JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 11, pp. 161-168, November 2025
https://doi.org/10.9708/jksci.2025.30.11.161

A Study on Strengthening User Authentication Using OAuth Authentication Tickets

Eun-Gyeom Jang*

*Professor, Dept. of Computer Engineering, Jangan University, Hwaseong, Korea

[Abstract]

In the paper, we propose a technical approach to enhance the security of user authentication services based on OAuth (Open Authorization), while maintaining the convenience and simplicity of login functionality. Currently, many systems rely on external OAuth authentication, which exposes them to various security threats such as ticket errors, breaches, and leakage. To address these vulnerabilities, the following security enhancement techniques are applied: first, secondary verification and authentication procedures are implemented to prevent unauthorized use of authentication tickets; second, client-side access control is reinforced to mitigate risks associated with ticket leakage; and third, real-time user authentication is provided to improve trust and system protection. Compared to conventional OAuth-only methods, the proposed approach introduces an additional OTP-based authentication step, which may slightly reduce user convenience. However, it offers significant advantages in terms of protecting both users and systems.

▸ **Key words**: OAuth, OTP, Authentication, Security Enhancement, Ticket

[요 약]

본 논문는 OAuth(Open Authorization) 기반의 사용자 인증 서비스가 제공하는 편의성과 간편한 로그인 기능을 유지하면서도, 보안성을 강화할 수 있는 기술적 방안을 제안한다. 현재 많은 시스템이 외부 OAuth 인증에 의존하고 있으며, 이로 인해 인증 티켓의 오류, 침해, 유출 등 다양한 보안 위협에 노출될 수 있다. 이를 해결하기 위해 본 논문에서는 다음과 같은 보안 강화 기술을 적용하였다. 첫째, 인증 티켓에 대한 2차 검증 및 인증 절차를 도입하여 티켓의 무단 사용을 방지하였다. 둘째, 인증 티켓 유출에 대비하여 사용자 클라이언트의 접근 제어를 강화하였다. 셋째, 실시간 사용자 인증을 통해 인증의 신뢰성과 시스템 보호 수준을 향상시켰다. 제안기술은 기존 OAuth 인증만을 사용하는 방식에 비해 OTP 기반의 추가 인증 절차가 포함되어 사용자 편의성이 다소 저하될 수 있으나, 사용자 및 시스템 보호 측면에서는 높은 보안성을 제공한다.

▸ **주제어**: OAuth, OTP, 인증, 보안강화, 티켓

• First Author: Eun-Gyeom Jang, Corresponding Author: Eun-Gyeom Jang
*Eun-Gyeom Jang (jangeg@jangan.ac.kr), Dept. of Computer Engineering, Jangan University
• Received: 2025. 10. 20, Revised: 2025. 11. 17, Accepted: 2025. 11. 19.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

현대의 IT 기술은 네트워크 접근 방식을 크게 개방형 서비스와 제한형 서비스로 구분할 수 있다. 이 중 제한형 서비스는 보안적 특성에 따라 사용자 인증 절차를 요구하며, 일반적으로 사용자 식별정보와 인증정보를 기반으로 인증이 이루어진다. 사용자 식별정보는 중복되지 않는 고유한 정보여야 하며, 인증정보는 해당 사용자만이 알고 있는 비밀스러운 정보로 구성되어야 한다[1,2].

디지털 시대를 살아가는 현대인들은 다양한 웹사이트와 앱 서비스를 이용하기 위해 수많은 플랫폼에 가입하고 있다. 쇼핑, 업무, 공공 서비스, 문화 콘텐츠 등 각기 다른 목적의 서비스에 가입하면서, 모든 계정 정보를 기억하는 것은 매우 어려운 일이다.

서비스를 제공하는 기업과 공공기관은 시스템 보호를 위해 막대한 비용을 들여 보안 시스템을 구축하고 운영하고 있다. 그럼에도 불구하고 2025년 4월 발생한 SK 유심 정보 유출 사고에서는 수천만 건의 가입자 정보가 외부로 유출되었고, KT 미니 기지국 해킹 사건에서는 불법 초소형 기지국을 통해 KT 통신망이 침해되어 무단 소액결제와 개인정보 유출 피해가 발생하였다. 이처럼 많은 비용을 투자하더라도 보안 침해는 여전히 발생하고 있다.

이러한 환경 속에서 영세하거나 소규모 기업은 사용자와 시스템을 보호하기 위한 보안 시스템을 구축하는 데 있어 비용 부담과 운영상의 어려움으로 인해 안정성을 확보하기가 쉽지 않다. 따라서 보다 효율적이고 현실적인 보안 대책 마련이 절실한 상황이다[1,2].

OAuth(Open Authorization)는 대중적으로 널리 사용되는 웹사이트를 기반으로 사용자 인증 서비스를 제공하는 방식으로, 사용자로 하여금 별도의 회원가입 절차 없이 간편하게 로그인할 수 있도록 지원한다. 이 방식은 사용자 편의성을 높이는 동시에 시스템의 부하를 줄일 수 있다는 장점을 지닌다. 국내에서는 카카오, 네이버, 구글 등이 대표적인 인증 API 제공자로 활용되고 있다.

그러나 이러한 대중 포털 사이트에 의존한 인증 방식은 사용자에게 편의성을 제공하지만, 자체 시스템 및 서비스의 보안성을 확보하는 데 있어 잠재적인 위험 요소를 내포하고 있다. 특히 인증 패킷이 침해될 경우, 사용자 정보 유출 및 시스템 접근 통제 실패와 같은 보안 위협이 발생할 수 있다[3,4].

본 논문에서는 대중화된 인증 서비스를 활용하는 환경에서 발생할 수 있는 보안 위협을 분석하고, 이를 방지하기 위한 사용자 인증 정책 및 절차를 자체 시스템에 적용함으로써 보다 안전하고 신뢰성 있는 인증 체계를 구축하고자 한다.

논문 구성으로 2장에서는 시스템 구성에 필요한 관련된 기술 요소 및 사용자 인증 기술을 제시하고, 본 연구에서 제시하는 인증 정책 및 프로세스를 3장에서 논한다. 4장에서는 제시한 인증 기술을 분석하여 제공하는 보안 서비스별로 안전성을 보인다. 마지막으로 5장에서는 연구 논문의 활용 및 연구 방향 제시로 논문을 마무리한다.

II. Authentication Technology Analysis

2.1 OAuth Authentication

OAuth는 제3자 애플리케이션이 사용자 자격 증명을 직접 취급하지 않아도 사용자의 리소스에 제한된 접근 권한을 부여받을 수 있도록 설계된 개방형 표준 프로토콜이다. 이는 사용자와 서비스 제공자 간의 신뢰를 기반으로 하며, 인증과 권한 부여를 분리함으로써 보안성과 유연성을 동시에 확보할 수 있다. OAuth 인증은 다음과 같은 구성요소를 갖는다[3,4].

Table 1. OAuth Components

Components	Descriptions
Resource Owner	The user who owns the protected resources.
Client	A third-party application that requests access to the resources.
Authorization Server	The server responsible for authenticating the user and granting
Resource Server	The server that stores and manages the protected resources.
Refresh Token	A credential that allows the client to access the resource server on behalf of the user.

- 자원 소유자(Resource Owner) : 보호된 정보를 실제로 소유하고 있는 사용자이다. 예를 들어, Google 계정을 가진 개인이 자신의 이메일이나 사진에 대한 접근 권한을 제어하는 경우이다.
- 클라이언트 애플리케이션(Client) : 클라이언트는 사용자의 데이터를 활용하고자 하는 외부 서비스나 앱으로서 이 애플리케이션은 사용자의 동의를 받아야만 해당 정보에 접근할 수 있다.
- 권한 부여 서버(Authorization Server) : 사용자의 신원을 확인하고, 클라이언트에게 접근 권한을 부여하는 서버. 로그인 절차와 권한 요청을 처리하며, 인

증이 완료되면 토큰을 발급한다.

- 리소스 서버(Resource Server) : 실제로 사용자 데이터를 저장하고 있는 서버로, 클라이언트가 유효한 토큰을 제시할 경우에만 요청을 허용한다. 이 서버는 토큰의 유효성을 검증한 후 데이터를 제공한다.
- 접근 토큰(Access Token) : 클라이언트가 사용자 대신 특정 정보에 접근할 수 있도록 허용하는 디지털 키로서 이 토큰은 일정 시간 동안만 유효하며, 접근 가능한 범위도 제한된다.
- 재발급 토큰(Refresh Token) : 접근 토큰이 만료되었을 때, 사용자가 다시 로그인하지 않아도 새로운 토큰을 받을 수 있도록 해주는 수단으로 이를 통해 사용자 경험을 끊김 없이 유지할 수 있다.

OAuth 인증은 일반적으로 Fig. 1과 같은 절차로 작동한다.

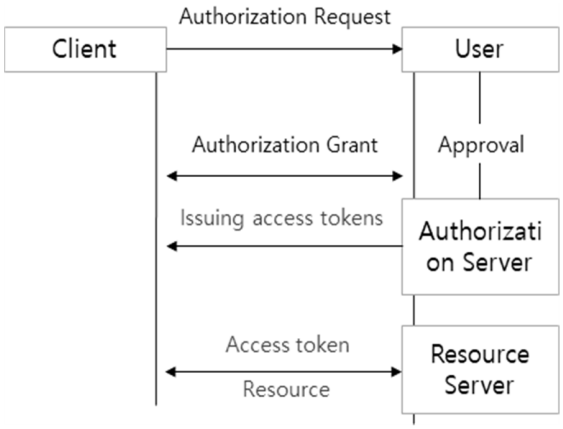


Fig. 1. Operating Procedures

1) 클라이언트는 사용자에게 권한 부여를 요청한다.
2) 사용자는 인가 서버를 통해 인증을 수행하고, 권한 부여를 승인한다.
3) 인가 서버는 클라이언트에게 접근 토큰을 발급한다.
4) 클라이언트는 해당 토큰을 이용하여 자원 서버에 접근한다.

이 과정에서 사용자의 비밀번호는 클라이언트에게 노출되지 않으며, 접근 권한은 명시적으로 제한될 수 있다[3,4,5]. OAuth의 기능을 정리하면 Table 2와 같다.

Table 2. Key Features of OAuth Authentication

Item	Descriptions
Delegated Authorization	Users can grant limited access rights to third-party applications without directly sharing their login credentials. This enables secure integration between services while protecting personal information.
Scope Definition	OAuth allows precise control over which resources the client can access. By defining scopes, users and servers can limit access to only the necessary data.
Token-Based Authentication	Authentication is handled through tokens rather than persistent sessions. Clients use these tokens to access resources on behalf of the user, improving efficiency and scalability.
Refresh Token Support	When an access token expires, a refresh token can be used to obtain a new one without requiring the user to log in again. This ensures a seamless and uninterrupted user experience.

OAuth 인증은 보안성, 편의성, 확장성, 유연성의 장점을 갖는다.

- 보안성 향상: 사용자 자격 증명이 직접 노출되지 않음
- 사용자 편의성: 소셜 로그인 등으로 빠르고 간편한 인증 가능
- 확장성: 다양한 서비스 간 연동이 용이
- 유연한 권한 관리: 세부적인 접근 권한 설정 가능

OAuth 인증은 현대의 웹 및 모바일 환경에서 필수적인 인증 및 권한 부여 메커니즘으로 자리 잡고 있다. 사용자 중심의 보안 설계와 서비스 간의 유연한 연동을 가능하게 함으로써, 다양한 분야에서 그 활용도가 증가하고 있다. 향후 OAuth는 더욱 정교한 보안 요구사항과 다양한 서비스 환경에 맞춰 진화할 것으로 기대된다.

2.2 Authentication Technology

1) Biometric authentication

생체인증(Biometrics)은 개인의 고유한 신체적 · 행동적 특성을 디지털 정보로 추출하여 활용하는 기술로서 모바일 기기, IoT, 핀테크, 원격의료 등 다양한 분야에서 개인만의 고유정보 활용 기술 영역에 적용되고 있다. 신체적 특성(지문, 얼굴, 홍채, 정맥 등)의 고유성과 정확성에 장점을 가지지만, 위조 가능성과 환경의 민감성에 의한 오류의 문제가 존재한다. 또한 행동적 특성(음성, 걸음 걸이, 서명 등)은 외부 영향에 취약하고 정확도가 낮기도 한다.

주요 활용 분야로는 금융, 보안, 의료, 공공서비스, 엔터테인먼트 등을 들 수 있다.

- 금융 : 모바일 뱅킹, ATM인증
- 보안 : PC 로그인, 출입통제
- 의료 : 환자 신원 확인, 원격진료
- 공공서비스 : 전자주민증, 선거관리
- 엔터테인먼트 : 얼굴인식 기반 사진 분류

ISO/IEC JTC1 SC37은 생체인식 기술의 국제 표준화 기구로서 생체 데이터 포맷, 시스템 구현, 시험평가 등 6개 작업반을 운영하고, 국내의 경우, TTA, KISA, ETRI 등에서 생체인식 기술의 표준화 및 융합기술 개발을 주도하고 있고 딥러닝, 블록체인과의 융합으로 생체인식의 정확성과 보안성 향상을 위한 연구가 활발하게 진행되고 있다[6,7]. 그러나 생체인식 기술은 생체정보의 보안 및 윤리적 문제, 사용자의 수용성, 국제적 표준 통합과 상호운용성 확보가 해결해야 할 과제이다.

2) OTP

OTP(One-Time Passwords)는 일회성 비밀번호로서 인증시 마다 새로운 비밀번호를 생성하여 보안성을 높이는 방식이다. 고정된 패스워드 방식의 취약점을 보완한 것으로 피싱, 키로깅, 중간자 공격 등에 강한 보안성을 제공한다. 금융기관의 이체 인증, 기업 시스템 로그인, 클라우드 서비스 접근, 블록체인 인증 등에 활용되고 있다.

시간 기반 OTP(Time-based One-Time Password)는 일정한 시간 간격마다 새로운 비밀번호를 자동으로 생성하는 방식이다. 예를 들어, 30초마다 새로운 코드가 생성되며, 서버와 클라이언트가 동일한 시간 기준을 공유해야 인증이 성공한다. 이 방식은 사용자가 별도로 조작할 필요 없이 자동으로 갱신되므로 편리하고 보안성이 높지만, 시간 동기화가 맞지 않으면 인증 오류가 발생할 수 있다는 단점이 있다.

이벤트 기반 OTP(HMAC-based One-Time Password)는 사용자가 특정 행동을 취할 때마다 새로운 비밀번호를 생성하는 방식입니다. 예를 들어, 로그인 버튼을 누르거나 특정 작업을 수행할 때마다 새로운 OTP가 생성된다. 이 방식은 시간 동기화가 필요 없고 오프라인 환경에서도 사용할 수 있어 구현이 간단하지만, OTP가 사용되지 않고 남아 있을 경우, 재사용될 가능성이 있어 보안에 취약할 수 있다. 또한 서버와 클라이언트 간의 이벤트 카운터가 불일치하면 인증이 실패할 수 있다.

최근에는 생체 기반 OTP 방식도 주목받고 있다. 이 방식은 사용자의 얼굴, 지문, 홍채 등 고유한 생체정보를 기반으로 OTP를 생성하거나 인증에 활용된다. 예를 들어, 얼굴 인식 후 해당 정보를 암호화하여 OTP로 변환하는 방식이 이에 해당한다. 생체 기반 OTP는 위변조가 어렵고 사용자가 기억하거나 입력할 필요가 없어 편리하며, 높은 보안성을 제공한다. 그러나 생체정보가 유출되면 복구가 어렵고, 인식 장비나 환경에 따라 정확도가 달라질 수 있다는 단점도 존재합니다[8].

이처럼 OTP 생성 방식은 각각의 환경과 목적에 따라 선택되며, 최근에는 여러 방식을 융합한 하이브리드 인증 시스템도 연구되고 있다. 특히 블록체인, 인공지능과의 결합을 통해 더욱 안전하고 효율적인 인증 기술로 발전하고 있는 추세이다.

3) Other authentication technologies

매직 링크(Magic Links) 인증은 사용자가 비밀번호 없이 로그인할 수 있도록 하는 방식으로, 이메일 주소를 입력하면 해당 이메일로 일회용 로그인 링크가 전송되고, 사용자가 그 링크를 클릭하면 자동으로 인증한다. 이 방식은 비밀번호를 기억하거나 입력할 필요가 없어 사용자 편의성이 높고, 피싱이나 키로깅 같은 공격에도 강한 보안성을 제공한다. 일반적으로 링크는 짧은 시간 동안만 유효하며, 한 번 사용하면 무효화 된다.

다만 매직 링크 인증은 이메일 서비스에 의존하기 때문에 이메일이 해킹되거나 링크가 유출될 경우 보안 위협이 발생할 수 있다. 또한 다중 인증(MFA)을 완전히 대체하기 어렵기 때문에, 보조 인증 수단으로 함께 사용하는 것이 권장되고 최근에는 인증 플랫폼들이 매직 링크 기능을 API 형태로 제공하며, 사용자 경험을 개선하기 위한 다양한 고급 기능도 함께 개발되고 있다.

보안키 및 하드웨어 토큰 인증은 물리적인 장치를 이용해 사용자의 신원을 확인하는 강력한 인증 방식으로, 일반적으로 USB, NFC 또는 Bluetooth 기반의 키 형태로 제공된다. 사용자는 로그인 시 해당 장치를 컴퓨터나 모바일 기기에 연결하거나 터치함으로써 인증을 완료하며, 이 과정에서 장치 내부의 암호화된 키와 서버 간의 상호 인증이 이루어진다. 대표적인 예로, FIDO2 기반의 보안키(YubiKey, Google Titan 등)가 있으며, 피싱 공격이나 중간자 공격에 매우 강한 저항력을 갖고 있어 금융, 기업 보안, 개발자 플랫폼 등에서 널리 사용된다. 이 방식은 비밀번호를 대체하거나 다중 인증(MFA)의 일부로 활용되며, 사용자 경험을 해치지 않으면서도 높은 수준의 보안성을 제공한다[9,10].

III. Enhanced User Authentication Model

3.1 System Architecture

제안 시스템은 사용자 영역의 에이전트와 서비스 제공 서버, 사용자 OAuth 인증을 제공하는 인증 서버로 크게 3개의 영역으로 나눌 수 있다. Fig. 2와 같다.

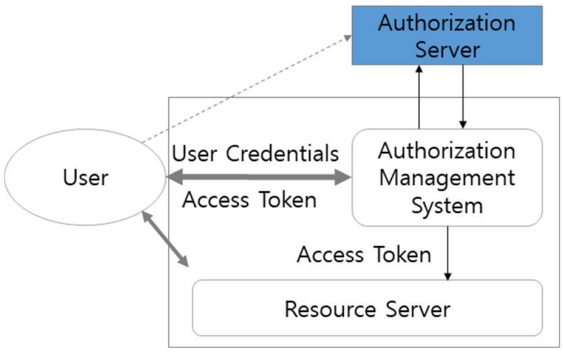


Fig. 2. Proposed System Architecture

사용자는 리소스 서버에 접근하기 위해 인증관리 시스템에 접근하여 사용자 인증을 요청한다. 인증 요청 사용자의 정보는 인증 서버(OAuth 인증)에 접속하여 사용자를 확인하고 확인 결과를 인증관리 시스템에 전송하여 1차 인증을 확인하고 2차 인증을 위한 보안 패킷을 생성하여 사용자에게 전송한다. 또한 인증된 사용자의 인증정보를 리소스 서버에 접근할 수 있도록 사용자 인증 정보를 리소스 서버에 전송한다.

3.2 System Flow Procedure

Fig.3.은 인증 시스템의 프로세스 흐름을 나타낸다.

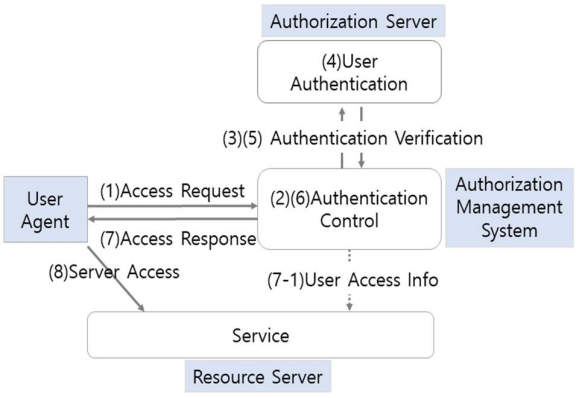


Fig. 3. System Authentication Procedure

사용자 인증 과정은 다음과 같다.

(1) Access request : 사용자는 인증을 요청한다.
(2-3) Authentication Control : 권한 관리 시스템은 사용자 인증 매체 선택(2)에 따라 인증 요청 패킷을 인증서버에 전송한다(3).
(4) User Authentication : 인증 서버는 등록되어 있는 사용자를 검색하여 인증 절차를 진행한다.
(5) Authentication Verification : 인증 결과를 권한 관리 시스템에 전송한다.
(6) Authentication Control : 권한 관리 시스템은 사용자 리소스 서비스를 위한 인증 패킷을 생성한다.
(7-7.1) Authetication Info : 권한 관리 시스템은 인증 사용자 정보를 리소스 서버에 전송한다(7-1). 사용자에게는 인증 티켓을 전송한다(7).
(8) Service Access : 사용자는 인증티켓으로 리소스 서버에 접속하여 서비스를 이용한다.

3.3 Authentication Ticket

인증 권한 관리 시스템은 사용자 인증을 위해 OAuth 서버에 접근하여 등록된 사용자 임을 확인한다.

① User → Authorization Management System → Authorization Server

$$User\ Info : \{User_{ID} \,||\, User_{PW}\}$$

OAuth는 사용자를 인증하고 권한 관리 시스템에 티켓을 발급한다.

② Create an OAuth Ticket

$OAuth_{ticket}$:
{Response_type||Client_id||Redirect_URI||Scope||State||Code||Grant_Type||Client_Secret||Access_Token||Refresh_token}

- Response_type : Type of response expected
- Client_id : Unique identifier for the client application
- Redirect_URI : URI to redirect the user after authorization
- Scope : Permissions the application is requesting
- State : Random string to prevent CSRF attacks
- Code : Authorization code returned by the server
- Grant_Type : Type of grant being used
- Client_Secret : Secret key known only to the application and authorization server
- Access_Token : Token used to access protected resources
- Refresh_token : Token used to obtain a new access token

인증 권한 관리 시스템은 사용자 맞춤형 티켓을 생성한다.

③ Generate a user authentication ticket

$Authen_{Ticket}$:
$\{User_{ID}||Token\_Issuer||ValidityPeriod_{Time}||Issue\_Date||Access_{Mac}||Usage_{List}\}$

- $User_{ID}$: User ID for anthentication request
- Token_Issuer : Ticket issuer(OAuth authentication Server ID)
- $ValidityPeriod_{Time}$: Ticket validity period
- Issue_Date : The date the ticket was issued
- $Usage_{List}$: List of servers a accessed using the ticket
- $Access_{Mac}$: MAC address of the user's client system

인증 권한 관리 시스템은 사용자에게 OTP 코드를 생성하여 전송한다.

④ Authorization → User

OTP_{Packet} :
$\{User_{ID}||Trans_{Method}||Sent_{Time}||Packet_{Lifetime}||Authen_{Code}\}$

- $User_{ID}$: User identification code
- $Trans_{Method}$: Authentication message transmission method
- $Create_{Time}$: Packet creation time
- $Sent_{Time}$: Message sent time
- $Packet_{Lifetime}$: Packet validity period
- $Authen_{Code}$: User authentication code

OTP_{Packet}은 인증 권한 관리 시스템에서 사용 확인 및 인증 패킷 보호를 위한 패킷이다. 사용자 식별을 위한 고유 ID($User_{ID}$)와 이메일 및 SMS 등 전송 방식($Trans_{Method}$), 패킷 생성시간($Create_{Time}$)과 전송시간($Sent_{Time}$), 패킷의 유효시간($Packet_{Lifetime}$), 인증 코드($Authen_{Code}$)를 포함한다.

사용자가 OTP를 인증하면 $Authen_{Code}$를 키로 활용하여 [$Authen_{Ticket}$]을 암호화한다.

⑤ Encryption of packets

$User\_Authen_{Ticket}$:
$En[User_{ID}||Token\_Issuer||ValidityPeriod_{Time}||Issue\_Date||Access_{Mac}||Usage_{List}]$

*Encryption Key*는 "$Authen_{Code}$"이다. 이렇게 생성한 암호문($User\_Authen_{Ticket}$)은 사용자 인증 티켓으로 사용된다.

일시적 사용자 인증 OTP는 사용처인 리소스 서버에 공유되어 사용자 인증 티켓을 인증하고 티켓의 유효기간 동안 리소스를 접근할 수 있도록 한다.

IV. Proposed technology analysis

제안 기술은 사용자 및 운영자 측면에서의 보안성을 검증한다. 외부의 보안 위협으로부터 시스템 및 인증 패킷 보호로 보안 서비스 제공에 초점을 두어 분석한다.

보안 서비스 분석 영역 및 시나리오는 다음과 같다.

- OAuth Ticket 침해
- OAuth 인증 미가입자
- OTP 침해 및 유출

(1) OAuth Token Breach and Exposure

$OAuth_{Ticket}$은 권한 부여 서버(Authorization Server)에서 생성된다. 서버가 침해되어 티켓 오류 및 침해 발생 또는 네트워크상에서 패킷이 유출되거나 침해가 발생할 수 있다.

이렇게 침해된 $OAuth_{Ticket}$은 인증 권한 관리 시스템(Authorization Management Server)에서 사용자에게 OTP를 전송하고 사용자의 Mac Address을 포함한 패킷을 생성한다. 사용자와 공유하는 OTP Code를 활용한 패킷 암호화로 리소스 서버 접근 티켓일 생성되어 OAuth Ticket이 침해가 발생하더라고 공격자가 리소스를 접근할 수 있는 방법은 없다. 사용자 인증 티켓의 구성 요소는 다음과 같다.

$User_{ID}||Token\_Issuer||ValidityPeriod_{Time}||Issue\_Date||\underline{\boldsymbol{Access_{Mac}}}||Usage_{List}$

사용자의 OTP 정보와 사용 시스템의 MAC Address 정보가 일치하지 않아 인증이 되지 않는다. MAC Address는 가입의 정보로 미리 등록 처리되어 있어야 한다.

(2) Users Without OAuth Enrollment

권한 부여 서버(Authorization Server)는 외부 서버로서 대중화되어 인증되고 많은 사용자가 가입되어 있는 서버를 의미한다. OAuth에 가입되지 않은 사용자의 티켓은 인증 권한 서버에서 직접 사용자 기본 정보를 등록처리 한다. 티켓 정보는 다음과 같다.

$\{User_{ID}||Address_{MAC}||Client_{ID}||Scope||State||Code||Grant\_Type||Access\_Token||Refresh\_token\}$

OAuthTicket와의 차이점은 사용자 식별 코드와 사용시스템의 MAC Address 이다. 이렇게 생성된 티켓은 3장의 ③과 같이 처리되어 운영된다.

(3) OTP Breach and Exposur

OTP는 사용자와 서버간의 일시적/실시간 인증 서비스를 제공한다. 본 연구에서는 티켓 접근키로 활용된다. OTP 코드는 $Authen_{Ticket}$을 보호하기 위한 암호키로 사용된다. 인증 권한 관리 서버와 리소스 서버와 공유되는 키로서 리소스 접근과 티켓 관리에 활용된다. OTP Code가 유출되면 일반적으로 접근 서버의 ID나 Password가 유출되었을 때, 시스템 침해가 발생할 수 있다. 하지만, 제안 시스템은 $Authen_{Ticket}$를 암호화하는데 사용하는 키로 활용되어 직접적인 서버 접근을 허용하지 않는다. 또한 $Authen_{Ticket}$를 암호화하여 권한 관리 시스템에 접근할지라도 접근을 시도하는 클라이언트 시스템의 MAC Address가 달라서 접근을 허용하지 않는다.

본 연구는 기존 OAuth 인증 시스템에서 발생할 수 있는 위협적인 사항에 대해 보안을 강화하기 위한 기술을 제안하였다. 외부 서버(OAuth 이증 서버)에 의존한 사용자 인증으로 리소스 접근에 대한 사용자 인증 강화 기술이다.

첫째, 사용자 입장에서 OAuth 인증 서비스가 가능한 사용자에게 기존 서비스를 운영 방법을 그대로 적용하면서 2차 인증 서비스에 사용되는 OTP인증 방식과 인증 티켓의 보호 기법으로 보안을 강화하였다. 둘째, 사용자 시스템의 물리적 고유 장치 번호를 사용하여 사용자의 클라이언트 시스템을 인증하여 사용 매체 인증 서비스를 제공하였다. 셋째, 한번의 티켓 발생으로 여러 리소스 서버를 접근할 수 있는 SSO(Single Sign-On)와 같은 서비스를 제공하고 있다. 추가적으로 사용자 인증 패킷의 오류 및 침해를 방지하기 위한 기법으로 Hash 함수를 활용할 수 있다. 사용자 인증 티켓일 무결성 검증을 위해 다음과 같이 패킷을 보호할 수 있다.

$H[User_{ID}||Token\_Issuer||ValidityPeriod_{Time}||Issue\_Date||\underline{\boldsymbol{Access_{Mac}}}||Usage_{List}]$

해싱된 코드와 사용자 인증 티켓을 결합하여 사용자 인증과 티켓의 무결성 서비스를 제공할 수 있다.

V. Conclusion

본 논문은 OAuth 인증 서비스를 기반으로 한 편리하고 간편한 사용자 인증 및 로그인 방식의 보안성 강화를 위한 기술적 방안을 제안한다. 기존 시스템 구조는 외부 OAuth 인증에 의존하고 있으며, 이로 인해 발생할 수 있는 보안 위협에 대응하기 위해 사용자와 서비스 관리 시스템 간의 보안 서비스를 강화하였다.

이를 위해 다음과 같은 기술적 접근을 적용하였다. 첫째, OAuth 인증 티켓의 오류 및 침해 가능성에 대비하여 2차 검증 및 인증 기술을 도입하였다. 둘째, 인증 티켓 유출에 대응하기 위해 사용자 클라이언트의 접근 제어를 강화하였다. 셋째, 실시간 사용자 인증을 위한 보안 서비스를 제공함으로써 인증의 신뢰성을 높였다.

본 연구에서 제안한 기술은 기존의 OAuth 인증만을 사용하는 사용자에게는 추가적인 OTP 인증 절차가 요구되어 다소 불편함이 있을 수 있으나, 사용자 및 시스템 보호 측면에서는 높은 보안적 이점을 제공한다. 또한 인증 티켓의 유효성을 기반으로 다양한 리소스에 접근할 수 있어 반복적인 인증 절차를 줄이는 편의성도 확보하였다. 특히 티켓이 유출되었을 경우, 사용자의 로컬 시스템 정보를 인증 데이터로 활용함으로써 보안성을 더욱 강화하였다.

ACKNOWLEDGEMENT

The Work was supported by Jangan University Research Grant in 2025.

REFERENCES

[1] Yeong Su Park, & Byoung Yup Lee, User Authentication Mechanism based on Authentication Information using One-time Sessions. JOURNAL OF THE KOREA CONTENTS ASSOCIATION, 19(7), 421-426. 2019. DOI: 10.5392/JKCA. 2019.19.07.421.

[2] Jin-Woo Lee, Seon-Joo Kim, & In-June Jo, Design and Implementation of User Authentication System Using USIM Information. JOURNAL OF THE KOREA CONTENTS ASSOCIATION, 17(7), 571-578. 2017. DOI:10.5392/JKCA. 2017.17.07.571.

[3] Jinouk Kim, 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 & Cryptology, 27(1), 5-14. 2017. DOI:10. 13089/JKIISC.2017.27.1.5.

[4] Seong-Tae Yoo and Soo-hyun Oh, OAuth-based User Authentication Framework for Internet of Things. Journal of Korea Academia-Industrial cooperation Society, 16(11), 8057-8063. 2015. DOI:10.5762/KAIS.2015.16.11.8057.

[5] Nakhyun Choi, Dongwoo Joe, & Sunoh Choi, Oauth 2.0 Authentication with Blockchain-based ERC-721 Tokens. The Journal of Korean Institute of Information Technology, 21(12), 179-187. 2023. DOI: 10.14801/jkiit.2023.21.12.179.

[6] Yuk, M., Kim, H., & Shim, H, User perception according to biometric-based personal authentication methods. Journal of the Korea Contents Association, 16(11), 11-19. 2016. DOI: 10.5392/ JKCA.2016.16.11.011.

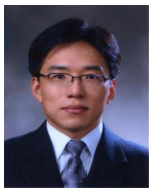
[7] Yuk, Moses, Hee-Yeon Kim, and Hye-Rin Shim, User Perception According to Biometric-Based Personal Authentication Methods. Journal of the Korea Contents Association, vol. 16, no. 11, pp. 11-19. 2016. DOI:10.5392 /JKCA.2016.16.11.011.

[8] Lee, D. G. A Study on the Current Status and Performance of OTP Utilization of Blockchain Technology. KIPS Transactions on Computer and Communication Systems, 10(2), 47-52. 2021. DOI:10.3745/KTCCS.2021.10.2.47.

[9] Eun-Gyeom Jang. A Study on the Authentication of Digital Content in Cloud Computing Environment. Journal of the Korea Society of Computer and Information , 27(11), 99- 106. 2022. DOI:10. 9708/jksci.2022.27.11.099.

[10] Lee, H., & Yun, R. J. User Experience Study on Multi -authentication Login of Online Services. Journal of the HCI Society of Korea, 18(3), 5-18. 2023. DOI:10.17210 /jhsk.2023. 09.18.3.5

Authors



Eun-Gyeom Jang received a Ph.D in Daejeon University in 2007. Hi is currently a Professor in the Department of Computer Engineering, Jangan University. It has an interest in mobile communications, system security and Computer Forensics.