시스템 구성은 그림 2와 같이 공격자가 공격할 대상과 같은 네트워크상에 존재하며, 사용자들 또한 같은 네트워크상에 존재한다.

3.2 취약점 분석

공격에 필요한 취약점을 찾기 위해선 펌웨어에 대한 정적 분석과 동적 분석 과정이 필요하다.

정적 분석은 펌웨어 이미지를 직접 분석하는 것이다. 분석을 위한 펌웨어 획득을 위해 JTAG(Joint Test Action Group) 등의 직접 접근 방법으로 이미지 사본을 제작하거나, 제조사 홈페이지 등에서 제공하는 펌웨어 이미지를 다운로드 받을 수 있다[1].

동적 분석은 동작하고 있는 시스템을 실시간으로 디버깅을 하는 것이다. 분석을 위해 UART(Universal Asynchronous Receiver/Transmitter) 등을 사용하여 시스템에 직접적으로 접근하여 동작에 따른 결과를 디버깅할 수 있다[1].

먼저 정적 분석 이미지를 통해 사용하는 모듈 및 프로그램을 분석하고, netstat, top 등 프로세스 및 열린 통신 포트를 조회하며 동적 분석을 진행하였다. 분석 과정에서 프로세스 이름이 없는 20005번의 TCP 포트가 존재하는데, 이 포트는 리눅스 기반 임베디드 시스템에서 연결된 프린터, 외장 하드 디스크, 플래시 드라이브 등을 사용자가 웹상에서 접근할 수 있도록 하는 ‘Net-USB’ 커널 모듈의 포트이다.

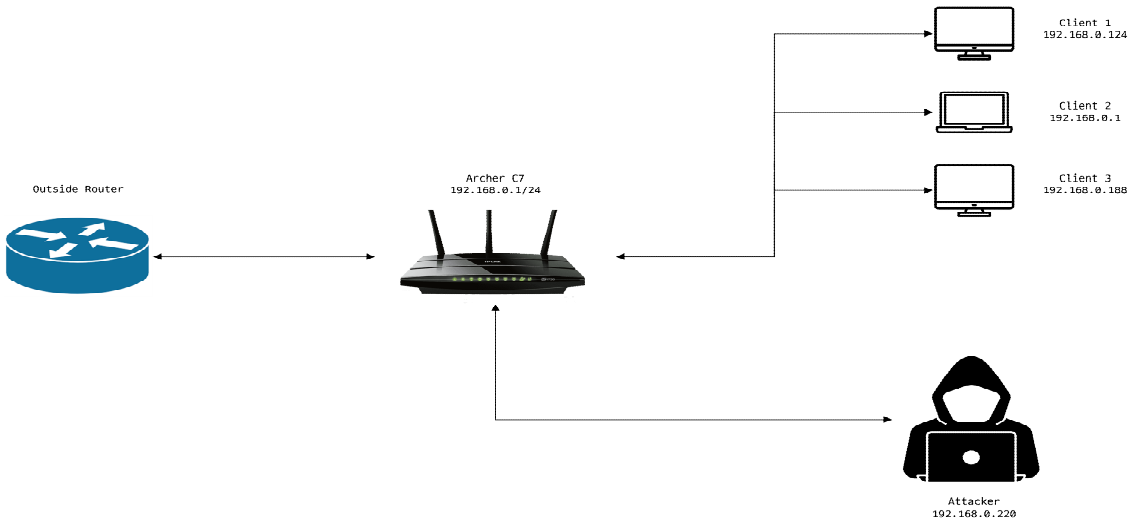

그림 2. 구성 및 실험 시스템 네트워크 및 사용자 구성도
Fig. 2. Configuration and experiment system network and user configuration diagram

지난 2021년 사이버 보안 업체 'SentinelOne'은 Net-USB 커널 모듈에 버퍼 오버플로 공격을 통한 원격 코드 실행 취약점이 존재한다는 사실을 CVE-2021-45608로 등록하였다[2]. 버퍼 오버플로 공격이란 프로그램이 실행될 때 입력받는 값이 버퍼에 저장될 수 있는 값을 넘어서 버퍼 이후의 공간을 침범하는 현상을 의미한다[2]-[4].

앞서 설명한 버퍼 오버플로 공격을 수행하기 위해 살펴본 결과, 목표 시스템 내부에서 "SoftwareBus_dispatchNormalEPMsgOut" 라는 함수가 존재한다. 이 함수는 메시지 버퍼를 통해 버퍼를 읽는 부분이 존재한다. 이때 전체 버퍼를 읽지 않고도 함수를 반환할 수 있게 되어있는데 이러한 부분은 작은 버퍼를 강제로 할당하고 공격자가 원하는 만큼의 데이터를 오버플로 할 수 있는 취약점이다[3]. 이러한 취약점을 작동시킬 수 있는 pseudo-code는 그림 3과 같다.

이 코드는 윗 절에서 설명 한 특정 주소 및 포트에 대한 통신을 개방한 후, 오버플로우를 시킬 수 있는 데이터를 전송한다[5]. 이때 오버플로가 발생하면 시스템 커널의 소켓 스택에 인접한 구조체에 영향이 가, 오버플로가 일어난 공간에 간단한 쉘을 실행할 수 있는 데이터를 전송한다[6].

데이터가 안정적으로 실행되면 그림 4와 같은 쉘 스크립트를 공격자에게서 다운로드 받아, 변조된 펌웨어를 업그레이드하는 명령을 실행한다.

```python
class TrasherThread(threading.Thread):
    def __init__(self, target):
        threading.Thread.__init__(self, name = 'trasher')
        self.target = target
        self.port = netusb_port
        self.overflow_event = threading.Event()
        self.overflow_done_event = threading.Event()
        self.overflow_content = None

    def overflow_and_wait(self, overflow_content):
        self.overflow_content = overflow_content
        self.overflow_event.set()
        self.overflow_done_event.wait()

    def run(self):
        s, aes_ctx = create_connection(self.target, self.port, self.name)
        s.send(le8(0xff))
        s.send(le8(0x51))
        s.send(le32(0xff_ff_ff_ff))
        s.send(le32(0x11_22_33_44))
        self.overflow_event.wait()
        s.send(le32(0x55_66_77_88))
        p = le8(0xaa) * (128 - 0x10)

        # Fill up overflow
        p += self.overflow_content
        s.send(p)
        self.overflow_done_event.set()
        s.close()

def main():
    args = "192.168.0.1"
    trasher = TrasherThread(args.target)
    trasher.start()
    overflow_payload = b'A' * 0x10_000
    trasher.overflow_and_wait(overflow_payload)
    print('Done')
```

그림 3. 실험 시스템의 취약 모듈 데이터 오버플로가 가능한 pseudo-code
Fig. 3. Pseudo-code capable of overflowing data from vulnerable modules in the experimental system

```
#!/bin/sh
cd /tmp && wget https://{attacker-ip}:3000/archerc7-firm.bin
sysupgrade -F archerc7-firm.bin firmware
reboot -f
```

그림 4. 변조 펌웨어 다운로드 후 설치하는 악성 쉘 스크립트 예시
Fig. 4. Example of malicious shell script that downloads and installs modified firmware

3.3 변조 펌웨어 제작

앞서 설명한 공격 과정을 통해 공격자가 원하는 공격을 할 수 있지만, 완전한 제어를 얻기 위해서는 더욱더 많은 취약점 분석과 다른 연관 프로그램들의 분석이 필요하다. 하지만 사용자의 데이터를 탈취하는 라이브러리를 공격 과정에서 포함하기는 어렵다.

그래서 “OpenWRT”를 통한 펌웨어 제작 및 네트워크 상황 파악 및 분석할 수 있는 라이브러리를 포함한 변조 펌웨어을 업로드하는 방법을 고안하였다.

이러한 방법은 제한된 환경 내에서 특정 공격을 수행하면서도 사용자나 관리자는 인지하기가 어렵다는 것인데, 예를 들어 단순히 공유기를 네트워크 공유 목적으로 사용하는 경우 관리 페이지에 접속을 자주 하지 않기 때문에, 변조 사실을 알아채기는 어렵다[8][9].

이를 적용하기 위해 그림 5와 같이 오픈소스 라우터 OS 중 하나인 ‘OpenWRT’를 활용하여 시스템 구성에 맞는 펌웨어 선택 및 사용자 패킷 데이터를 분석할 수 있는 TCPDump를 포함하여 빌드하였다.

최종적으로 빌드 된 펌웨어를 통해 아래와 같은 공격 실험을 진행하였다.

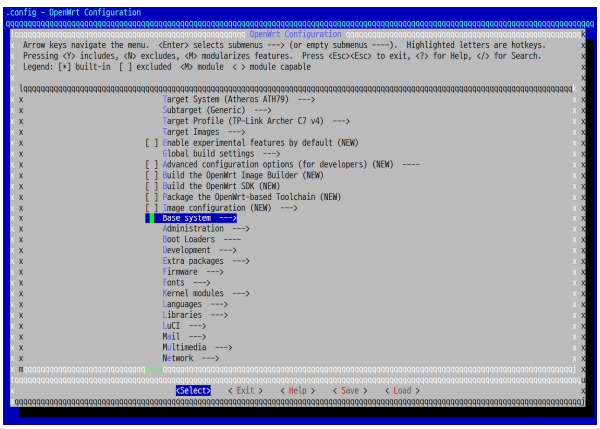

그림 5. OpenWRT 타겟 시스템 변조 이미지 빌드 설정 화면

Fig. 5. OpenWRT target system modulated image build settings screen

3.4 공격 시나리오 및 실험

실제 공격을 위한 실험을 위해 위 절에서 설명한 시스템 구성으로 진행한다.

먼저 공격자와 사용자는 같은 네트워크상에서 존재한다. 이때 공격자가 버퍼 오버플로 공격을 수행할 경우 그림 6과 같이 UART를 이용한 디버그 화면에서는 실시간으로 메모리 페이징이 일어나며 의도하지 않은 메모리 영역을 사용한다[10]. 이후 악성 스크립트를 통해 변조된 펌웨어를 다운받아 펌웨어 업그레이드를 강제로 진행한다.

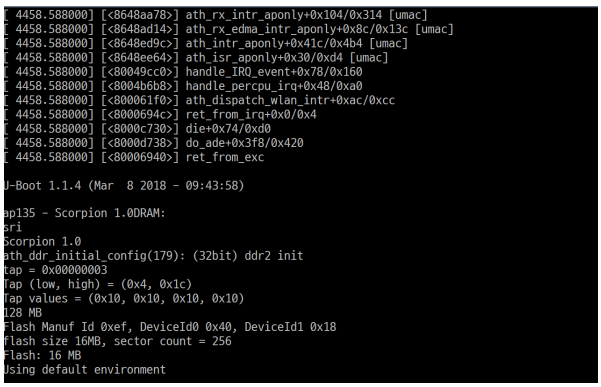

그림 6. 버퍼 오버플로 공격 성공 이후 강제 재부팅 로그 디버그 화면

Fig. 6. Forced reboot log debug screen after successful buffer overflow attack

업그레이드가 완료되면 재부팅이 되며 변조된 펌자들의 데이터를 확인 혹은 변조를 진행할 수 있다.

만약 사용자 또는 관리자가 관리 페이지에 접근하면 기존에는 그림 7과 같이 제조사 펌웨어 공유기 관리 페이지를 볼 수 있었으나, 변조된 이후에는 그림 8과 같이 변조 펌웨어로 사용한 ‘OpenWRT’의 관리 페이지로 변경된 것을 볼 수 있다.

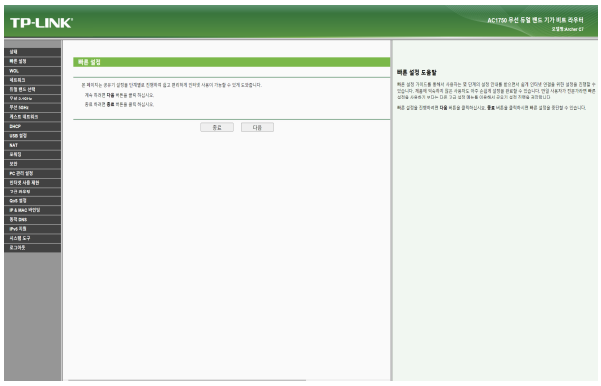

그림 7. 공유기 접속 시 기존 제조사 펌웨어 관리 페이지

Fig. 7. Manufacturer firmware management page when connected to a router

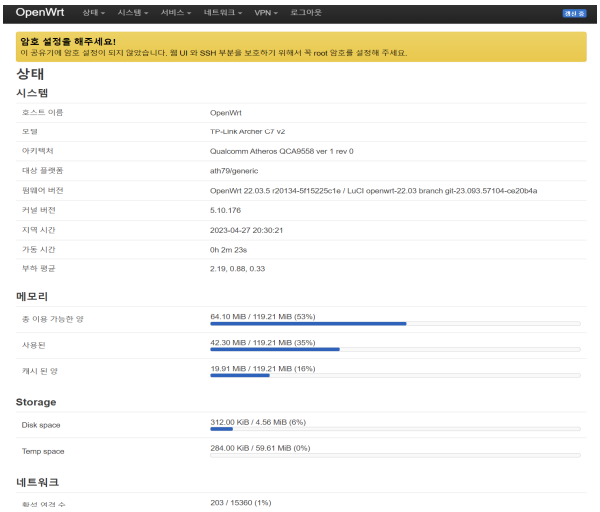

그림 8. 공유기 접속 시 변조 펌웨어의 관리 페이지
Fig. 8. Modified firmware management page when connected to a router

결과적으로 단순히 공격자가 내부 네트워크에 접근하는 것만으로도 공격을 수행할 수 있으며 이러한 과정을 초기에 알아차리거나 데이터를 탈취당한다는 사실을 인지하기는 매우 어렵다는 것을 알 수 있다.

3.5 공격 방지 및 위협 완화 방법

위와 같은 공격을 방지하기 위해선 제조사 및 사용자가 관심을 가지고 지속적인 지원이 제조사 차원에서 필요하다. 2022년 1월 본 연구진이 시도한 방법은 제조사 측에서 라이브러리 및 다른 보안 취약점을 패치하여 취약점이 사라지게 되었다.

또한 위협을 완화하기 위해서 Captive Portal 등의 사용자 엑세스 권한을 확인할 수 있는 시스템을 통해 네트워크에 접속하였어도 이중 인증을 통해 비인가 사용자의 네트워크 이용을 차단할 수 있다[6].

Ⅳ. 결 론

최근 스마트 디바이스가 많이 보급되며 무선 네트워크 사용을 위한 유무선 공유기가 많이 보급되었다. 이러한 공유기는 취약점 혹은 사용자의 실수 등을 통해 데이터를 탈취당할 수 있는 공격이 많이 일어나고 있다.

본 논문에서는 비정상 동작을 수행할 수 있는 버퍼 오버플로 취약점을 시도하여 의도하지 않은 동작을 공유기 내부에서 수행할 수 있도록 하였고, 변조된 펌웨어를 업로드하여 사용자의 데이터를 탈취할 수 있었다.

이를 통해 공유기를 사용하는 사용자가 보안에 더욱 많은 관심을 가져야 하며, 공유기 제조사는 취약점이 존재하면 이를 대비하고 펌웨어 보안이 더욱 강화되어야 함을 보였다.

References

[1] K. Lee, M. Lee, and S. Cho, "DoS attack scenario through wireless router firmware vulnerability", KISSE winter conference, pp. 75-77, Dec. 2015.

[2] SentinelOne CVE, https://www.sentinelone.com/labs/cve-2021-45608-netusb-rce-flaw-in-millions-of-end-user-routers/ [accessed: Jul. 16, 2023]

[3] Zenith, https://github.com/0vercl0k/zenith [accessed: Jun. 10, 2023]

[4] D. Kim, "A study on buffer overflow attacks and defense techniques", Domestic master's thesis Graduate School of Information, Chung-Ang University, Feb. 2013.

[5] C. Qin, J. Peng, P. Liu, Y. Zheng, K. Cheng, W. Zhang, and L. Sun, "UCRF: Static analyzing firmware to generate under-constrained seed for fuzzing SOHO router", Computers & Security, Vol. 128, May 2023 https://doi.org/10.1016/j.cose.2023.103157.

[6] E. Wikman, T. D. Nguyen, and C. Irvine, "ODSS: A Ghidra-based Static Analysis Tool for Detecting Stack-Based Buffer Overflows", Hawaii International Conference on System Sciences, Vol 56, pp. 6726-6735, Jan. 2023.

[7] B. Kim, "Implementing a Fuzzing Agent to Dectect Buffer Overflow Vulnerability", Journal of the Korea Convergence Society, Vol. 12, No. 1, pp. 11-17, Dec. 2021. https://doi.org/10.15207/JKCS.2021.12.1.011.

[8] W.-H. Jung and S.-H. Lee, "Detect and respond to security attacks on wireless routers", The Journal of Korea Information and Communications Society - Industrial Applications, pp. 87-93, Feb. 2016.
[9] C.-H. Son, G.-J. Mun, I.-A. Cheong, and D.-J. Ryu, "Analysis and Countermeasures of Security Vulnerabilities in Wireless Routers", KICS conference, pp. 1193-1194, Jan. 2015.
[10] H.-R. Bae, M.-Y. Kim, S.-K. Song, S.-G. Lee, and Y.-H. Chang, "Security Attack Analysis for Wireless Router and Free Wi-Fi Hacking Solutions", The Journal of Convergence on Culture Technology (JCCT), Vol. 2, No. 4, pp. 65-70, Feb. 2016. https://doi.org/10.17703/JCCT.2016.2.4.65.

저자소개

남 윤 수 (Younsu Nam)

2018년 3월 ~ 현재 : 전북대학교 소프트웨어공학과 학사과정
관심분야 : 소프트웨어공학, 운영체제, 클라우드, 네트워크, 보안

최 선 오 (Sunoh Choi)

2008년 2월 : 고려대학교 컴퓨터학과(이학사 및 이학석사)
2014년 5월 : Purdue대학교 전자및컴퓨터공학부(공학박사)
2014년 ~ 2019년 : 한국전자통신연구원 정보보호연구본부 선임연구원
2019년 ~ 2021년 : 호남대학교 컴퓨터공학과 조교수
2021년 3월 ~ 현재 : 전북대학교 소프트웨어공학과 부교수
관심분야 : 인공지능보안, 네트워크보안, 데이터보안