# OAuth 프레임워크의 취약점 개선을 위한 클라우드 및 생체인증 기술 활용 방안

박진성, 이창훈†
서울과학기술대학교 컴퓨터공학과
e-mail: {jseongpark157, chlee}@seoultech.ac.kr

# Utilization of Cloud and Biometrics to improve Vulnerability of OAuth Framework

Jin-Seong Park, Chang-Hoon Lee†
Dept of Computer Science and Engineering, Seoul National University of Science & Technology

요 약

Open API의 사용이 대중화되면서 대형 포털사이트 및 대형 소셜 네트워크 서비스에 입력한 개인정보를 바탕으로 사용자들이 원하는 서비스를 제공하는 외부 서비스(Third-party application)의 수가 계속 증가하고 있다. 이러한 외부 서비스들이 정보 제공자에게 개인정보를 요청할 때, 외부 서비스가 권한 부여를 받은 정당한 요청을 하는지 여부를 확인하기 위해 OAuth 인증 프레임워크를 사용한다. 그러나 외부 서비스가 유효한 요청을 하고 있음을 판단하기 위해 사용하는 엑세스 토큰을 탈취하면 이를 이용하여 로그인 과정을 우회할 수 있으며, 개인정보 또한 쉽게 취득 가능하다. 본 논문에서는 이러한 취약점을 해결하기 위해 OAuth 인증 프레임워크에 2차적인 인증과정을 추가하여 엑세스 토큰의 탈취를 방지하여 사용자의 계정정보와 개인정보를 보호하는 프레임워크를 제시한다.

## 1. 서론

인터넷의 대중화로 인해 사용자가 필요한 서비스를 제공하는 웹사이트 및 애플리케이션의 수는 점차 늘어가고 있다. 2010년, 약 2억 개의 사이트에서 2018년 현재, 약 19억 개로 증가하였다[1]. 또한 Open API(Application Programming Interface)의 사용이 대중화되면서 사용자에게 서비스를 제공하는 웹사이트 및 애플리케이션을 구현하는 것이 더욱 간단해졌다. 이는 Open API를 제공하는 서비스 제공자(Service Provider)가 보유하고 있는 사용자의 개인 정보에 접근하기 위해서 외부 서비스(Third-party application)가 정당한 접근 권한을 가지고 있음을 인증할 필요성을 증대시켰다. 기존에는 인증방식의 표준의 부재로 인해 계정정보를 통해 인증을 수행하였는데, 이는 외부 서비스가 사용자의 계정 정보를 수집하게 하는 결과를 야기하였다. 이는 보안상 취약한 구조로 개선이 필요하다. 사용자의 계정정보를 제공하지 않고 서비스 제공자가 보유하고 있는 데이터에 접근 권한을 부여하기 위해 고안된 것이 OAuth 프로토콜이다. OAuth는 사용자의 계정정보를 제공받지 않아도 외부 서비스가 서비스 제공자의 제한된 데이터에 접근할 수 있도록 권한을 부여하는 보안 프로토콜을 제공한다. OAuth 프로토콜은 인증과 권한부여를 동시에 할 수 있다. 또한 사용자가 권한부여에 대한 권한을 지니고 있기 때문에 언제든지 부여한 권한을 회수 가능하다는 특징이 있다. 2008년 OAuth의 비공식회합 이래로 트위터, 구글, 페이스북, 네이버 등 많은 업체에서 OAuth 프로토콜을 지원하고 있다. OAuth 프로토콜은 현재 기능을 확장시켜 프레임워크로 만든 2.0버전이 “OAuth 2.0 Authorization Framework”라는 이름으로 IETF(Internet Engineering Task Force)에 제정되었으며 RFC6747로 발표되었다[2].

사용자는 OAuth 인증 프레임워크를 이용하여 외부 서비스를 이용하고자 할 때, 서비스 제공자의 계정 정보를 통해 엑세스 토큰을 발급받아 외부 서비스에게 서비스 제공자가 보유한 데이터에 접근할 수 있는 권한을 부여한다. 그러나 계정정보를 외부 서비스에게 제공하지는 않지만, 서비스 제공자의 데이터에 접근 권한을 가지는 토큰을 탈취당하면 사용자의 개인정보가 유출될 수 있다는 문제점이 존재한다. 이는 엑세스 토큰이 발급된 이후 사용자의 로컬 디바이스에 저장되기 때문에 탈취당하기 쉽다는 점에서 더더욱 문제가 된다. 안드로이드 환경에서 모바일 기기를 강제로 루팅하는 악성코드가 존재하며[3], 엑세스 토큰을 이용하여 로그인을 우회하는 방법 또한 연구된 바 있다[4].

이러한 문제점을 해결하기 위해서 본 논문에서는 엑세

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-00344, 최신 모바일 기기에 대한 암호해독 및 포렌식 분석)

† 교신저자, chlee@seoultech.ac.kr(Corresponding author)

스 토큰을 클라우드 서버에 저장한 뒤, 2차적인 인증과정을 OAuth 인증 프레임워크에 추가하는 방법을 제시하고자 한다. 2장에서는 OAuth와 본 연구에서 제안하는 인증 프레임워크의 기반이 되는 클라우드 및 생체인증 기술과 관련된 연구들을 소개하고, 3장에서 문제점을 해결하기 위한 수정된 프레임워크를 제시한다. 4장에서는 결론 및 향후 연구 방안을 논한다.

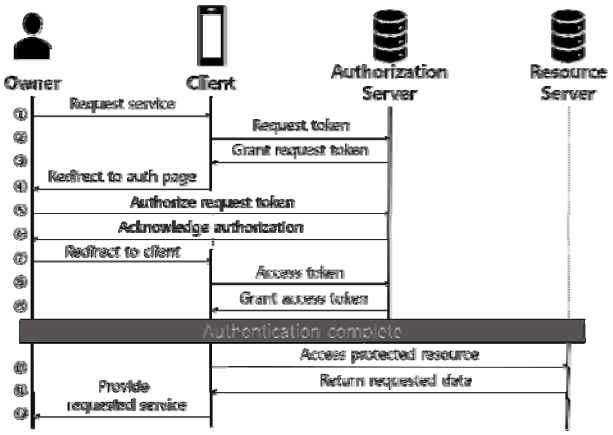
## 2. 관련연구

### 2.1 OAuth 2.0 인증 프레임워크

OAuth 2.0 인증 프레임워크는 이전 1.0a 버전과 호환되지는 않지만 1.0a 버전에서 불편한 점으로 지적되었던 모바일에서 사용성 문제, 부족한 기능과 규모, CPU를 과다 소비하는 문제를 해결하기 위한 여러 가지 개선점을 포함하고 있다. 이에 따라 OAuth 서비스 제공자들도 1.0a 버전에서 2.0 버전으로 넘어가는 추세이며, 따라서 본 논문에서도 OAuth 2.0 인증 프레임워크를 기준으로 설명한다.

OAuth 인증 프레임워크는 크게 자원 소유자(Resource Owner), 자원 서버(Resource Server), 인증 서버(Authorization Server), 클라이언트(Client)로 이루어지며, 자원 서버와 인증 서버가 동일한 개체인 경우도 존재한다. 자원 소유자는 사용자를 지칭하며, 서비스를 사용하는 주체이다. 자원 소유자는 클라이언트가 제공하는 서비스를 사용하기 위해 인증 서버에게 계정 정보를 입력하여 클라이언트가 엑세스 토큰을 발급받을 수 있게 함으로써 권한 부여를 수행한다. 클라이언트는 외부 서비스를 지칭하며, Open API를 이용하여 자원 서버에 저장된 사용자의 개인 정보를 이용하려 하는 애플리케이션이다. 클라이언트는 자원 서버에 접근하기 위해서는 인증 서버에게 엑세스 토큰을 이용하여 유효한 요청 여부를 판별 받은 후에, 허가받은 데이터에 접근하여 이용할 수 있다. 인증 서버는 엑세스 토큰을 발급하고, 사용자의 인증 및 권한부여를 수행한다. 자원 서버는 실제 사용자의 데이터를 보관하는 서버이며, 자원 서버에 접근하기 위해서는 인증 서버가 발급하는 엑세스 토큰이 필요하다[2]. (그림 1)은 OAuth 인증 프레임워크의 전체 프로세스를 나타낸다.

자원 소유자는 클라이언트가 제공하는 서비스를 이용하기 위해 클라이언트에게 서비스를 요청한다(①). 클라이언트는 인증을 위해 요청 토큰을 인증 서버에 요청하고(②), 인증 서버는 요청 토큰을 발급하여 클라이언트에게 전송한다(③). 클라이언트는 인증을 위해 자원 소유자에게 인증 페이지를 연결시켜주고(④), 자원 소유자는 인증 서버에게 요청 토큰을 인증한다(⑤). 인증 서버가 인증 결과를 자원 소유자에게 전달하면(⑥), 자원 소유자는 그 결과를 다시 클라이언트에게 전달한다(⑦). 클라이언트는 인증 결과를 바탕으로 엑세스 토큰을 인증 서버에게 요청하고(⑧), 인증 서버는 엑세스 토큰을 발급하여 클라이언트에




(그림 1) OAuth 2.0 인증 프레임워크 프로세스

게 전달한다(⑨). 클라이언트는 엑세스 토큰을 이용하여 자원 서버에 접근하여 데이터를 요청하고(⑩), 자원 서버는 요청받은 데이터를 클라이언트에게 전달한다(⑪). 클라이언트는 획득한 사용자의 데이터를 바탕으로 서비스를 제공하게 된다(⑫).

클라이언트는 발급된 엑세스 토큰을 보유하고 있다가 OAuth 서비스를 이용하는 API를 호출할 때, HTTP의 헤더에 엑세스 토큰을 포함시켜서 요청을 보낸다. 그 후에 인증 서버는 HTTP의 헤더에 포함된 엑세스 토큰을 검사하여 유효한 요청인지 확인한 후에 자원 서버에 대한 접근을 허가한다. 그러나 엑세스 토큰을 클라이언트가 보유하고 있기 때문에, 엑세스 토큰은 로컬 디바이스, 특히 모바일 환경에서는 사용자의 스마트폰 내부에 저장되게 된다. 따라서 클라이언트가 엑세스 토큰을 보관하는 위치를 찾아내서 엑세스 토큰을 복사함으로써 쉽게 탈취 가능하다는 취약점을 가지게 된다. 남기훈 외 3명은 이러한 취약점을 바탕으로 엑세스 토큰을 탈취하여 타 스마트폰 기기 및 가상환경에서 로그인을 우회하는 연구를 진행하였다[4][5].

### 2.2 클라우드

클라우드는 네트워크로 연결되어 언제 어디서나 사용할 수 있도록 구성된 컴퓨팅 자원을 지칭한다. 클라우드는 필요한 만큼의 컴퓨팅 자원을 요청하여 사용할 수 있으며, 최소한의 관리 노력으로 컴퓨팅 자원을 할당하고 해제할 수 있다. 클라우드 서비스는 클라우드에 데이터를 저장하여 사용하는 서비스를 지칭한다. 데이터를 단순히 저장할 뿐만 아니라 컴퓨팅 자원을 할당받아 사용하는 것도 가능하기 때문에, 따로 프로그램을 설치하지 않아도 웹에서 애플리케이션의 기능을 사용할 수 있다. 클라우드 서비스는

가장 기본적인 모델로서 가상 머신과 기타 컴퓨팅 자원을 제공하는 IaaS(Infrastructure as a Service), 컴퓨팅 플랫폼을 제공하는 PaaS(Platform as a Service), 클라우드 상에서 서비스를 제공하는 SaaS(Software as a Service) 세 종류의 모델이 있다[6][7].

### 2.3 생체인증

기존의 인증 방식인 패스워드 혹은 물리적 토큰에 의한 인증은 망각이나 노출, 도난, 분실 등의 우려가 존재한다. 반면 생체인증은 개인이 지니고 있는 생체정보를 이용해 인증하는 방법으로 인증 시에 측정한 결과를 사전에 등록한 생체정보와 비교하여 인증하는 방법이다. 생체인증은 망각 및 노출, 도난, 분실의 위험성이 낮고, 특히 제 3자가 인증하는 것을 방지할 수 있으며, 계정 및 토큰을 관리할 필요가 없어 편리하다. 현재 모바일 환경에서 지문인식은 보편화 되었으며, 홍채 인식과 얼굴 인식도 지원하는 기기들이 출시되고 있다. A Alotaibi 외 1명은 OAuth 인증 프레임워크에서 중간자 공격(Man-in-the-middle attack)을 방지하기 위해서 생체 인증을 이용하는 연구를 진행하였다[8].

## 3. 개선된 OAuth 인증 프레임워크

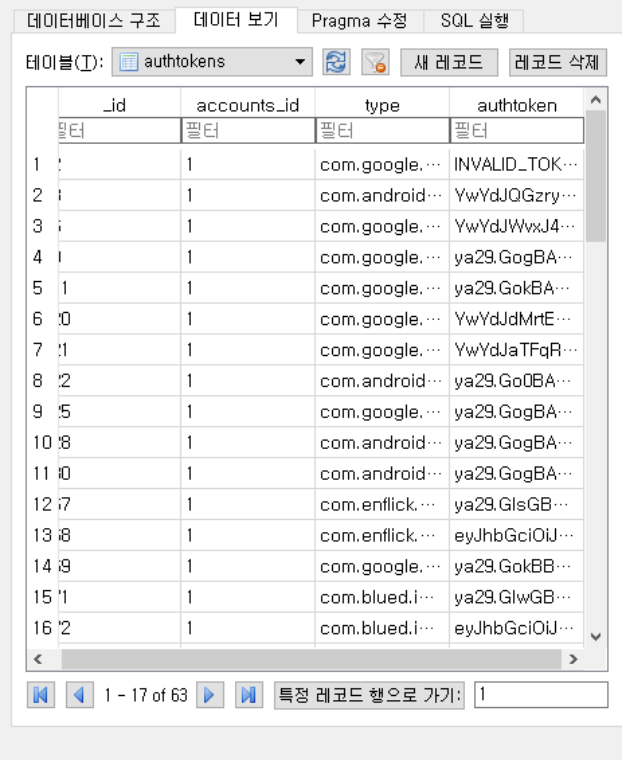
### 3.1 기존 프레임워크의 문제점

기존의 OAuth 인증 프레임워크의 취약점은 엑세스 토큰을 클라이언트가 소유하고 있고 엑세스 토큰이 위치한 경로에 접근 권한만 있다면 별다른 인증절차 없이 엑세스 토큰에 접근할 수 있다는 점에서 기인한다. 실제로 안드로이드 스마트폰을 이용하기 위해서 필수적인 구글 계정을 타 기기에서 로그인하는 방법이 여러 차례 연구되었다[4][5][9]. (그림 2)는 실제 구글 계정의 엑세스 토큰을 추출한 뒤, 그 내용을 나타낸 것이다.

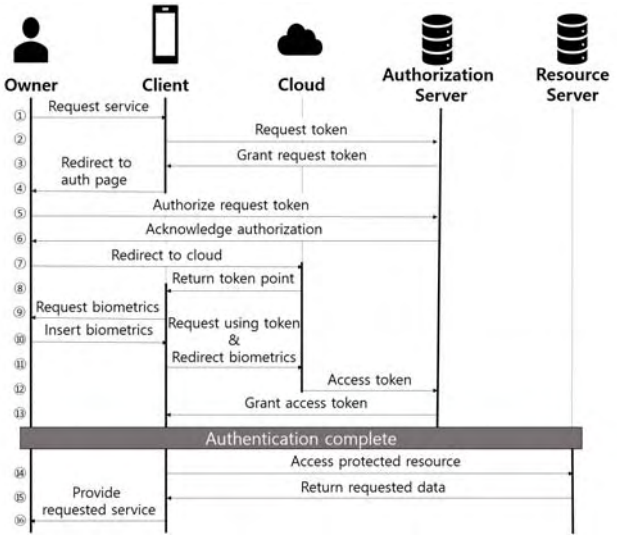
### 3.2 클라우드 기반 인증 프레임워크

본 논문에서는 클라우드 서버를 클라이언트와 인증 서버 사이에 배치하여 엑세스 토큰을 클라우드 서버에 보관하고, 엑세스 토큰을 사용할 때는 제 3자가 인증하는 것을 방지하기 위해서 생체 인증을 이용하는 프레임워크를 제안한다. 또한 생체 인증 데이터를 로컬 디바이스에 저장한 뒤에 일치여부를 판별할 경우, 엑세스 토큰을 탈취하는 방법과 마찬가지로 생체 인증 데이터를 탈취할 가능성이 존재하므로, 생체 인증 데이터도 클라우드 서버에 저장된다. (그림 3)은 본 논문에서 제시하는 인증 프레임워크의 프로세스를 나타낸다.

본 논문에서 제시하는 클라우드 기반 인증 프레임워크의 인증 프로세스와 기존 프레임워크의 인증 프로세스는 ⑥까지 동일하다. 그러나 본 논문에서 제시하는 프레임워




(그림 2) 구글 계정의 엑세스 토큰




(그림 3) 클라우드 기반 인증 프레임워크 프로세스

크는 클라이언트가 아니라 클라우드로 엑세스 토큰을 전달한다(⑦). 클라우드는 전달받은 토큰의 위치를 클라이언트에게 전달하고, 클라이언트는 클라우드에서의 토큰 위치 데이터만을 보관하게 된다(⑧). 클라이언트는 엑세스 토큰을 이용하고자 하면 자원 소유자에게 생체 정보를 요구한다(⑨). 자원 소유자가 생체 정보를 클라이언트에게 넘겨주면(⑩), 클라이언트는 엑세스 토큰 사용 요청과 생체 정보를 클라우드에 전송한다(⑪). 클라우드는 인증 요

청이 들어오면 넘겨받은 생체 정보와 기존에 입력된 생체 정보를 비교한 뒤, 일치한다고 판단되면 인증 서버에 엑세스 토큰을 전송한다(⑫). 인증 서버는 전송된 엑세스 토큰을 바탕으로 클라이언트가 자원 서버에 접근하는 것을 허가하며(⑬), 이후 클라이언트는 자원 서버에 접속하여 필요한 사용자의 데이터를 받아서 사용하는 과정은 기존과 동일하다(⑭,⑮,⑯).

## 4. 결론

Open API의 사용은 앞으로 더욱 많아질 것이며, 이에 따라 OAuth 인증 프레임워크를 사용하는 애플리케이션도 증가할 것이다. 따라서 OAuth 인증 프레임워크의 취약점을 보완하는 것은 굉장히 중요한 과제가 된다. 본 논문에서는 엑세스 토큰을 탈취하여 인증되지 않은 사용자가 자원 서버에 저장된 개인 정보에 접근하는 것을 방지하기 위해 클라우드와 생체인증 기술을 이용한 인증 프레임워크를 제시하였다.

본 연구에서 제안하는 프레임워크는 사용자의 엑세스 토큰이나 생체정보와 같은 민감한 정보를 사용하기 때문에 인증 정보를 저장하는 주체에 대한 연구도 수행되어야 한다. 클라우드 서버를 클라이언트, 인증 서버, 혹은 제 3에 해당하는 기업 중 어느 곳에서 관리를 해야 하는가에 대한 관리 소재를 정하는 문제에 대한 향후 연구가 필요하다. 또한 기존 프레임워크에 비해 클라우드 서버를 추가적으로 사용하게 됨으로 기존에 비해 약간의 성능저하가 발생할 수 있기 때문에 이러한 성능 저하를 최소화하기 위한 최적화 방안에 대해 추가적인 연구가 필요하다.

## 참고문헌

[1] Internet Live Stats, Available on : http://www.internetlivestats.com/total-number-of-websites/

[2] D. Hardt, Ed., “OAuth 2.0 Authorization Framework”, Internet Engineering Task Force (IEFT) RFC 6749, Oct 2012

[3] AhnLab ASEC Blog, Available on: http://asec.ahnlab.com/259

[4] 남기훈, et al. 안드로이드 환경의 OAuth 프로토콜을 이용한 원격지 데이터 수집 방법 연구. 정보보호학회논문지, 2018, 28.1: 111-122.

[5] Kim Jinouk, Jungsoo Park, Long Nguyen-Vu, Souhwan Jung, “A Study on Vulnerability Prevention Mechanism Due to Logout Problem Using OAuth”, Journal of The Korea Institute of Information Security & Cryptography, 27(1), pp. 5-14, Feb 2017.

[6] Robison S., A Bright “Future in the Cloud” Financial Times, March 4, 2008

[7] J. S. Park, Y. M. Bae, S. J. Jung, “Technical analysis of Cloud Storage for Cloud Computing”, Journal of the Korea Institute of Information and Communication Engineering, Vol. 17, No.5, pp. 1129-1137, 201

[8] ALOTAIBI, Aziz; MAHMMOD, Ausif. Enhancing OAuth services security by an authentication service with face recognition. In: Systems, Applications and Technology Conference (LISAT), 2015 IEEE Long Island. IEEE, 2015. p. 1-6.

[9] Choi Jongwon, Yi Jeonghyun, “Analysis on Personal Information Leakage of Google Account App on Android”, Journal of Digital Forensics, 8(2), pp. 65-81, Dec 2014.