Kerberos를 활용한 OAuth 기반 IoT 네트워크 인증기법

김남호O, 홍충선*
경희대학교 컴퓨터공학과
{knm1471, cshong}@khu.ac.kr

OAuth-Based Authentication with Kerberos for IoT Network

Namho KimO, ChoongSeon Hong*
Department of Computer Science and Engineering, Kyung Hee University

요 약

최근 IoT 디바이스 네트워크가 최근 연구 동향에서 IT사회의 큰 이슈가 되고 있는 만큼 보안적인 측면 또한 대두되고 있다. 따라서 본 논문에서는 최근 많은 웹 서비스가 제공하는 OAuth 인증 프로토콜과 사용자와 서버 간의 인증 프로토콜인 Kerberos를 적용한 IoT 네트워크 인증 기법을 제안하고자 한다. 기존의 OAuth 인증 프로토콜을 IoT 네트워크에 적용 시 IoT 네트워크 인증 서버를 추가한다. 또한, OAuth 프로토콜에서의 Consumer에 Kerberos를 접목시킴으로써 IoT 네트워크 인증 서버로의 접근을 제공한다. IoT 네트워크 인증 서버는 사용자에게 제공되는 보안카드 인증 프로세스를 수행함으로써 기존 보안적 취약점을 보완한다.

1. 서 론

최근 IoT 다양한 디바이스들이 등장하고 있으며, 국내외 IoT 시장 규모가 향후 10년동안 최대 5배이상 성장할 것으로 예상하고 있다.[1] 이렇듯 IoT 디바이스 네트워크 규모의 증가와 디바이스 종류가 다양해짐에 따라, 이와 관련한 많은 서비스들이 등장하고 있다.

모든 서비스에 대해서, 서비스는 이용하는 사용자의 정보를 보호할 수 있어야 하며 각 서비스들은 접근 가능한 사용자에게만 제공되어야 한다. 이는 마찬가지로 IoT 환경에서의 서비스 또한 다르지 않다.

인증된 서비스가 제공되기 위해서는, 접근 가능한 사용자 인증이 필요하지만, 제공되는 서비스 수와 디바이스의 개체 수는 큰 범위를 갖는다. 따라서, 기존의 각 서비스별 사용자 인증 방법은 많은 사용자의 불편함을 초래한다.

따라서 본 연구에서는 OAuth라는 오픈 인증 프로토콜을 적용한 IoT 네트워크 인증 기법을 제시하며, 더불어 올바른 사용자의 서비스 접근 제어를 위해 Kerberos V5를 OAuth 프로토콜에 접목시켜 보다 안전한 서비스 제공을 보장한다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0190-16-2017, IoT 기기의 물리적 속성, 관계, 역할 기반 Resilient/Fault-Tolerant 자율 네트워킹 기술 연구)

*Dr. CS Hong is the corresponding author

2. 관련 연구

2.1. OAuth 2.0 (Open Authentication 2.0)

OAuth는 최근 웹 어플리케이션 및 웹 서비스에서 한 번의 인증으로 웹 어플리케이션의 API 권한을 획득할 수 있는 오픈 인증 프로토콜이다. OAuth 2.0 프로토콜의 기본 Flow는 그림 1[2]과 같다.

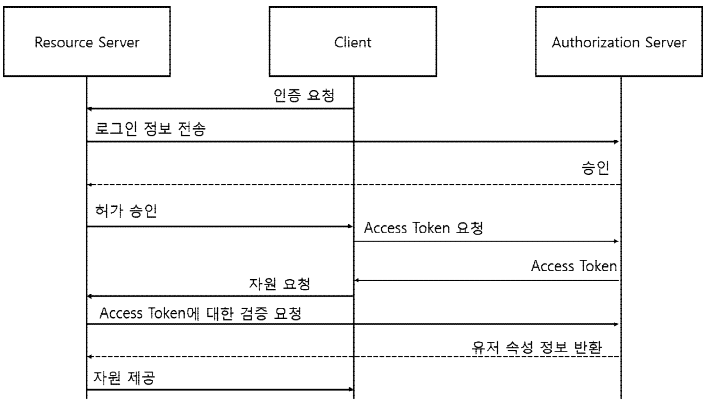

그림 1. 기본 OAuth 2.0 인증 프로토콜 flow

사용자는 Client에게 서비스 요청을 한다. 서비스 요청을 받은 Client는 사용자를 로그인 페이지로 유도한다. 사용자는 Resource Server에서 로그인을 실시하고, Resource Server는 Authorization Server에게 승인 요청을 한다. Resource Server는 Authorization Server로부터 승인이 허가되면 Authorization Server에게 Access Token을 요청한다. Authorization Server는 Access Token 요청에 대한 응답으로 Access Token을 Client에게 제공한다.

Client는 제공받은 Access Token을 사용해서 Resource Server에게 Protected Resource를 요청한다. Resource Server는 Client의 Access Token이 유효한지를 Authorization Server에 인증 요청한다. Authorization Server는 Access Token의 유효성 검사 후 인증 정보를 Resource Server로 반환한다. Resource Server는 인증 여부 확인 후 Client에게 Protected Resource를 제공한다.[3]

2.2 Kerberos V5

Kerberos는 사용자와 서버 간 인증을 할 수 있는 중앙 집중식 인증 서버를 제공한다. Kerberos는 Kerberos 서버, TGS(Ticket Granting Server), 티켓, 인증자로 구성되어있다.[4] Kerberos의 인증 메커니즘은 다음과 같다.

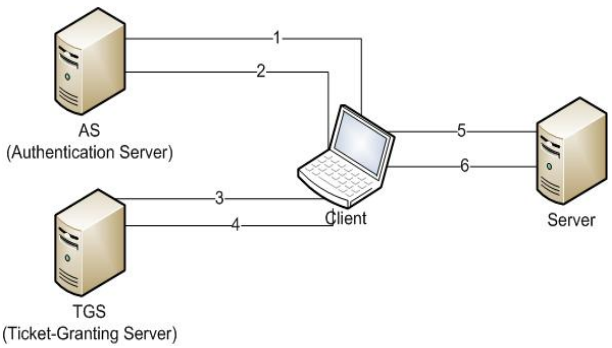

그림 2. Kerberos Local 인증 메커니즘[5]

1) 클라이언트는 ID와 TGS ID를 인증 서버에 전송한다.

2) 인증 서버는 티켓 생성 후에 클라이언트에 전송하며, 이 때 클라이언트는 사용자의 패스워드 입력을 요구한다.

3) 클라이언트는 사용자 ID, 요구하는 서비스 ID, 티켓 승인 티켓을 TGS에 전송함으로써 서비스 승인 티켓을 요구한다.

4) TGS는 들어온 티켓을 복호하하고 ID의 존재여부, 복호화의 성공 여부, 유효기간 확인, ID/네트워크 주소 점검 후 접속 승인 티켓을 발급한다.

5) 클라이언트는 서비스에 접속을 요구하고, 서비스 서버는 ID / 서비스 승인 티켓을 인증함으로써 서비스를 제공한다.

3. 제안 사항

본 연구에서는 OAuth 인증 프로토콜을 IoT 네트워크에 적용하고 OAuth 프로토콜의 취약점을 보완하기 위해 Kerberos V5와 결합을 통한 인증 기법을 제안한다.

기존의 OAuth 프로토콜의 Flow를 IoT 네트워크에 적용할 경우 클라이언트가 모든 유저의 접근을 허가할 때는 효과적이나 접근이 제한되어야 하는 서비스에 적용할 경우에는 적합하지 않다.[6] 따라서 유저의 접근을 제어할 수 있는 인증 서버를 구성하고 인증된 사용자가 서비스 접근이 가능하도록 사용자 등록의 프로세스가 필요하다. 사용자 등록 프로세스의 과정은 그림 3과 같다.

1) 사용자는 사용자 정보를 Kerberos Client에게 제공하며, Kerberos Client는 Service Server로의 접근 허가를 위한 티켓을 받는다.

2) IoT Service Server는 사용자로부터 Service Ticket을 확인 후 사용자 인증서를 제공한다.

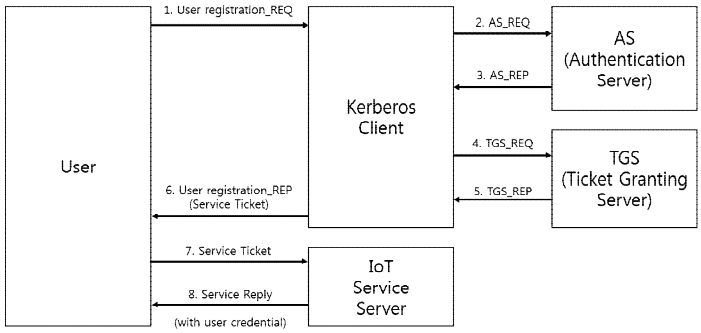

그림 3. 사용자 등록 Flow

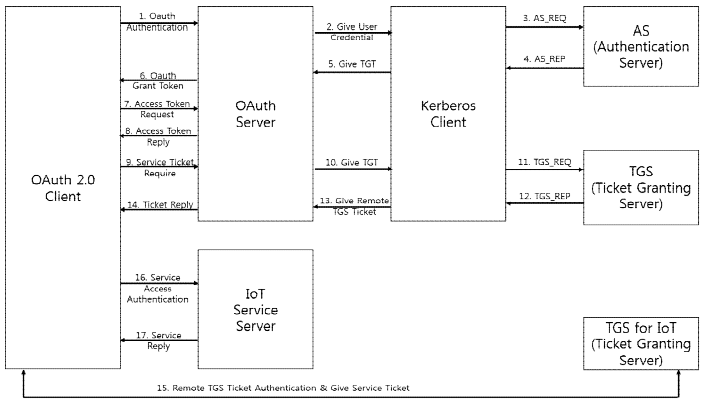

그림 4. 제안 인증 기법 Flow

제안하는 인증 기법의 Flow는 그림 4와 같다. OAuth Server와 연결된 Keberos와 IoT Service Server가 연결된 Kerberos간의 상호 인증은 사전에 진행되었다고 가정한다.

1-6) 사용자는 OAuth 2.0 Client를 통해 인증을 요청한다. OAuth Server는 사용자 정보를 Kerberos Client로 전달하면, Kerberos Client는 AS(Authentication Server)로부터 인증을 받는다. 인증이 완료 되면 TGT(Ticket Granting Ticket)를 AS로부터 제공받고 Kerberos Client는 OAuth Server로 TGT를 전달한다. TGT를 받은 OAuth Server는 OAuth Grant Token을 제공한다.

7-8) OAuth Client는 Resource Access Token을 요청한다. OAuth Server는 Access Token을 Client에게 발급한다.

9-14) Access Token을 보유한 Client는 실제 Service를 제공받기를 요청한다. Kerberos Client는 TGS에게 IoT Service Server가 인증되어있는 TGS로의 원격 TGS 티켓을 요청하고 발급받는다.

15) 원격 TGS 티켓을 IoT Service Server가 인증된 TGS로 전달하고 인증함으로써 TGS는 Service Server로의 접근 티켓을 발행한다.

16) OAuth Client가 Service Server로의 접근을 티켓을 통해 시도하면, IoT Service Server는 사전 등록되어있던 사용자 여부 확인을 위해 인증 절차를 수행한다.

17) 인증된 사용자의 경우 IoT Service Server는 Service를 제공한다.

4. 성능 평가

기존의 OAuth 프로토콜을 IoT 네트워크에 적용할 경우, Consumer의 피싱을 통한 사용자 정보 절취라는 문제점이 존재한다. 악의적인 Consumer는 사용자를 가짜 서비스 인증 페이지로 유도함으로써 ID, Password를 절취하거나 서비스 Provider로부터 받은 Access Token을 절취하여 IoT 네트워크 내부로 접근하려는 문제가 존재한다.[7][8]

그러나 본 제안 시스템에서는 악의적인 Consumer의 피싱 공격을 두 가지 방법으로 보완한다. 먼저 IoT 네트워크 접근을 제한하기 위해 Kerberos V5와 IoT 네트워크 인증 서버를 구성하는 것이다. 이와 같은 방법은 사용자 정보 및 Access Token 절취만으로 IoT 네트워크로의 접근을 제한할 수 있다.

또한, IoT 인증 서버에서의 인증 시스템은 단순히 사용자의 로그인 정보 및 Redirect URI 뿐만 아니라 사전 사용자 등록 시 제공된 사용자 인증서를 활용하여 인증한다. 이러한 방법을 통해 IoT 네트워크는 해당 네트워크 접근 가능한 사용자의 정확한 판별을 할 수 있다. 그림 5는 이와 같은 시나리오에 대한 네트워크 구성도이다.

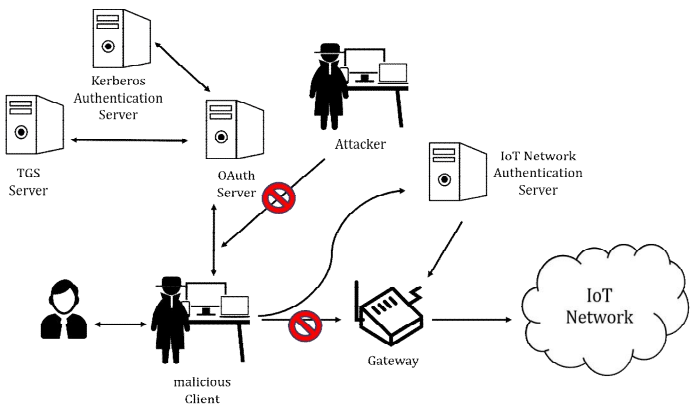

그림 5. Kerberos를 활용한 OAuth 기반 IoT 네트워크 구성도

따라서 본 논문에서 제안하고자 하는 기법은 OAuth 프로토콜을 제한된 사용자 접근이 필요한 IoT 네트워크에 적용하고자 할 때 보안적인 측면에서 효과적이라고 할 수 있다.

5. 결론 및 향후 연구

본 논문은 IoT 환경에서 인증된 네트워크를 보장하고자 OAuth와 Kerberos를 활용한 IoT 네트워크 인증 기법을 제안한다. 이러한 OAuth를 통한 접근을 통해 사용자는 각각의 서비스에 대한 개별 인증이 아닌 한 번의 인증을 통해 접근이 가능하다. 그리고 이러한 인증 Flow의 취약점은 Kerberos의 티켓을 활용한 프로토콜을 활용함으로써 보완될 수 있다. 현재 IoT 디바이스를 활용한 웹 서비스 혹은 웹 어플리케이션으로 구성이 서비스를 제공할 때, 제안하는 인증 기법을 IoT 네트워크에 적용한다면 접근 허용이 된 사용자들에게 안전한 서비스 제공이 가능해 질 것으로 예상된다.

향후 연구 방향은 제안 인증 기법의 보안 취약점 분석 및 보안카드를 이용한 2차 인증에 대한 보안 알고리즘을 강화하는 것이다.

참 고 문 헌

[1] “IoT 현황 및 주요 이슈”, 한국소프트웨어기술진흥협회

[2] D.Hardt, “The OAuth 2.0 Authorization Framework“, RFC 6749, Internet Engineering Task Force, Oct 2012

[3] Simone Cirani, Marco Picone, Pietro Gonizzi, Luca Veltri and Gianluigi Ferrari, “IoT-OAS : An O Auth-Based Authorization Service Architecture for Secure Services in IoT Scenarios“, In IEEE Sensors Journal, vol 15, no.2, February 2015.

[4] Cheol-hyun Kim, Yon-Sik Lee, “Kerberos 인증메커니즘에 관한 연구“, 정보보호학회논문지, vol 15, no.3, June 2005

[5] http://www.edgis-security.org/crypto-protocols/introduction-ttp/

[6] 최영규, 김선정, 김강석, 김기형, “OAuth기반 IoT Network 인증기법”, 한국정보과학회 학술발표논문집, 1069-1071, June 2015

[7] 저영곤, 이상래, 장기헌, 염홍열, “안전한 OAuth 인증 프로토콜을 위한 보안 문제점 연구”, 한국통신학회 종합학술 발표회 논문지 2011, 952-953, February 2011

[8] T Lodderstedt, “OAuth 2.0 Threat Model and Security Considerations“, RFC 6819, January 2013