한국컴퓨터정보학회 하계학술대회 논문집 제24권 제2호 (2016. 7)

오픈뱅킹플랫폼에서 해쉬체인을 이용한 안전한 액세스토큰 모델

정진교O, 김용민*
O전남대학교 정보보호협동과정
*전남대학교 전자상거래학과
e-mail: jinkyo@gmail.comO, ymkim@chonnam.ac.kr*

Secure Access Token Model of Open Banking Platform using Hash Chain

Jin-Kyo JungO, Yong-Min Kim*
OInterdisciplinary program of Information Security, Chonnam National University
*Dept. of Electronics Commerce, Chonnam National University

● 요 약 ●

본 논문에서는 오픈 뱅킹 플랫폼의 안전한 권한 부여를 위한 OAuth 인증 과정의 권한코드 획득 단계와 액세스토큰 사용 단계에서의 보안 취약점을 분석하여 위협 모델을 정의하고 위협에 대응하는 방법을 제안한다. 제안 하는 방법은 크게 3단계로 구분한다. 1단계로 핀테크 앱을 등록 한 후, 2단계로 사용자가 권한을 핀테크 앱에 제공하기 위하여 사용자와 핀테크 앱의 신원을 안전하게 확인하고, 액세스토큰을 준비하는 단계, 3단계로 액세스토큰 탈취에 의한 재사용 공격에 대한 안전한 액세스토큰의 사용으로 구성되어 있다. 본 논문에서는 기존 연구와의 비교를 통하여 OAuth 인증 플로우의 보안 위협에 대해 기존 권한승인 획득 단계와 액세스토큰 사용 단계를 포괄하는 넓은 위협에 대응을 할 수 있다.

키워드: 오픈뱅킹플랫폼(open banking platform), 오쓰(OAuth), 액세스토큰(Access Token), 해쉬체인(Hash Chain)

I. Introduction

핀테크는 금융 서비스에 IT 혁신 기술을 적용하여 이용자 편의 서비스를 제공하고, 금융 업무 처리의 효율화를 꾀하는 것이다. 핀테크 산업이 발전하기 위해서는 금융회사가 보유하고 있는 금융 정보에 대하여 일반 IT 기업도 접근할 수 있는 경로를 만들어 주어야 한다. 영국의 경우 2014년부터 영국 재무부를 중심으로 금융 서비스를 표준 오픈 API로 전환하는 오픈 뱅킹 플랫폼 정책을 수립하여 추진 중에 있으며[1] 국내는 농협과 IBK 등 일부 은행에서 금융 서비스 시스템의 오픈 플랫폼 구축을 진행하고 있다[2].

본 논문에서는 오픈 뱅킹 플랫폼에서 안전한 금융 거래에 필요한 안전한 권한 부여 방식을 제안하고자 한다. 이를 위하여 오픈 뱅킹 플랫폼의 권한 인증 프로토콜인 OAuth의 보안 위협에 대하여 분석하고 이를 극복할 수 있는 클라이언트 인증 방법과 일회용 액세스토큰 사용방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 오픈 뱅킹 플랫폼의 개요와 구성에 대해 살펴보고, 오픈 뱅킹 플랫폼의 인증 모델로 권고되고 있는 OAuth 2.0의 개요와 보안 위협에 대하여 분석한다. 3장에서는 2장에서 분석한 보안 취약성을 극복할 수 있는 방식을 제안한다. 마지막으로 5장에서는 결론 및 향후 연구에 대하여 제시하고 마무리한다.

II. Preliminaries

1. 오픈 뱅킹 플랫폼 개요

오픈 뱅킹이란 API 플랫폼, 앱 스토어 그리고 앱을 통하여 사용자에게 즉각적인 서비스를 제공하는 것이다[3]. 금융회사와 핀테크 기업이 서비스 개발 및 운영 과정에서 소통할 수 있는 통로로 사용되어 금융 서비스를 비금융 개발사에서도 손쉽게 구현할 수 있게 된다.

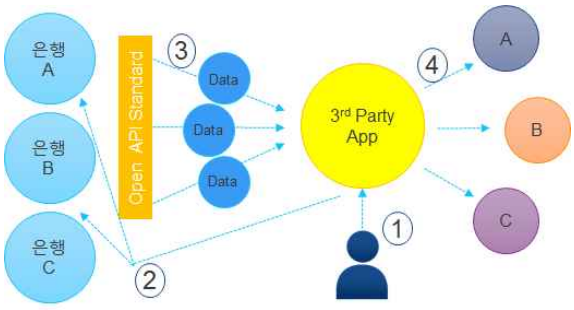

Fig. 1. 개방형 API 표준 개요

Fig.1은 사용자가 핀테크 앱을 이용하여 안전한 방법으로 각 은행 계정의 데이터를 이용하여 다양한 서비스를 이용하는 개념을 보인 것이다[4].

Gartner에 의하면 전세계 상위 50개 글로벌 은행 중 75%가 API 플랫폼을 출시할 것이고, 25%의 은행은 고객 서비스용 앱스토어를 출시할 것으로 예상하고 있다[3].

2. OAuth를 이용한 권한 부여

OAuth(Open standard for Authorization)는 HTTP 서비스 환경에서 자원 소유자가 허락한 제3자가 서버에 보관된 소유자의 자원(Owner's Resource)중 일부에 대한 접근권한을 부여하기 위하여 제3자에게 사용자의 정보(e.g. username, password)를 알려주지 않고, 권한을 부여하기 위한 프로토콜이다[6]. OAuth 2.0은 클라이언트-서버 웹 어플리케이션, 모바일 어플리케이션 등에서 필요한 권한 부여(delegated authorization)의 방법에 대하여 정의하고 있고, 특히 오픈 뱅킹 플랫폼과 같은 핀테크 영역에서도 기본 인증 프로토콜로 안내 되고 있다.

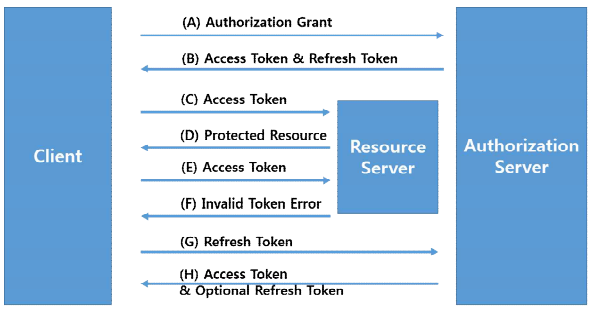

Fig. 2. OAuth 2.0 프로토콜 흐름

OAuth에서 액세스토큰을 도입하여 권한위임을 함으로 얻어지는 이익은 다음과 같다. 첫째, 컨슈머가 아이디/패스워드를 가지지 않고 API를 사용할 수 있는 방법을 제공한다. 둘째, 필요한 API에만 제한적으로 접근할 수 있도록 권한 제어가 가능하다. 셋째, 사용자가 서비스 프로바이더의 관리 페이지에서 권한 취소가 가능하다. 넷째, 패스워드 변경 시에도 액세스토큰은 계속 유효하다.

Fig.2는 OAuth2.0 프로토콜의 일반적인 동작 절차를 보여주고 있다[5].

3. OAuth의 보안 취약점

1) 권한코드 보안 위협

자원소유자가 신원을 확인하고 권한 서버로부터 권한 코드를 부여받아 클라이언트에 전달하는 구간에서의 공격자에 의한 보안위협이 존재한다.

가) 재사용 공격(Replay Attack)

OAuth 2.0 프로토콜의 권한코드는 리소스 오너가 클라이언트에게 자신의 자원에 대한 접근을 허용하는 것을 의미한다. 공격자는 리소스 오너와 클라이언트간의 권한코드 흐름을 캡처한 후 탈취한 권한코드를 재사용하여 정상적인 클라이언트에 접근하여 피해자의 자원에 불법적으로 접근할 수 있게 된다[6].

나) 위장공격

공격자는 자원사용자가 권한코드를 클라이언트에 리다이렉트 할 때 해당 권한 코드를 확보 한 후 정상 클라이언트에 권한코드가 전달되는 것을 차단한다. 이를 통하여 권한코드 사용 횟수를 1회로 한정할 경우에도 탈취한 코드를 이용하여 자원 소유자의 자원에 접근할 수 있는 액세스토큰을 발급 받는다[6].

다) 코버트 리다이렉트(Covert Redirect) 공격

2014년 발견 당시 페이스북, 구글 등 주요 오픈 API 제공사 대부분이 노출되었던 위협으로, 공격자가 클라이언트와 권한 서버간의 권한 승인 요청을 가로채서 전달 인자 중 redirect_uri값을 공격자의 서버로 변조하여 권한 코드를 탈취하여 액세스토큰을 탈취하는 공격이다[7].

2) 액세스토큰 보안 위협

자원 접근 권한을 갖고 있는 액세스토큰이 탈취 될 경우 공격자가 자유롭게 자원에 접근할 수 있는 보안 위협이 발생한다. 실질적으로 OAuth 2.0은 API 요청시 아무런 서명이 필요 없이 사용할 수 있어 보안성이 떨어지는 Bearer토큰 타입만 사용 할 수 있다. 이 때문에 Bearer Token을 사용하여 액세스토큰 요청과 보호 자원을 요청할 때에는 TLS 통신에 의존하여 토큰의 탈취를 막고 있다[8].

2011년 5월 시라아 정부는 Https 기반의 페이스북에 대하여 MITM 공격이 있었다는 사실을 발표하였다[9]. TLS 통신도 사용자의 부주의 또는 Fig3과 같이 프록시 MITM 등과 같은 신종 공격에 의해 액세스토큰이 유출 될 위험이 있다[10].

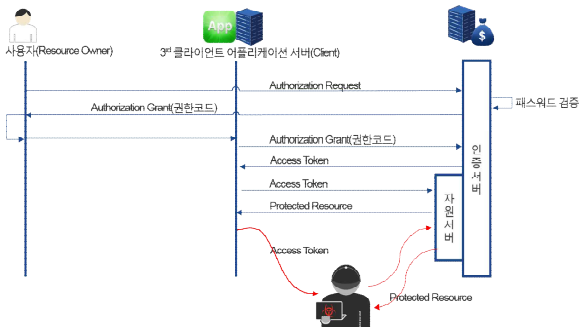

Fig. 3. 프록시 MITM 공격에 의한 액세스토큰 유출

III. The Proposed Scheme

본 연구에서는 2장에서 설명한 OAuth 2.0에서의 권한코드 확보 단계의 취약점과 액세스토큰의 유출 문제를 해결하고자 해쉬 체인 알고리즘을 이용한 액세스토큰을 제안한다. 제안하는 기법은 액세스토큰의 부정 발급 또는 탈취 되어 부정한 금융 거래에 사용되는 위협을 방어하기 위한 방법으로 핀테크 클라이언트 앱 등록, 사용자와 클라이언트 인증 및 액세스 토큰 준비, 액세스토큰의 사용과 갱신의 순서로 3단계로 구성된다.

1. 핀테크 클라이언트 앱 등록

오픈 뱅킹 플랫폼에 접근하여 각종 금융 서비스를 제공하기 원하는 핀테크 앱(클라이언트)은 Fig.4와 같이 사전 등록 및 가입 절차를 진행한다. 서비스 제공자는 통상적인 OAuth 가입시 정의되는 속성외

에 향후 액세스토큰의 Seed 값으로 사용하기 위해 생성되는 OTP를 위한 OTP-MAP과 앱의 PIN코드를 생성하고, 인증서버의 안전한 곳에 저장을 한 후, 사본을 클라이언트에게 발급한다.

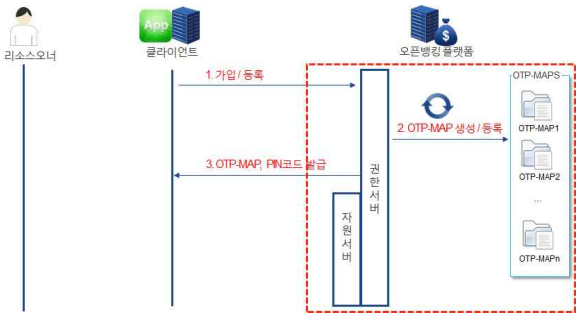

Fig. 4. 제안하는 핀테크 클라이언트 앱 등록

2. 사용자와 클라이언트 인증 및 액세스 토큰 준비

본 논문에서는 사용자뿐만 아니라, Fig.5와 같이 핀테크 앱이 적법한 클라이언트인가를 식별함으로써 재인증공격 및 위장공격에 대비할 수 있다.

(1) 자원소유자는 클라이언트로부터 권한서버로 리다이렉트 되어 ID와 비밀번호를 입력.

(2) 권한서버는 자원 소유자로부터 입력 받은 정보가 유효한지 확인 후 적법한 사용자 여부 확인.

(3) 권한서버는 클라이언트를 확인하기 위해 해당 client_id의 OTP-MAP에서 nonce를 생성.

(4) 권한서버는 client_pin과 nonce로 OTP를 생성.

(5) 권한서버는 생성된 nonce와 HMAC(OTP,nonce)를 클라이언트에게 리다이렉트.

(6) 클라이언트는 nonce를 이용하여 OTP 생성 후,

(7) 올바른 권한서버로부터 nonce가 도착했는지 HMAC(OTP,nonce)를 계산하여 검증.

(8) 클라이언트는 계산된 OTP를 Seed로 n개의 HashChain을 생성하여 API 요청에 사용할 액세스토큰 리스트를 생성.

(9) 클라이언트는 최초 액세스토큰을 권한서버에 안전하게 전달하기 위하여 $Access\,Token_n$을 OTP를 Key로 암호화하고, 클라이언트의 신원 확인용으로 HMAC(OTP,client_pin)값과 함께 권한서버에 전달.

(10) 권한서버는 HMAC(OTP,client_pin)을 검증하여 올바른 핀테크 앱으로부터 검증.

(11) 권한서버는 자원서버에 $Access\,Token_n$을 전달.

(12) 권한서버는 클라이언트에게 권한승인코드를 전달하여 API 사용 준비가 되었음을 통보하고, 향후 재인증을 수행하기 위한 리프래쉬 토큰을 발행하여 클라이언트에 전달.

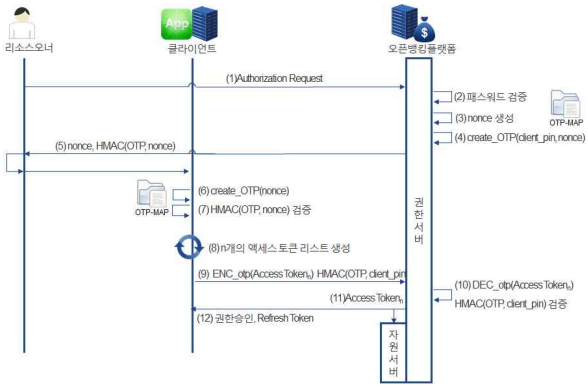

Fig. 5. 제안하는 사용자와 클라이언트 OAuth 인증

3. 액세스토큰의 사용과 갱신

금융 거래 과정 중 액세스토큰이 유출 되었을 경우에도 불법적인 금전 거래를 막을 수 있도록 매 통신시마다 액세스토큰의 값이 틀려지는 OTP의 속성을 가진 액세스토큰이 필요하다. 해쉬체인을 이용한 액세스토큰의 사용과 갱신 과정은 Fig.6과 같다.

(1) n개의 액세스토큰 리스트가 완성되면 클라이언트는 $Access\,Token_{n-1}$ 부터 사용하여 오픈 뱅킹 플랫폼의 API를 이용하여 데이터를 요청.

(2) 자원서버는 전달받은 $Access\,Token_{i-1}$의 해쉬값을 자원서버가 보유하고 있는 $Access\,Token_i$와 비교하여 적법한 요청을 검증.

(3) 자원서버는 $Access\,Token_{i-1}$를 다음 액세스토큰의 검증을 위하여 안전한 곳에 보관.

(4) 자원서버는 요청 값을 클라이언트에 전달.

(5) 클라이언트는 액세스토큰 리스트의 소진 검사.

(6) 클라이언트는 액세스토큰 리스트가 모두 소진되면 리프레쉬 토큰을 이용하여 권한서버에 새로운 액세스토큰 리스트 발급을 위한 OTP 생성을 요청.

(7) 권한서버는 리프래쉬 토큰 검증을 진행.

(8) 권한서버는 리프래쉬 토큰 검증 후 nonce 생성.

(9) 권한서버는 Client_pin과 nonce로 OTP 생성.

(10) 권한서버는 클라이언트에 OTP와 nonce의 해쉬값과 nonce를 클라이언트에 전달한다.

(11) 이후 과정은 Fig.5의 (6)~(12)과정과 동일하다.

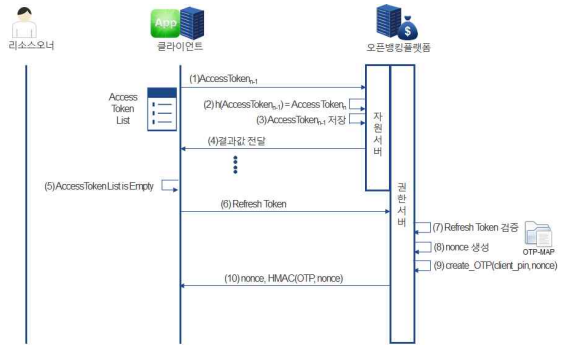

Fig. 6. 제안하는 액세스토큰의 사용과 갱신 흐름

제안하는 OAuth 프로토콜은 권한코드 획득 시점의 3개 공격 모델을 방어할 수 있다. 재사용 공격, 위장공격, 코버트 리다이렉트 공격에 대해서 중간의 통신 값을 공격자가 탈취 할 경우에도 OTP 값을 계산할 수 없으므로 최종적으로 권한 승인을 얻지 못하게 된다. 두 번째 위협요소인 액세스토큰의 경우 공격자가 클라이언트와 자원서버 간의 TLS 통신 구간을 해킹하여 $Access\ Token_i$ 을 탈취할 경우에도 이후 API 요청에는 $Access\ Token_{i-1}$이 사용되어야 하므로 공격자는 탈취한 액세스토큰을 사용하여 부정한 금융거래를 성공시킬 수 없다.

IV. Conclusions

오픈 뱅킹 플랫폼에서는 사용자 뿐만 아니라 클라이언트의 신원확인이 필요하며, 거래의 권한을 증명하는 액세스토큰을 발급 받는 과정에서의 보안성과 액세스토큰이 공격자에 유출될 경우에도 이를 이용한 공격을 차단하는 기술적 대책이 필요하다. 본 논문에서는 안전한 신원확인 과정을 위하여 클라이언트의 신원 확인을 위하여 OTP를 생성하여 클라이언트의 신원을 확인토록 했다. 액세스토큰이 공격자에 탈취될 경우에도 공격자가 이를 사용하여 거래 승인을 받지 못하도록 해쉬토큰을 이용하여 일회용 액세스토큰 기법을 제안하였다.본 논문에서 제안하는 기법을 통해 높은 보안성을 요구하는 오픈 뱅킹 플랫폼에서 편리하고 쉽게 보안 목표를 달성할 수 있을 것이다. 다만, 이번 연구에서는 TLS 공격을 통해 공격자가 통신 내용을 확보했을 때 거래 내용의 유출에 대해서는 한계가 있으므로, 거래 내용의 유출을 방지함과 동시에 성능을 보장하는 방법에 대한 추가적인 연구가 필요하다.

References

[1] Open Data Institute, “Data Sharing and Open Data for Banks- A report for HM Treasury and Cabinet Office”, Sep. 2014.

[2] Korea Financial Services Commission, “Press Release : Fintech Open Platform infra will be constructed at first in the world”, Jul 2015.

[3] Gartner, “Hype Cycle for Open Banking”, 2013.

[4] UK Gov, “Call for evidence on data sharing and open data in banking”, 2015.

[5] IETF, “RFC 6749 : The OAuth 2.0 Authorization Framework”, Oct 2012.

[6] Feng Yang, Sathiamoorthy Manoharan, “A Security Analysis of the OAuth protocol”, In Proc. Of Communications, Computer and Signal Processing PP.271-276, 2013.

[7] Dong-Jin Kim, “Introducing OAuth 2.0 and Security Consideration”, E-Finance and Financial Security of Financial Security Institute, PP 120 - 132, Mar 2016.

[8] M. Jones and D. Hardt, “OAuth 2.0: Bearer token usage,” August 2012. [Online]. http://tools.ietf.org/html/draft-ietf-oauth-v2-bearer-23)..

[9] ECKERSLEY. P. A Syrian MITM attack against Facebook.https://www.eff.org/deeplinks/2011/05/syrian-man-middle-against-facebook

[10] IETF, “RFC 6819 : OAuth 2.0 Threat Model and Security Considerations”, Jan 2013.