CDN 에서 Delegated Credential 활용한 종단 간 보안 세션 수립 기법

오석원, 황은비, 권태경
서울대학교

swoh@mmlab.snu.ac.kr, ebhwang@mmlab.snu.ac.kr, tkkwon@snu.ac.kr

Establishing End-to-End Secure Session in CDNs Using Delegated Credentials

Seokwon Oh, Eunbee Hwang, Taekyoung “Ted” Kwon
Seoul National Univ.

요 약

현재 콘텐츠 전송 네트워크 (CDN)는 웹/앱의 성능을 최적화하기 위한 필수 요소이다. 다만 엣지 서버에게 오리진 서버의 TLS 인증서와 개인키의 공유가 요구된다. 이에 클라이언트와 오리진 서버는 종단 간 (End-to-End) 기밀성이 보장된 보안 세션을 수립할 수 없다. 즉 클라이언트에서 오리진 서버에게 전달되는 모든 데이터는 중간에 존재하는 엣지 서버에 의해 복호화 가능하다. 이에 본 논문에서는 Delegated Credential (DC)과 TLS 1.3 프로토콜을 활용하여 엣지 서버가 중간에 존재하는 상황에서 클라이언트와 오리진 서버가 종단 간 보안 세션을 수립하는 기법을 소개한다. 엣지 서버가 발급한 DC 로 클라이언트와 엣지 서버 사이의 일반적인 TLS 1.3 세션 수립하고 오리진 서버가 발급한 DC 로 기 수립된 TLS 1.3 세션 위에서 클라이언트와 오리진 서버 사이의 별도 키 교환/합의를 수행한다.

Ⅰ. 서 론

콘텐츠 전송 네트워크 (CDN)는 웹/앱의 성능을 최적화하기 위한 필수 요소이다. 클라이언트와 오리진 서버 사이에 존재하는 엣지 서버가 지리적으로 여러 개로 분산되어 클라이언트가 가장 가까운 것에 접근할 수 있도록 제공하는 구조다. 추가로 Distributed Denial of Service 와 같은 공격을 완화시키는 역할도 수행한다[1].

이러한 클라이언트, 엣지 서버, 오리진 서버 구조에 문제점이 존재한다. 오리진 서버와 엣지 서버 사이는 오리진 서버의 TLS 인증서와 개인키로 TLS 인증하여 HTTPS 통신이 가능하다. 하지만 그림 1 과 같이 엣지 서버가 클라이언트와 HTTPS 통신을 하기 위해서는 오리진 서버만 보유하고 있던 TLS 인증서와 개인키가 엣지 서버에게 공유되어야 한다. 이는 클라이언트와 오리진 서버의 종단 간 (End-to-End) 보안을 제공하지 못한다. 오리진 서버를 보유한 서비스 제공자의 자체 CDN 이 아닌 서드 파티 CDN 이 주로 사용되는 만큼 CDN 의 보안 위반에 대해서 오리진 서버에 영향이 있어도 간섭하지 못한다[2].

이를 해결하기 위한 다양한 기법이 연구되었다. 특히 InviCloak[3]의 경우 공개 데이터, 비공개 데이터를 분리해서 비공개 데이터만 클라이언트가 별도 암호화해서 오리진 서버에 전달한다. 별도 암호화가 되어 있기 때문에 오리진 서버만 비공개 데이터를 복호화 가능하다. 오리진 서버는 DNSSEC[4] 지원되는 환경에서 공개키를 보호해야 한다.

본 논문에서는 Delegated Credential (DC)[5]과 TLS 1.3 프로토콜을 활용하여 DNSSEC[4] 지원되지 않는 환경에서 공개 데이터와 비공개 데이터를 2 개의 다른 세션으로 전달하는 기법을 제안한다. 공개 데이터는 일반 TLS 1.3 세션을 통해 전달하고 비공개 데이터는 일반 TLS 1.3 위에서 추가로 암호화된 종단 간 보안 세션을 수립하여 전달하는 기법이다. 이를 통해 공개 데이터는 엣지 서버가 복호화 할 수 있지만 비공개 데이터는 엣지 서버가 복호화 할 수 없도록 하여 엣지 서버 및 CDN 의 정보가 공격자에 의해 탈취되어도 비공개 데이터는 보호되는 환경을 제공한다.

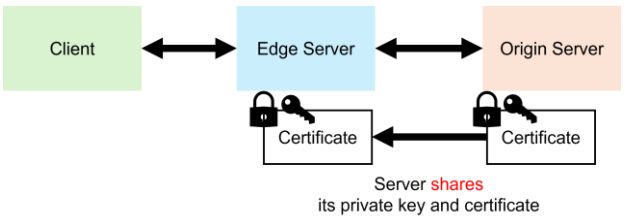

그림 1 TLS 인증서, 개인키 공유

Ⅱ. 본 론

Delegated Credential (DC)[5]이란 엣지 서버에게 제공되는 임시 TLS 인증서다. 엣지 서버가 DC 를 발급하고 오리진 서버가 이를 서명한다. 그림 2 와 같이 엣지 서버는 DC 를 활용하여 클라이언트와 TLS 세션을 수립한다. 이러한 방식은 오리진 서버의 원본 TLS 인증서의 개인키가 엣지 서버에게 공유되지 않는다는 장점을 가진다. 또한 DC 발급을 위해서 신규 인증서 발급이 필요하지 않아 비교적 짧은 주기로 DC 를 교체 가능하다. 이는 엣지 서버의 DC 정보가 공격자에 의해 탈취되어도 짧은 시간 내에 탈취된 DC 정보가 만료된다는 점을 의미한다.

우선 엣지 서버와 오리진 서버는 기존에 TLS 1.3 세션 수립되어 안전한 통신이 이루어지고 있다고 가정한다. 이는 오리진 서버의 TLS 인증서와 개인키를 기반으로 수립된다. 그러면 이제

클라이언트와 엣지 서버 간의 세션, 클라이언트와 오리진 서버 간의 세션이 필요하다.

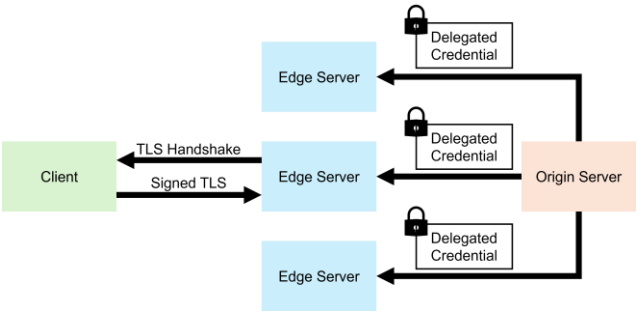

그림 2 Delegated Credential

그림 3 과 같이 두 가지 종류의 DC 가 필요하다. 엣지 서버가 발급하고 오리진 서버가 서명하는 DC (Public DC)와 오리진 서버가 발급하고 자체 서명하는 DC (Private DC)다. Public DC 의 개인키는 엣지 서버만 소유하고 Private DC 의 개인키는 오리진 서버만 소유한다. Private DC 의 경우 엣지 서버에게 공유하지 않고 자체적으로 사용하기 위해 도입했고 Self DC 생성/서명으로 그림 3 에 언급한다.

이제 클라이언트와 엣지 서버는 Public DC 기반 TLS 1.3 세션을 수립한다. 그러면 이제 클라이언트와 엣지 서버, 엣지 서버와 오리진 서버는 각 TLS 1.3 세션을 갖는다. 클라이언트가 공개 데이터를 TLS 세션키로 암호화해서 엣지 서버에게 전달하면 엣지 서버는 이를 TLS 세션키로 복호화 한다. 엣지 서버는 이를 다시 TLS 세션키로 암호화해서 오리진 서버에 전달하고 오리진 서버는 TLS 세션키로 최종 복호화 한다. 그림 3 과 같이 이러한 클라이언트, 엣지 서버, 오리진 서버 간의 통신을 일반 통신 채널이라고 부르겠다.

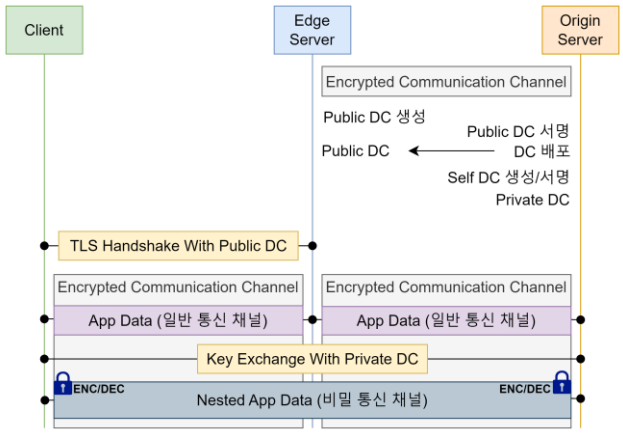

그림 3 일반/비밀 통신 채널 수립 절차

이어서 클라이언트와 오리진 서버는 일반 통신 채널 위에서 Private DC 기반 키 교환/합의를 수행한다. 이는 일반 통신 채널 위에서 디피-헬먼 키 교환과 같은 키 교환/합의 알고리즘을 수행하는 것으로, TLS App Data 를 통해 키 교환 메시지를 주고받는다. 키 교환이 끝나면 엣지 서버에게 공개되지 않는 추가 비밀키가 합의되고 일반 통신 채널 위 별도 보안 세션이 수립된다. 클라이언트는 비공개 데이터를 TLS 세션키, 비밀키로 이중 암호화해서 엣지 서버에게 전송한다. 엣지 서버는 TLS 세션키는 알지만 비밀키를 모르기 때문에 비공개 데이터를 완전히 복호화 할 수 없다. 하지만 엣지 서버가 해당 비공개 데이터를 오리진 서버에게 전달하면 오리진 서버는 이를 완전히 복호화 할 수 있다. 이를 비밀 통신 채널이라고 부르겠다.

클라이언트는 일반 통신 채널로 공개 데이터를 전달하고 비밀 통신 채널로 비공개 데이터를 전달하는 것이 가능하다. 예를 들어 사용자 로그인 정보의 경우에는 비밀 통신 채널로 데이터 전달하면 오리진 서버만 이를 복호화 후 읽을 수 있다. 공개 데이터는 엣지 서버가 복호화 가능하지만 비공개 데이터는 엣지 서버가 복호화 할 수 없다. 실제 동작 환경 구성을 위해서는 일반 통신 채널 위에서의 클라이언트와 오리진 서버의 키 교환/합의 수행과 비공개 데이터 구별 및 암/복호화에 대한 구현이 클라이언트와 오리진 서버 측에 요구된다.

3 가지 엣지 서버 공격 시나리오가 존재한다. 첫 번째, 엣지 서버가 자체적으로 Private DC 를 생성하는 경우다. 이는 오리진 서버만 유효한 DC 서명을 수행할 수 있기 때문에 클라이언트가 이를 유효한 DC 로 판단하지 않는다. 두 번째, 클라이언트와 별개로 엣지 서버가 자체적으로 오리진 서버와 키 교환/합의를 시작하는 경우다. 이는 엣지 서버와 오리진 서버 사이의 비밀키는 합의되지만 클라이언트는 포함되지 않는다. 보호해야 하는 비공개 데이터는 클라이언트가 보유하고 있기 때문에 문제되지 않는다. 세 번째, 엣지 서버가 클라이언트와 오리진 서버의 비밀 통신 채널 수립을 위한 키 교환/합의 메시지를 변조하는 경우다. 이는 Private DC 의 개인키를 오리진 서버만 보유하고 있기 때문에 디지털 서명 기법을 활용하여 식별 및 방지 가능하다.

Ⅲ. 결 론

본 논문에서는 Delegated Credential (DC)[5]과 TLS 1.3 프로토콜을 활용하여 클라이언트와 오리진 서버 사이의 일반/비밀 통신 채널을 구축하는 기법을 소개한다. 고성능 웹/앱 제공을 위해서는 CDN 활용이 필수이고 개인정보와 같은 보호가 필요한 데이터가 많아지면서 CDN 에게 높은 보안성이 요구된다. 본 논문에서 제안하는 방식은 CDN 및 엣지 서버의 변경 없이 비공개 데이터를 보호하고 기존 DC, TLS 1.3 프로토콜을 변경 없이 사용한 만큼 쉽게 적용 가능하며 타 방식과 병행하여 적용 가능하다.

ACKNOWLEDGMENT

이 연구는 정부 (과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2023-00220985)

참 고 문 헌

[1] Cloudflare. What is CDN?. https://www.cloudflare.com/learning/cdn/what-is-a-cdn/

[2] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and analysis of private key sharing in the https ecosystem. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 628– 640.

[3] Shihan Lin, Rui Xin, Aayush Goel, and Xiaowei Yang. 2022. InviCloak: An End-to-End Approach to Privacy and Performance in Web Content Distribution. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. 1947–1961.

[4] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose. 2005. RFC 4033: DNS Security Introduction and Requirements. IETF.

[5] Richard Barnes, Subodh Iyengar, Nick Sullivan, and Eric Rescorla. 2023. RFC 9345: Delegated Credentials for TLS and DTLS. IETF.