한국컴퓨터정보학회 하계학술대회 논문집 제20권 제2호 (2012. 7)

피싱 공격 방어를 위한 확장된 OAuth 프로토콜[1)]

문종호°*, 남윤호*, 정재욱*, 원동호*[2)]
°*성균관대학교 정보보호연구실
e-mail: {jhmoon, yhnam, jwjung, dhwon}@security.re.kr°*

An Extended OAuth Protocol for Prevent Phishing Attack

Jong-Ho Mun°, Yoon-Ho Nam*, Jae-Wook Jung*, Dong-Ho Won*
°*Information Security Group, Sungkyunkwan University

● 요 약 ●

최근 다양하고 전문적인 웹 서비스와 어플리케이션들이 사용자를 위해 제공되고 있다. 이러한 서비스들은 보통 사용자의 인증을 거친 뒤 검증된 사용자에 한해 서비스를 제공하기 때문에 사용자는 매번 서비스별로 회원가입을 해야 하는 불편함을 겪고 있다. 이러한 불편함을 해결하기 위해 제 3의 서비스가 사용자의 보호된 데이터에 접근할 수 있도록 허가하는 OAuth 프로토콜이 등장하게 되었다. 본 논문에서는 OAuth 프로토콜의 동작과정과 문제점을 분석하고 피싱 공격 방어를 위한 확장된 프로토콜을 제안한다. 본 논문에서 제안하는 프로토콜은 기존의 OAuth 프로토콜에 인증 절차를 추가한 것이다.

키워드: 오픈인증(Open Authentication), 피싱 공격(Phishing Attack), OAuth

I. 서 론

인터넷과 스마트폰의 발전과 더불어 다양한 서비스를 제공하는 수많은 웹 사이트들과 어플리케이션이 생겨나 운영되고 있다. 사용자는 자신의 취향에 맞는 서비스를 제공하는 웹 사이트 또는 어플리케이션을 선택해서 사용하는데 이러한 서비스들은 대부분 회원가입을 요구한다. 이러한 회원가입 절차 때문에 사용자를 인증하는 여러 방법들이 활발히 논의되고 있다. 대표적인 인증 시스템으로 서비스 제공자와 서드파티 어플리케이션 사이에서 사용자를 인증하고 보호된 데이터로의 접근을 허가하는 OAuth 프로토콜이 널리 쓰인다. 본 논문에서는 이러한 대표적 인증 시스템인 OAuth 프로토콜의 동작과정과 문제점을 분석하고 피싱 공격을 방어하기 위한 방법을 제시한다. 논문의 구성은 다음과 같다. 2장에서 OAuth 프로토콜의 동작과정과 OAuth 프로토콜에 대한 위협을 분석하고, 3장에서는 피싱 공격을 방어하기 위한 확장된 OAuth 프로토콜을 제안한다. 마지막으로 4장에서 결론을 맺는다.

1) "본 연구는 방송통신위원회의 방송통신융합미디어원천기술개발사업의 연구결과로 수행되었음"(KCA-2012-12-912-06-003)

2) 교신저자, dhwon@security.re.kr

II. 관련 연구

1. OAuth

1.1 OAuth Protocol

OAuth는 서드파티 어플리케이션이 HTTP 서비스에 대한 제한된 접근 권한을 얻을 수 있도록 해주는 보안 프로토콜이다. 웹 사이트 및 어플리케이션이 서비스 제공자에 가입된 사용자의 개인정보 또는 개인 서비스에 접근할 수 있도록 어플리케이션과 서비스 제공자 사이의 인증을 유도한다.

1.2 용어 정의[1]

OAuth 프로토콜 2.0의 스펙에 정의된 용어들은 다음과 같다.

표 1. 용어 정의
Table 1. Term Definition

용어	정의
Resource Owner	보호된 리소스에 대한 접근 권한을 부여하는 개체로서 서비스를 이용하는 주체인 사용자
Client	OAuth 프로토콜을 사용해서 개발된 어플리케이션
Authorization Server	Client 및 Resource Owner를 인증하고 Client에 Access Token을 발급하는 서버
Access Token	Client가 Authorization Server에 의해 인증된 Resource Owner의 개인 정보 또는 기타 서비스에

	대한 접근을 요청할 때 사용되는 키의 역할을 하는 도구
Resource Server	Client가 Resource Owner의 보호된 리소스를 요청할 때 응답하고 호스팅 하는 서버

1.3 인증 절차[2][3]

Client는 OAuth 프로토콜 2.0을 이용하기 위해서 사전에 Authorization Server로부터 아이디(client_id)와 패스워드(client_secret)을 발급받고 redirect_uri를 등록해야 한다.

Step 1. Client는 Authorization Server에 자신의 Identifier와 Redirect URI를 통한 인증을 요청한다.

Step 2. Authorization Server는 등록되어 있는 client_id와 redirect_uri의 정보가 유효한지 확인한 후 유효하다면 즉시 Resource Owner의 로그인을 유도한다. 로그인 성공과 함께 Resource Owner가 연결에 동의를 하면 Authorization code를 발급하고 다음 단계를 기다린다. 발급은 최초 등록 시 입력한 redirect_uri로 보내게 된다.

Step 3. Authorization code를 발급받은 Client는 Authorization Server에 Access Token 발급을 요청한다. Client는 이전 단계에서 발급받은 code를 이 요청에 같이 실어서 보내야 한다.

Step 4. Authorization Server는 client_secret과 redirect_uri가 유효한지 검증하고 Client가 보내온 Authorization code가 재사용되거나 expire되지 않았는지 확인한다. 기타 모든 항목 값이 일치할 경우 Access Token을 발급한다.

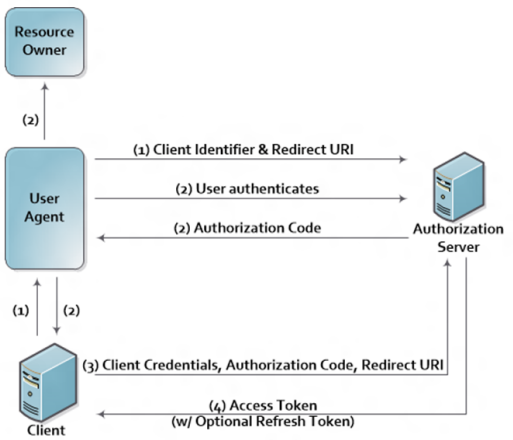

그림 1. Access Token 발급 절차
Fig. 1. Access Token Flow

Client는 Authorization Server로부터 발급받은 Access Token을 이용해서 Resource Server에 Resource Owner의 보호된 서비스 또는 개인정보를 요청한다. Access Token의 발급 절차를 포함한 OAuth 프로토콜의 기본 인증 절차는 다음 그림과 같다.

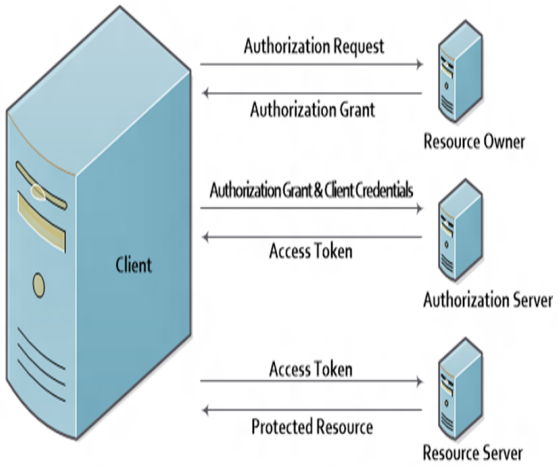

그림 2. OAuth 프로토콜 인증 절차
Fig. 2. OAuth Protocol Flow

2. 위협 분석

2.1 피싱 공격[1][4][5][6]

안티피싱워킹그룹(APWG)은 피싱 공격의 시작을 목적으로 하는 서브 도메인 호스팅이 2010년 하반기에만 약 11,700여개에 달한다는 보고서를 발표했다. 이 같은 수치는 2010년 상반기에 비해 43%나 증가한 수치이다. 또한 보안업체 시만텍은 최근 한국 인터넷 사용자를 겨냥한 피싱 공격이 증가하고 있다고 발표했다. 기존의 OAuth 프로토콜은 Client 인증 후 redirect를 통한 Resource Owner의 인증과정을 거치게 된다. 이 과정에서 공격자가 악성 사이트 또는 어플리케이션을 통해 위장된 로그인 페이지로 연결할 경우 Resource Owner의 아이디와 패스워드가 유출되고 공격자가 이를 악용해서 Resource Owner의 개인정보 및 서비스를 탈취하는 문제점이 존재한다. 현재 국내외 다수의 포털 및 SNS 사이트들이 OAuth 프로토콜을 이용하고 있기 때문에 OAuth를 가장한 피싱 공격은 충분히 발생할 수 있다.

III. 본 론

Resource Owner의 인증과 Access Token의 발급을 담당하는 Authorization Server는 자신이 보유하고 있는 사용자의 정보를 활용해서 다른 방식으로 Resource Owner를 인증할 수 있다. Authorization Server는 사전에 사용자가 인증 후 등록한 이메일 또는 휴대폰 번호를 이용한 인증을 수행함으로써 피싱 공격에 의한 Resource Owner의 패스워드 유출을 방지할 수 있다. 본 장에서는 이를 수행하기 위해 기존 OAuth 프로토콜의 인증 과정을 확장하는 방법을 제안한다.

1. 확장된 OAuth 프로토콜 제안

1.1 사전 요구사항

Resource Owner는 사전에 자신의 이메일 주소 또는 휴대전화 번호를 인증한 후 Authorization Server에 등록해야 한다. OAuth 프로토콜의 인증 과정에서 기존의 아이디/패스워드 인증 방식과 본 논문에서 제안하는 인증 방식 중 하나를 선택할 수 있다.

1.2 확장된 인증 절차

Step 1. Authorization Server는 Resource Owner의 아이디만 우선적으로 입력받는다.

Step 2. 입력받은 아이디의 존재 여부와 유효성을 검사한 후 해당 아이디를 사용하는 사용자의 개인 정보에 등록된 이메일 주소 또는 휴대폰 번호를 검색한다. 이메일 주소 또는 휴대폰 번호가 존재할 경우 사용자 인증을 위한 인증번호와 식별자를 생성한다. 인증번호는 사용자의 편의를 위해 6~8자리의 숫자로 구성한다. Resource Owner는 이메일과 휴대폰 중 자신이 사용할 인증 방식을 선택하고 인증번호 및 식별자의 발급을 요청한다.

Step 3. Authorization Server는 Resource Owner가 선택한 방식으로 인증번호 및 식별자를 발급하며 발급이 완료되면 Resource Owner에게 발급했던 식별자를 화면에 출력한다.

Step 4. Resource Owner는 자신에게 발급된 식별자와 Authorization Server가 출력한 식별자가 일치할 경우 발급받은 인증번호를 입력한다.

Step 5. Authorization Server는 Resource Owner가 입력한 값과 발급한 인증번호의 값이 일치할 경우 Client에 Authorization code를 발급하고 이후 과정은 기존의 OAuth 프로토콜과 동일하게 진행한다.

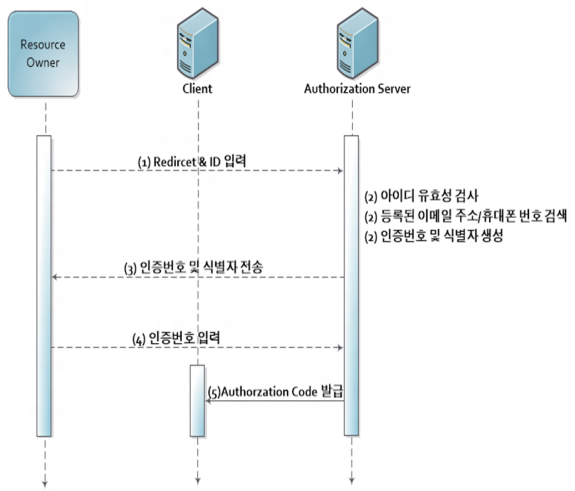

그림 3. 확장된 인증 절차
Fig. 3. Extended Authentication Flow

식별자는 현재 인증과정을 수행하고 있는 주체가 Authorization Server임을 증명하기 위한 값이다.

2. 검증

2.1 Authorization Server 인증

Resource Owner는 식별자와 인증번호가 자신의 이메일 또는 휴대전화로 전송된 경우 인증의 주체가 Authorization Server임을 확인할 수 있다. 이메일 주소 및 휴대전화 번호는 사전에 Authorization Server에 등록된 정보이기 때문에 노출되지 않는다.

2.2 식별자를 통한 대리인증 방지

공격자는 피싱 페이지를 구축한 후 Resource Owner가 입력하는 값을 이용해서 공격자 자신이 대리인증을 시도할 수 있다. 본 논문에 제안된 프로토콜에서는 Resource Owner가 발급받은 인증번호를 입력하기 직전에 인증을 수행하는 페이지가 출력한 식별자와 자신이 발급받은 식별자의 일치 여부를 확인하는 과정을 거치기 때문에 피싱 공격에 의한 공격자의 대리인증을 방지할 수 있다.

Ⅳ. 결 론

스마트폰, 스마트패드 등과 같은 휴대용 단말의 발전과 함께 다양한 서비스와 어플리케이션이 운영되고 있다. 그 중에서도 특히 SNS 서비스를 제공하는 사이트에서 OAuth 프로토콜이 광범위하게 사용된다. 사용자는 페이스북과 같은 SNS 사이트에 등록된 어플리케이션을 OAuth 인증 프로토콜을 사용한 인증 방식을 통해 이용하고 있다. 본 논문에서는 OAuth 프로토콜의 동작과정과 위협을 분석하고 피싱 공격에 의한 사용자 패스워드 유출을 방지하기 위한 확장된 프로토콜을 제안하였다. 제안된 프로토콜은 사전에 인증을 통해 등록된 이메일 또는 휴대폰을 이용한 임시 패스워드 방식을 사용함으로써 사용자의 패스워드 노출을 방지할 수 있으며 개인정보 보호에도 기여할 것으로 예상한다.

참고문헌

[1] IETF, The OAuth 2.0 Authentication Framework draft-ietf-oauth-v2-26
[2] Tistory, *http://www.tistory.com/developer/oauth.php*
[3] Facebook, *https://developers.facebook.com/docs/reference/dialogs/oauth/*
[4] APWG Phishing Report, *http://www.antiphishing.org/*
[5] Symantec, *http://www.symantec.com/ko/kr/*
[6] IETF, OAuth 2.0 Threat Model and Security Considerations draft-ietf-oauth-v2-threatmodel-02