Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.6, Dec. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.6.1023

타이젠 용 데스크톱 버스 (D-Bus) 권한 우회 취약점 분석 및 자동 탐지*

김 동 성,[1†] 최 형 기[2‡]
[1,2]성균관대학교 (대학원생, 교수)

Automatic Detection and Analysis of Desktop Bus'(D-Bus) Privilege Bypass in Tizen*

Dongsung Kim,[1†] Hyoung-Kee Choi[2‡]
[1,2]Sungkyunkwan University (Graduate student, Professor)

요 약

스마트워치로 대표되는 웨어러블 기기들은 시스템 내에서 민감한 개인정보를 처리하고 저장하기 때문에 높은 보안 기능이 요구된다. 타이젠 운영체제는 애플리케이션에서 특별한 권한을 요구하는 서비스에 접근 시 시스템 데스크톱 버스(D-Bus)를 통하여 요청하는데, 시스템은 해당 애플리케이션의 권한을 검증하여 서비스 접근 허용 여부를 결정한다. 본 논문에서는 타이젠 웨어러블 운영체제의 권한 정책과 접근 제어에 취약한 서비스들을 검출하기 위해 데스크톱 버스 분석 자동화 도구를 소개한다. 자동화 도구는 타이젠 시스템 내 다양한 서비스들에서 권한 검증을 우회할 수 있는 다수의 취약한 호출 메소드들을 발견하였다. 상용 애플리케이션을 제작하여 무선랜 위치 추적, 푸시 알림 수집 등 최소 5개의 취약점에 대한 시연을 하였다.

ABSTRACT

Wearable devices, such as a smart watch and a wrist band, store owner's private information in the devices so that security in a high level is required. Applications developed by third parties in Tizen request for an access to designated services through the desktop bus (D-Bus). The D-Bus verifies application's privileges to grant the request for an access. We developed a fuzzing tool, so-called DAN (the D-bus ANalyzer), to detect errors in implementations for privilege verifications and access controls within Tizen's system services. The DAN has found a number of vulnerable services which granted accesses to unauthorized applications. We built a proof-of-concept application based on those findings to demonstrate a bypass in the privilege examination.

Keywords: Tizen, Wearable, Privilege, Desktop Bus, Access Control, D-Bus, Vulnerability

I. 서 론

스마트워치를 필두로 웨어러블 기기의 수요가 증대됨에 따라 다양한 형태의 제품이 출시되고 있다. 고성능 임베디드 하드웨어를 갖추고 있는 스마트워치는 스마트폰과 연동하여 다양한 사용자 정보를 처리하고 저장한다. GPS(Global Positioning System)를 통한 현재 위치, 심박 센서를 통해 건강

Received(08. 18. 2020), Modified(10. 12. 2020), Accepted(10. 12. 2020)
* 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No, 2019-0-01343, 융합보안핵심인재양성)
† 주저자, kiding@skku.edu
‡ 교신저자, meosery@skku.edu(Corresponding author)

상태, NFC(Near-Field Communication) 신용카드 결제 기능 등 다양하고 민감한 개인정보를 다루기 때문에, 이에 따라 공격자로부터 사용자 정보를 보호하기 위한 다양한 측면에서의 보안 기능이 요구된다.

스마트워치 운영체제로 알려져 있는 타이젠(Tizen)은 삼성이 주도적으로 개발하고 있는 리눅스 기반 오픈소스 운영체제이다[1]. 타이젠은 다양한 산업 요구에 맞추어 차량 내 인포테인먼트(IVI, In-Vehicle Infotainment), 모바일, TV, 웨어러블 등에 사용된다. 운영체제 및 커널 소스 코드는 앱(app) 개발자용 환경과 타이젠을 탑재하고자 하는 하드웨어 개발사를 위해 공개되어 있다[2].

본 논문은 타이젠 웨어러블 2.x 버전을 대상으로 설치된 앱들에 대한 서비스 접근 권한 정책이 정상적용되고 있는지 확인하고자 하였다. 특별히, 앱이 서비스 데몬(daemon) 프로세스에 권한 및 정보를 요청할 때 사용하는 데스크톱 버스(D-Bus)를 대상으로 분석하였다. 그 결과 타이젠을 사용하는 상용 스마트워치 제품에서 정책을 우회하여 접근이 허용되지 않은 정보를 수집하거나 권한이 없는 명령을 실행할 수 있는 취약점을 다수 발견하였다.

II. 배경 지식

타이젠 웨어러블 2.x 버전에서 사용되는 서비스 접근 방식과 권한 정책 관리 방식에 대해 서술한다.

2.1 타이젠 권한(privilege)

서비스란 인터넷이나 블루투스, 카메라, 위치 정보, 알람 등 API(Application Programming Interface)를 통해 시스템이 제공하는 기능을 말한다. 사용자의 민감한 정보를 관리하는 접근이 제한되는 서비스에 앱이 정보를 요청하려면 권한을 부여받아야 한다[3]. 권한은 앱 패키지 내 매니페스트(manifest) 파일에 사용하고자 하는 API에 해당하는 특정 권한 문자열을 추가해서 부여한다. 설치 시 애플리케이션이 요청하는 권한들을 사용자가 허용하면 시스템이 관리하는 규칙에 추가된다.

타이젠 2.x 상에서 모든 프로세스는 루트(root) 사용자 또는 일반 사용자 권한으로 실행된다. 서드파티(third-party) 앱은 모두 일반 사용자로 실행된다. 프로세스 간 커널 내 리소스들에 대한 접근 제한

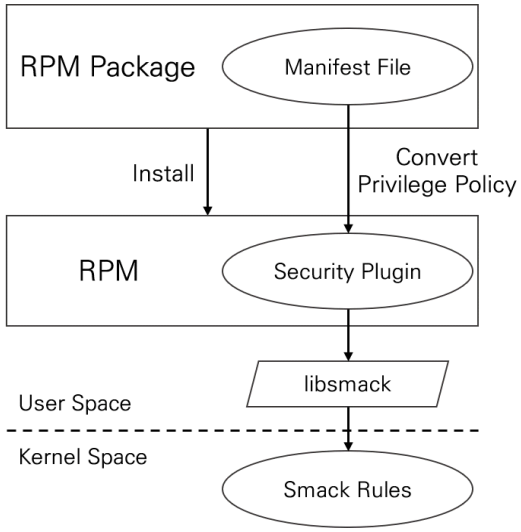

Fig. 1. Generation of Smack rule for Tizen applications in an RPM format at the time of installation.

은 Smack(Simplified Mandatory Access Control in Kernel)에 의해 통제된다. 앱은 설치 시에 Smack으로부터 고유의 레이블을 지정받는다. 각 프로세스와 리소스 간의 접근 권한 정보를 목록 형태로 구성한 레이블(label)을 불러들여 Smack 규칙을 검증한다. 권한이 없는 API 호출은 에러가 반환되고 요청은 실패한다.

Fig.1.을 보면, 빌드된 앱이 포함된 RPM(RPM Package Manager) 패키지가 설치 시, RPM의 보안 플러그인이 매니페스트 파일의 권한 정책을 변환하여 libsmack 라이브러리를 통해 커널 영역 내 Smack 규칙에 반영한다[4].

2.2 타이젠 데스크톱 버스의 권한 검증

데스크톱 버스란 각 애플리케이션과 서비스 데몬이 서로 통신하는 수단이다. 데스크톱 버스는 리눅스에서 주로 사용되는 프로세스 간 통신 방식(Inter-Process Communication)으로, 제공하는 기능을 객체 모델로 정의하여 사용한다. 각 객체는 유일한 패스(path) 이름으로 구분되며, 객체를 호출하기 위한 메소드(method)로 구성된다. 객체는 메소드들의 이름을 기능별로 조합한 인터페이스(interface)를 한 가지 이상 지원하며, org.freedesktop.Dbus.Introspectable가 그 예

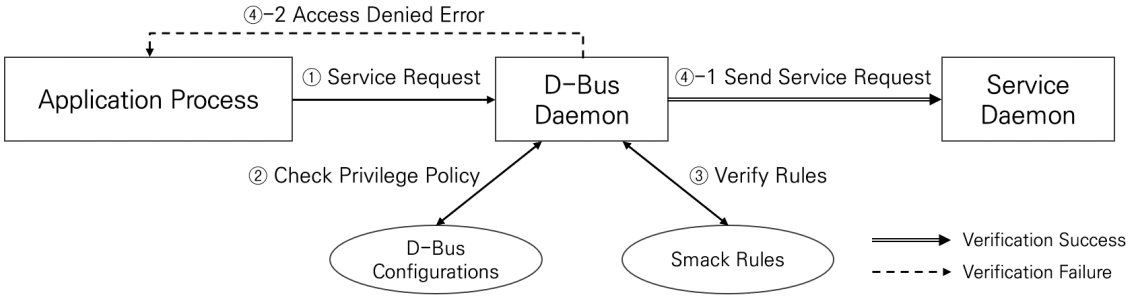

Fig. 2. A four-step verification for application's privileges in D-Bus. D-Bus verifies the privileges against the Smack rule before it connects the application to requested services.

시이다. 각 메소드는 쓰기 및 읽기 등의 속성을 참고할 수 있는 프로퍼티(property) 등으로 구성된다.

애플리케이션은 서비스 자원에 접근하기 위하여 데스크톱 버스에서 사용자용으로 정의된 메소드를 호출하거나 프로퍼티 값을 읽는 등의 요청 메시지를 데스크톱 버스 데몬으로 전송하여 처리한다.

Fig.2.와 같이, 애플리케이션은 사용하고자 하는 서비스의 버스 이름(bus name)과 함께 서비스 요청 정보를 데스크톱 버스 데몬으로 전송한다. 타이젠에서는 리눅스용 데스크톱 버스에 Smack 규칙을 인식하도록 개량된 버전을 사용한다. 요청을 수신한 데스크톱 버스 데몬은 시스템 설정을 통해 앱이 접근하려는 서비스 데몬이 요구하는 권한 정책을 확인한 후 Smack 규칙을 검증하여 요청에 대한 승인 여부를 결정한다. 정상적으로 검증이 되면 데스크톱 버스는 서비스로 애플리케이션의 요청을 전송한다[5].

Fig.2.의 예시처럼 데스크톱 버스는 Smack 규칙 검증을 다른 작업들에 우선적으로 시행한다. 데스크톱 버스에서 정의된 특정 메소드 또는 프로퍼티에 임의의 요청을 전달할 경우, 권한 검증에 실패한 요청들은 접근 거부 오류를 반환한다. 권한 검증은 성공하였지만 입력값에 오류가 있는 요청들은 접근 거부 오류와는 다른 오류를 반환한다. 이런 특징을 이용하면 데스크톱 버스에서 정의된 메소드와 프로퍼티들을 대상으로 임의의 입력값으로 호출한 후 각 호출의 반환값을 접근 권한 유무를 파악하는 오라클(oracle)로 이용할 수 있다. 정리하자면, 반환값이 접근 거부 오류인 경우는 애플리케이션 권한이 없는 것이고 기타 오류를 반환한 경우에는 애플리케이션이 접근 권한이 있는 것이다. 적절한 권한이 없는데도 기타 오류를 반환한 경우에는 데스크톱 버스가 권한 검증에 취약한 경우이다.

권한 검증을 소홀히 하는 데스크톱 버스 서비스들을 파악하고 이를 토대로 공격 시나리오를 작성하였다. 먼저, 데스크톱 버스가 제공하는 서비스 중 권한 검증에 취약한 서비스들을 파악하기 위해 위에서 설명한 오라클을 이용하는 자동화 도구를 설계, 제작 및 배포하였다. 권한이 없는 앱에서 권한이 요구되는 서비스에 접근할 수 있음을 보여주기 위해 검증 코드(PoC, Proof of Concept)를 제작하였다.

III. 데스크톱 버스 분석 자동화 도구

데스크톱 버스의 서비스 접근 권한 검증 시 취약점을 자동으로 검출하기 위한 분석 자동화 도구를 소개한다.

Fig.3.에 도식화된 바와 같이 자동화 도구의 구조는 1) 버스 이름 수집, 2) 객체 탐색 및 분석 그리고 3) 메소드 호출 및 분석 이렇게 3개의 단계로 구성하였다.

분석을 진행하기 위해서는 Fig.3.의 하단에 각각 표시된 추출된 기기 펌웨어 파일 시스템과 디버깅 가능한 실제 대상 기기가 필요하다. 기기에는 여러 가지 명령을 실행할 수 있도록 쉘(shell) 접근과 네이티브 앱을 실행할 수 있는 환경이 구축되어 있어야 한다. 개발자 환경에서 제공하는 SDB(Smart Development Bridge), Tizen Wearable Native Application 템플릿[6]을 사용할 수 있다.

3.1 버스 이름 수집, 1단계

1단계는 서비스의 버스 이름을 수집하는 단계이다. 자동으로 활성화되는 서비스와 동적으로 활성화되는 서비스의 수집 과정이 서로 상이하다. 먼저 자

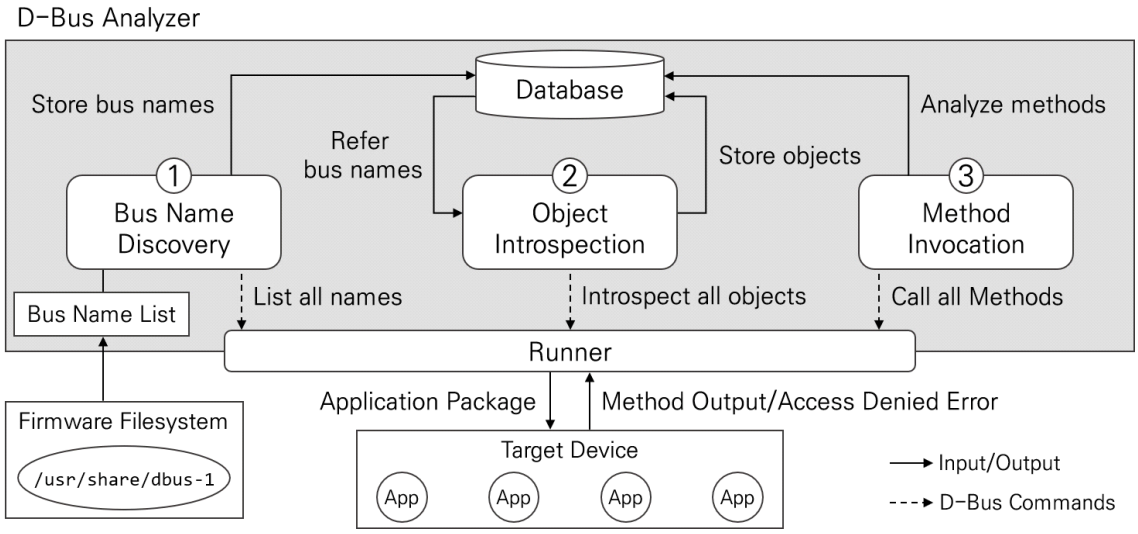

Fig. 3. A structure of a proposed D-Bus analyzer (DAN). The DAN is composed three modules which identify all methods in services in the firmware and tries those methods against a target devices to discover vulnerable methods.

동으로 활성화 가능한 서비스를 확인하기 위해, 추출된 기기 펌웨어 파일 시스템에서 /usr/share/dbus-1 디렉토리 하위에 포함된 모든 서비스 (예, org.freedesktop.system1.service) 파일을 읽어서 각 파일의 이름(name) 필드에 기재된 버스 이름을 수집하는 과정을 거친다[7].

동적으로 활성화되는 서비스를 확인하기 위해 현재 버스 상에 상주하는 모든 버스 이름을 수집한다. 데스크톱 버스의 dbus-send 명령어와 기본 기능인 org.freedesktop.DBus.ListNames 메소드[8][9]을 사용하면 동적으로 활성화된 서비스의 버스 이름들을 수집할 수 있다.

3.2 객체 탐색, 2단계

2단계는, 직전 단계에서 수집한 버스 이름을 대상으로 gdbus의 명령어[11]를 사용하여 각 서비스가 제공하는 하위 객체 구조를 탐색한다. 이 명령어는 org.freedesktop.DBus.Introspectable 인터페이스[12]을 통해 각 객체에 재귀적으로 호출하여 구조 열람을 요청한다.

대상 기기에서 객체 구조 열람을 허용하는 경우에는, 각각의 객체 구조가 텍스트 형식으로 출력된다. Fig.4.는 /org/bluez/hci0 객체 내에 Adapter1 인터페이스를 구성하는 메소드, 시그널 그리고 프로퍼티의 출력 예시이다. 경우에 따라서 정형화된 출력 형식을 따르지 않고 Fig.4.의 11번 줄과 같은 비정형 오류 메시지가 있어서 분석이 어렵다.

분석의 효율성을 높이기 위해 객체 구조의 출력을 Fig.5.와 같은 JSON(JavaScript Object Notation) 구조로 변형한 후 파싱하는 과정으로 변경하였다. 객체 구조와 관련 없는 오류 메시지를 제거한 후 JSON에서 사용되지 않는 각 프로퍼티의 타입 정보와 파싱 불가능한 문자열을 제거한다. 또한 각 메소드의 이름을 수집하면서 매개변수 정보를 제거하는데, 이는 데스크톱 요청이 처리될 때 권한이

```
1  node /org/bluez/hci0 {
2    interface org.bluez.Adapter1 {
3      methods:
4        SetDiscoveryFilter(in  a{sv} properties);
5      signals:
6        AdvertisingEnabled(i slot_id);
7      properties:
8        readonly s Address = '20:55:31:8C:16:C2';
9        readonly s Name = 'BlueZ 5.37';
10   };
11 Error:org.freedesktop.DBus.Error.NoReply:
   Message did not receive a reply (timeout by
   message bus)
12 };
```

Fig. 4. Output of a dbus command for methods, signals and properties of the Adapter1 interface in the /org/blues/hci0 object.

```
{
  "/org/bluez/hci0": {
    "org.bluez.Adapter1": {
      "methods": ["SetDiscoveryFilter"],
      "signals": ["AdvertisingEnabled"],
      "properties":
        { "Address": "20:55:31:8C:16:C2",
          "Name": "BlueZ 5.37" }
    }
  }
}
```

Fig. 5. Output in Fig. 4. is transformed to a JSON format by the proposed tool DAN for an efficient and accurate analysis.

먼저 검증되고 매개변수 일치 여부는 이후에 확인되기 때문이다.

구조 열람이 가능한(introspectable) 객체는 프로퍼티 등을 통해 민감한 정보를 유출할 위험성이 존재한다. Fig.5.를 보면 Fig.4.에서 확인된 객체 구조뿐 아니라, 7번째 및 8번째 줄에서 프로퍼티를 통해 대상 기기의 블루투스(Bluetooth) 주소와 사용하고 있는 시스템 라이브러리 버전 정보가 공개되고 있다.

3.3 메소드 호출, 3단계

이번 단계는 수집한 메소드를 직접 호출하고 분석하는 단계이다. 직전 단계에서 JSON 구조로 변환된 객체를 재귀적으로 탐색하면서 가능한 모든 인터페이스의 모든 메소드에 대한 호출을 시도한다.

호출을 시도할 때는 무작위 데이터를 매개변수로 사용한다. 이는 해당 메소드를 실제로 실행하지 않고 어떤 종류의 오류 메시지를 발생하는지만 수집하기 위함이다. 호출이 완료된 후, 각 메소드가 발생시킨 메시지를 수집하고 객체, 인터페이스, 그리고 메소드를 특정하며 에러 메시지를 분석한다.

2.2장에서 언급한 것처럼, 특정한 권한이 없는 애플리케이션에서 메소드를 호출하면 검증이 정상적으로 수행될 경우에 접근 거부 오류가 발생한다. 접근 거부 외 다른 오류를 반환하는 메소드들은 잠재적으로 권한 정책을 우회할 수 있는 위험이 존재한다. 접근 거부 없이 호출 가능한 메소드들의 정보를 추가 분석을 위하여 저장한다.

IV. 실험 및 결과

3장에서 설명한 분석 자동화 도구 적용을 위해 본 연구에서는 실험 대상으로 상용 제품인 삼성 기어 S2(SM-R720)와 2017년 11월 20일 배포된 QJ2 펌웨어를 사용하였다[13]. 자동화 도구에 사용된 파일 시스템은 2017년 8월 23일에 배포된 QH1 펌웨어를 기반으로 하였다. 두 버전 모두 타이젠 2.3.2.2 버전을 탑재하고 있다.

실험 결과, 총 227개의 서비스 버스 이름을 수집하였으며, 그 중 객체 구조 열람이 가능하여 프로퍼티를 통해 민감한 정보를 유출할 가능성이 있는 경우는 개발자 쉘에서 148개(65.2%), 네이티브 앱에서 151개(66.5%)였다. 수집된 메소드 중 총 2368개에 호출을 시도하였을 때, 접근 거부가 없어 잠재적으로 권한 정책이 우회될 수 있는 위험이 존재하는 경우는 852개(36.0%)였다.

무선랜 (Wi-Fi), 블루투스, 화면, 알림, 이메일 등 다양한 서비스들의 메소드들을 권한이 없는 애플리케이션들이 데스크톱 버스에 권한 검증을 우회하여 호출하는 것이 가능하였다. 실험을 통해 수집된 잠재적 위험 메소드 중 서비스를 기준으로 일부를 선별하여 추가적인 분석을 시행하였고, 이를 통해 발견된 권한 정책 우회 취약점 중 일부를 아래 소단원에서 설명한다.

발견된 다수의 취약점은 실험 대상으로 선정한 삼성 기어 S2 뿐 아니라 S3 프론티어 및 LTE 버전에서도 적용됨을 확인하였다.

4.1 검증 코드

데스크톱 버스 자동화 도구의 정확도를 실험하고 발견한 취약점들의 개념 증명을 위해 최소한의 권한을 가지고 있는 앱을 연구팀에서 제작하였다. 이 앱은 삼성 기어 S2에서 동작하면서 본 연구팀에서 발견한 취약점을 이용하여 앱에 주어지지 않은 권한을 실행하는 증명 코드로 사용하고자 한다. 첫번째 증명은 주변 무선랜 신호 탐색을 이용하여 단말의 물리적 위치를 파악하는 것이고 두번째 증명은 모바일 단말에서 스마트 워치로 무선통신 채널로 전송하는 푸시 알람들의 내용을 읽기 권한이 없는 증명 코드가 중간에서 가로채서 공격자에게 전송하는 공격이다.

강조하지만, 개념 증명용 코드는 단순한 최소한의 internet(인터넷 접속 용)과 network.get(네트워

크 상태 정보 수집) 권한만을 사용자에게 요구한다. 증명용 공격을 위해서는 이 두가지 외에 상위 권한을 요구하지만 연구팀이 발견한 취약점을 이용하면 별도의 권한없이 공격이 가능하였다.

4.2 무선랜 (Wi-Fi) 위치 추적

리눅스는 무선랜 탐색과 접속을 위해 오픈소스로 제공되는 wpa_supplicant 모듈을 사용한다. 애플리케이션에서의 무선랜 접속을 위하여 데스크톱 버스 서비스의 일종인 fi.w1.wpa_supplicant가 제공된다. 이 서비스를 통해 권한을 보유한 애플리케이션들은 네트워크 장치들의 추가 및 삭제, 네트워크 탐색, 추가 및 삭제 등 무선 네트워크 관련된 일련의 통제와 제어를 할 수 있다.

본 연구팀은 개발한 데스크톱 버스 자동화 도구를 통해서 권한이 없는 애플리케이션에서 fi.w1.wpa_supplicant에서 제공하는 모든 메소드들을 읽고 호출할 수 있음을 확인하였다. 이를 이용하면 무선랜 접근 및 제어 권한이 없는 애플리케이션이 기기를 소유한 사용자의 물리적 위치를 알아낼 수 있다.

애플리케이션에서 공격은 다음과 같이 진행된다. 취약한 wap_supplicant 서비스를 통해 무선 네트워크 스캔을 요청하면, Fig.6.에서 보는 것과 같이

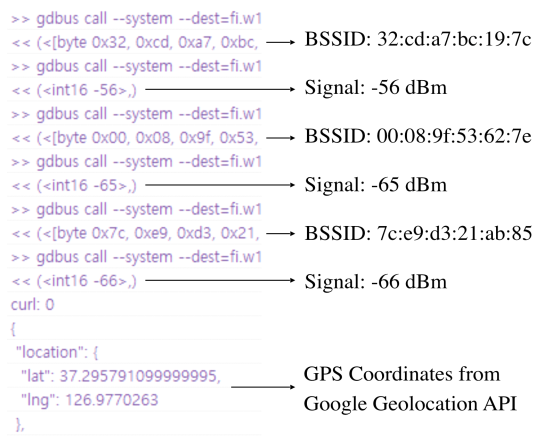

Fig. 6. Output returned by the fi.w1.wpa_supplicant service. Using the GPS coordinates displayed in the bottom one can trace device's current locations. Because of such privacy issues the service must validate application's privileges before it grants a request for an access.

사용자 단말 주변 네트워크들의 BSSID(Basic Service Set Identifier)와 무선 신호 세기를 반환한다. 이 값들을 사용하여 Google Maps Geolocation API[10]에 요청하면 해당하는 GPS 좌표를 획득 할 수 있다. Maps Geolocation은 WPS(Wi-Fi Positioning System) 방식을 이용하여 측위한 단말의 현재 위치 정보를 제공하는 서비스이다. 사용자가 무선랜의 위치 추적을 금지한 경우에도 좌표 획득이 가능하였다.

4.3 푸시(Push) 알림 수집

전화, 문자, 메일 등 푸시 알림은 블루투스나 셀룰러 네트워크 등을 통해 기기로 전달되어 표시된다. com.samsung.wnoti는 알림 관련 기능을 관리하는 데스크톱 버스 서비스 일종으로, 알림 전체 삭제, 특정 앱 알림 중단은 물론 특정 동작을 실행하고 메타데이터를 수집하는 등의 기능을 담당한다.

작성한 증명 코드 애플리케이션에서 추가 권한 없이 기기 상에 표시되고 있는 모든 알림 메시지 내용을 수집하는 공격이 가능하다. 이 서비스를 통해 ClearAll 메소드를 호출하면 스마트워치의 알림 메시지들을 모두 삭제할 수 있다. 또한, GetCategories 메소드를 호출하면, 모든 알림 메시지의 정보를 포함한 JSON 데이터를 Base64 인코딩한 데이터가 반환되며, 공격자는 이를 디코딩하여 열람한다. 알림을 통해 인증 정보가 수신된 경우, 공격자는 이를 토대로 금융이나 메일 계정 탈취와 같은 악의적인 행위를 수행할 수 있다.

4.4 기타 서비스 취약점 및 추가 연구

앞 소단원에서 설명한 경우 외에도 다양한 서비스들에서 다음과 같은 취약점이 존재한다.

- org.projectx.bt_core: 블루투스 어댑터를 조작하는 서비스이다. 어댑터 활성화 및 초기화, LE(Low Energy) 활성화, 공장 테스트 모드 등을 추가 권한 없이 조작할 수 있다.
- org.tizen.capability.manager: 기기와 페어링된 스마트폰의 기능 정보를 관리하는 서비스이다. 이름과 모델명, 운영체제 종류와 버전, 해상도 등의 정보를 추가 권한 없이 수집할 수 있다.
- org.tizen.system.busactd: 최고 관리자만 가

능해야 하는 systemd 서비스 관리자의 조작을 대신 수행할 수 있게 해주는 서비스이다. smack.mount, log_dump, xorg 등 민감한 시스템 서비스의 정보를 수집하거나 중단, 재시작 등의 조작이 개발자 쉘에서 가능하다.

- 동일한 타이젠 버전을 탑재한 다른 기기에서 제시한 취약점이 동작하는지 실험하였다. 미국에서 판매되는 삼성 Family Hub 2.0과 인도에서 판매되는 Samsung Z4 단말에서도 취약점이 동작하는 것을 확인하였다.
- 갤럭시 앱스토어는 위해한 응용프로그램들을 차단하기 위해 정적분석과 화이트리스트 방식을 사용하여 검열한다. 연구팀이 개발한 개념 증명 코드는 응용프로그램에서 필요가 없는 dbus 함수 호출을 다수 포함해서 검열을 통과하지 못하였다. Eldbus라는 라이브러리에서 제공하는 dbus의 wrapper 함수를 사용하여 검열을 통과하고 앱스토어에 등재하였다.

4.5 결과물

본 연구를 통해 밝혀진 취약점들은 제조회사에 모두 보고하였고 제조사로부터 취약점들의 우수성을 인정받아 버그 바운티(bug bounty)로 미화 3000불을 수령하였다. 산출물로는 2018년 이번 연구와 관련된 취약점 총 11개를 보고하였고 CVE-2018-16262로부터 CVE-2018-16272까지 11개의 CVE(Common Vulnerabilties and Exposures)를 등록하였다. 초기에 얻은 결과물들을 종합해서 DefCon 2018에 발표하였다.

V. 결 론

본 논문에서는 타이젠 웨어러블 2.x 버전 운영체제의 서비스 접근과 권한 정책 관리 방식에 대해 연구하고, 각 앱에 대한 서비스 접근 권한 정책이 올바르게 적용되고 있는지 검증하고자 하였다. 이를 위해 문제의 소지가 있는 서비스를 검출하기 위한 데스크톱 버스 분석 자동화 도구를 설계하고 상용 기기와 펌웨어에 적용하였다.

그 결과, 분석한 서비스 중 약 66%가 데스크톱 버스 객체 구조 열람이 가능하며 이를 통해 민감한 정보를 유출할 위험성이 있음을 발견하였다. 또한 분석한 메소드 중 36%가 접근 거부 없이 호출 가능하여 권한 정책이 우회될 수 있는 위험이 존재한다.

위험 메소드를 선별하여 추가적인 분석을 수행한 결과, 권한 정책을 우회하여 시스템을 조작하거나 사용자의 위치 정보나 개인 정보에 접근할 수 있는 취약점을 다수 발견하였다. 마지막으로, 발견된 취약점을 다른 기기에도 동일하게 적용 가능함을 보였다.

타이젠 3.0 버전은 본 연구에서 주목한 2.x 버전과 다른 보안 구조를 가지고 있다[14]. 향후 잠재적 위험 메소드를 추가적으로 선별하여 분석을 진행함과 동시에, 본 연구를 기반으로 하여 타이젠 웨어러블 3.x를 대상으로 연구를 확장할 예정이다.

References

[1] S. Waltzer, Global Smartwatch OS Market Share by Region : Q2 2017, Strategy Analytics, Aug. 2017.

[2] V. Amorim, S. Delabrida, and R. Oliveira, "A Constraint-Driven Assessment of Operating Systems for Wearable Devices," In Proceedings of *SBESC*, pp. 150-155, Nov. 2016.

[3] O. Gadyatskaya, F. Massacci, and Y. Zhauniarovich, "Security in the Firefox OS and Tizen Mobile Platforms," *Computer*, Vol. 47, No. 6, pp. 57-63, Jun. 2014.

[4] Tizen Wiki, "Tizen 2.X Architecture," https://wiki.tizen.org/Security/Tizen_2.X_Architecture, Dec. 2020.

[5] Tizen Wiki, "Tizen 2.X dbus," https://wiki.tizen.org/Security/Tizen_2.X_dbus, Dec. 2020.

[6] GitHub, "kiding/dan: Automatic privilege evaluation of D-Bus services," https://github.com/kiding/dan, Dec. 2020.

[7] Tizen Developers, "Introduction to Native Applications," https://developer.tizen.org/development/training/native-application, Dec. 2020.

[8] K. Vervloesem, "Control your Linux desktop with D-Bus," Linux Journal, Vol. 199, No. 3, Nov. 2010.

[9] dbus, "D-Bus Specification," https://dbus.freedesktop.org/doc/dbus-specification.html, Dec. 2020.
[10] Google Maps Platform, "Google Maps Geolocation API," https://developers.google.com/maps/documentation/geolocation/, Dec. 2020.
[11] GNOME Developer Center, "gdbus: GIO Reference Manual," https://developer.gnome.org/gio/unstable/gdbus.html, Dec. 2020.
[12] M. Marhefka et al., "Dfuzzer: A D-Bus Service Fuzzing Tool," In Proceedings of *ICSTVV*, pp. 383-389, Cleveland, OH, USA, Apr. 2014.
[13] Samsung Mobile, "Notify Update," http://goo.gl/K5iGw7, Dec. 2020.
[14] Tizen Wiki, "Tizen 3.X Overview," https://wiki.tizen.org/Security/Tizen_3.X_Overview, Dec. 2020.

〈저자소개〉

김 동 성 (Dongsung Kim) 학생회원
2017년 2월: 성균관대학교 컴퓨터공학과 학사 졸업
2018년 8월: 성균관대학교 전자전기컴퓨터공학과 석사 수료
2019년 10월~2020년 11월: 리디 주식회사 근무
〈관심분야〉 시스템 및 네트워크 보안

최 형 기 (Hyoung-kee Choi) 정회원
1992년 2월: 성균관대학교 전자공학과 졸업
1996년 2월: Polytechnic University in Brooklyn, NY 석사
2001년 2월: Georgia Institute of Technology in Atlanta, GA 박사
2001년 1월~2004년 12월: Cisco 근무
2004년 3월~현재: 성균관대학교 소프트웨어대학 교수
〈관심분야〉 네트워크 보안, 리버스 엔지니어링