PIOLINK

마이데이터 서비스와 mTLS

mTLS Technical Guideline

1) 상호간의 신뢰 mTLS(mutual TLS)

신뢰, 사전적 의미로 "굳게 믿고 의지함"이라는 뜻을 가지고 있는 이 단어는 둘 이상의 이해 관계자가 거래, 투자, 업무 등 전반적인 경제 활동을 하는데 있어 매우 중요한 가치이다. 이런 신뢰는 웹을 통해 거래가 이뤄지고, 중요한 정보를 주고 받는 시대로 접어들면서 웹 환경에서도 그 가치가 중요하게 받아들여졌다. 또한 웹이라는 환경의 특성상 피싱, 악의적 공격, 정보 유출 등 다양한 위협이 상시 존재하기 때문에 기존의 신뢰의 개념보다 더욱 보안과 밀접한 관계를 가진 개념으로 확장되었다. 그리고 이런 개념의 확장은 많은 기업과 기관들이 신뢰를 통한 웹 환경 보안강화 연구와 기술개발을 할 수 있는 명분을 제공해주었고, 그 결과 우리는 지금 SSL/TLS (이하 TLS) 라는 암호화 통신 표준을 사용하고 있다.

암호화 통신을 위한 TLS Handshake 과정을 간단히 살펴보면 다음과 같다.

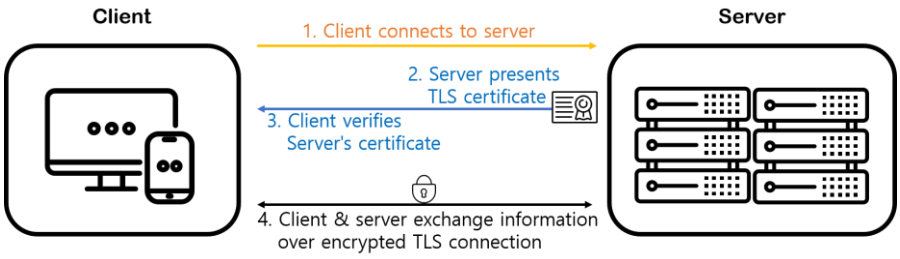
① 서버가 인증기관(CA : Certification Authority)으로부터 발급받은 x.509 기반의 인증서를 클라이언트에게 제공한다.

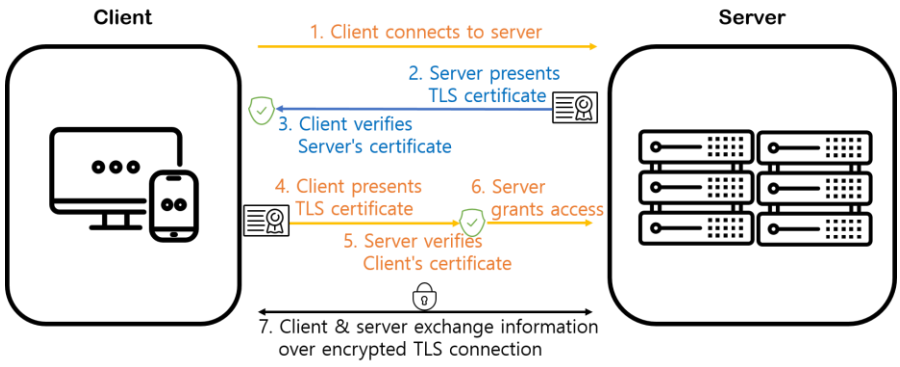
② 클라이언트는 인증서를 통해 서버의 신뢰 여부를 확인하고, 인증서 내의 공개키를 사용하여 대칭키를 암호화한 후 서버로 전달한다.

③ 암호화된 대칭키를 받은 서버는 개인키를 통해 대칭키를 복호화하여 세션을 맺는다.

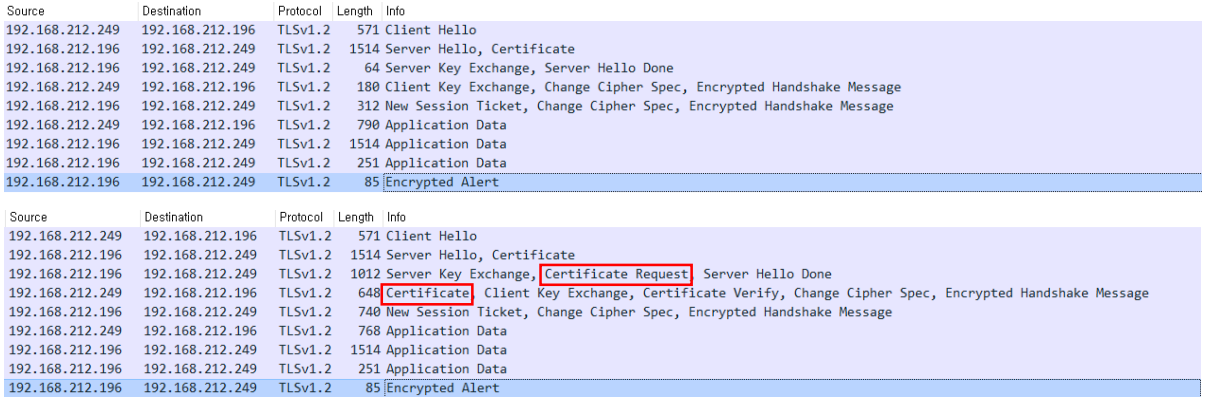
이같은 방식이 지금까지 우리가 일반적으로 사용하고 있는 웹 상에서의 암호화 통신을 위한 세션 연결 방법이었다. 하지만 최근 제로트러스트 환경이 강조되고 API를 이용하여 민감정보(개인정보, 금융정보 등)를 다루는 경우가 증가하면서 클라이언트 역시 본인이 검증된 사용자라는 걸 증명해야하는 mTLS(mutual TLS) 개념이 강조되고 있다. mTLS는 TLS Handshake 과정에서 일반적으로 생략되던 클라이언트 인증 절차를 수행하여 서버는 물론 클라이언트 역시 인증서로 본인을 증명하여 상호간 신뢰의 관계에서 통신을 하는 개념이다.

아래 [그림 1]은 기존의 TLS와 mTLS의 세션 연결 과정을 간단히 표현한 그림이고, [그림 2]는 실제 mTLS 적용 시 패킷이 어떻게 보여지는지 캡쳐한 사진이다.





[그림 1]TLS와 mTLS의 Handshake 과정



Source	Destination	Protocol	Length	Info
192.168.212.249	192.168.212.196	TLSv1.2	571	Client Hello
192.168.212.196	192.168.212.249	TLSv1.2	1514	Server Hello, Certificate
192.168.212.196	192.168.212.249	TLSv1.2	64	Server Key Exchange, Server Hello Done
192.168.212.249	192.168.212.196	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
192.168.212.196	192.168.212.249	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
192.168.212.249	192.168.212.196	TLSv1.2	790	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	1514	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	251	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	85	Encrypted Alert

Source	Destination	Protocol	Length	Info
192.168.212.249	192.168.212.196	TLSv1.2	571	Client Hello
192.168.212.196	192.168.212.249	TLSv1.2	1514	Server Hello, Certificate
192.168.212.196	192.168.212.249	TLSv1.2	1012	Server Key Exchange, Certificate Request, Server Hello Done
192.168.212.249	192.168.212.196	TLSv1.2	648	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
192.168.212.196	192.168.212.249	TLSv1.2	740	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
192.168.212.249	192.168.212.196	TLSv1.2	768	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	1514	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	251	Application Data
192.168.212.196	192.168.212.249	TLSv1.2	85	Encrypted Alert

[그림 2]TLS와 mTLS 패킷

2) 금융 마이데이터와 mTLS

지금부터 mTLS가 어디서 어떻게 사용되고 있는지 살펴보자. 우선 제로트러스트 보안 환경을 구축하고 있는 기업이라면 mTLS를 이용하여 인증 받은 클라이언트만이 시스템을 이용하도록 구성하고 있을 것이다. 다만 이런 경우는 각 기업별 환경에 맞게 구성하기 때문에 정형화된 가이드나 데이터를 확보하기 어려워 다른 케이스에 적용하기 제한적이다. 그렇다면 조금 더 범용적인 사례는 어떤 것이 있을까? mTLS 사용의 대표 사례는 바로 금융 마이데이터 서비스다. 2022년 1월 5일부터 시행된 금융 마이데이터 서비스는 개인정보와 관련된 서비스 제공 시 API를 표준으로 사용하여 암호화 통신을 하는 제도이다. API 역시 암호화 통신시에는 TLS를 이용하는데 이때 안전한 개인정보 및 데이터 전송을 위하여 [그림 3]처럼 금융보안원에서 TLS 이용 시 고려사항을 정리하였다. 그리고 그 고려사항에는 mTLS가 포함되어 있다.

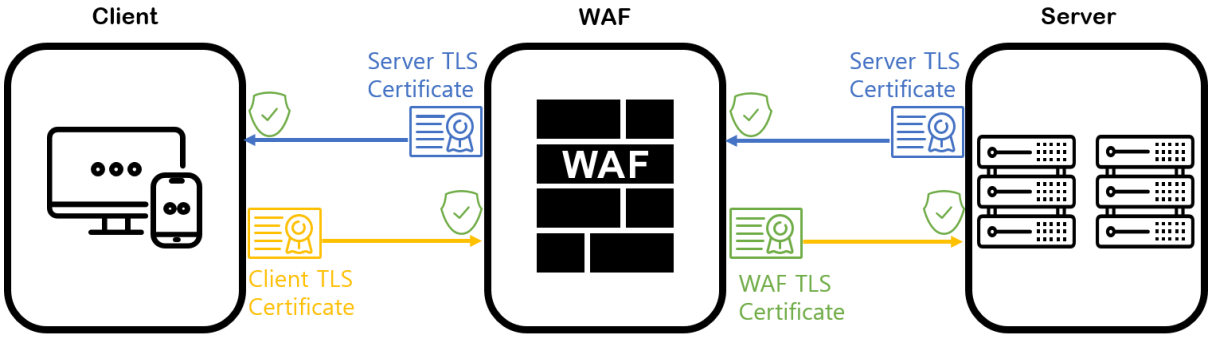
분류	설명
공신력 있는 인증서 이용	공신력 있는 인증기관에서 발급한 TLS 인증서 이용 * 신뢰성이 높고, 기관 정보를 확인할 수 있는 EV(Extended Validation) 등급 인증서 사용)
상호인증	양방향TLS(Mutual TLS)을 적용하여 상호인증 수행
최신 버전의 TLS 이용	TLS1.3 이상을 적용하며 최신 업데이트(패치)를 유지
안전한 암호 알고리즘 이용	안전한 데이터 교환을 위한 암호 알고리즘(암호화 키 교관, 메시지 인증, 데이터 암호화 등)을 이용
인증서 및 키 관리	인증서 및 데이터 전송·암호화에 이용되는 키를 안전한 방법으로 저장·관리
접근통제	TLS 설정 등에 접근 가능한 기기·사용자 등 접근통제 수행
보안에 취약한 옵션 미사용	재생공격에 취약한 0-RTT 핸드쉐이크 옵션 미사용

[그림 3] 금융 마이데이터 서비스 TLS 이용 시 고려사항 (출처 : 금융분야 마이데이터 서비스 가이드라인)

이처럼 금융 마이데이터 서비스를 위해서는 mTLS 적용이 필요하다. 하지만 지금까지 TLS 암호화통신을 사용하던 기업들 입장에서는 mTLS 적용이 막막 할 수 있기 때문에, 지금부터 어떤 부분을 고려해야 하는지 함께 살펴보자.

3) mTLS 적용하기

이 가이드라인에서 같이 살펴볼 내용은 크게 2가지이다. 우선 첫번째로 mTLS를 적용하여 세션을 맺는 과정에 대해 살펴보자. ① 세션을 맺는 방법을 알아보기 전 먼저 해야 할 것은 현재 시스템에서 TLS 통신을 담당하는 장비 및 솔루션을 확인하는 것이다. 아마 대부분의 환경은 서버 앞에서 프록시 역할을 하는 웹방화벽이 TLS 통신을 담당하고 있을 것이다. ② 해당 장비가 확인이 됐다면 다음은 TLS와 mTLS가 세션을 맺는 과정이 어떻게 다른지 이해하여야 한다. 앞서 말했듯 mTLS는 기존의 TLS Handshake 과정에 클라이언트 인증이 추가된 개념이다. 하지만 실제 클라이언트 인증과정을 추가하는 부분은 생각보다 간단하지 않다. ②-1 우선 가장 큰 차이는 세션을 맺는데 사용되는 인증서가 기존 TLS에서는 서버의 CA인증서 1개만 필요했지만 mTLS에서는 3개의 인증서가 필요하다는 것이다. 아마 대부분의 사람들이 인증서가 2개가 아니고 3개가 필요하다는 것에 의아해 할 것이다. 결론부터 말하자면 클라이언트 사이드[1]에서 세션을 맺는데 필요한 인증서 2개, 서버 사이드[2]에서 세션을 맺는데 필요한 인증서가 2개 필요하다. 이때 서버의 CA인증서는 양쪽 사이드에서 사용하기 때문에 결과적으로 3개의 인증서가 필요한 것이다. 그렇다면 지금부터 각 사이드에서 어떻게 세션이 맺어지는지 살펴보자.



[그림 4] mTLS 세션 연결시 인증서의 흐름도

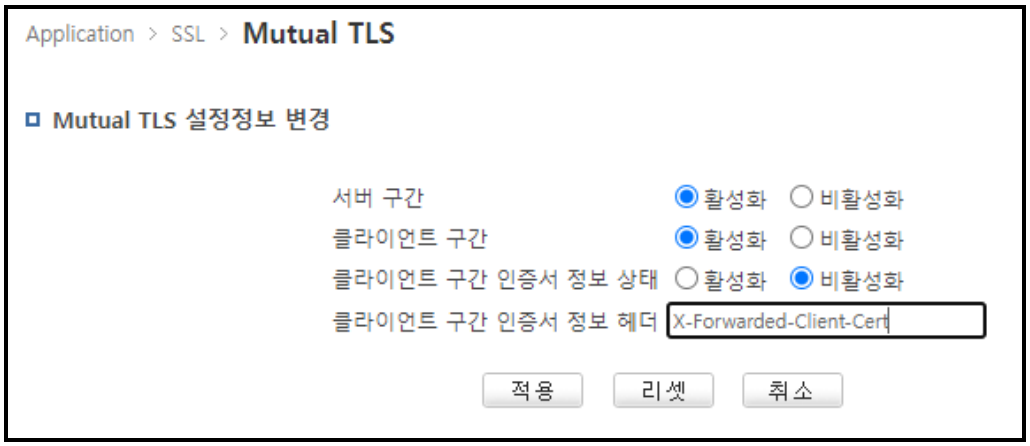
②-2 먼저 클라이언트 사이드에서의 세션 연결을 살펴보자. 기존의 TLS는 웹방화벽이 서버의 CA인증서를 클라이언트에게 보내면 클라이언트는 디바이스에 저장된 CA리스트를 통해 해당 인증서가 공인 받은 인증서인지 확인한 후 일련의 세션 연결 과정을 진행하게 된다. 하지만 mTLS는 클라이언트가 서버인증서를 확인한 후 웹방화벽이 클라이언트의 인증서를 확인하는 절차가 추가로 필요하다. 이때 중요한 것은 서버가 클라이언트의

[1] 클라이언트 사이드 : TLS/mTLS 통신시 클라이언트와 웹방화벽이 세션을 맺는 구간으로 웹방화벽이 서버의 역할을 한다.

[2] 서버 사이드 : TLS/mTLS 통신시 웹방화벽과 서버가 세션을 맺는 구간으로 웹방화벽이 클라이언트의 역할을 한다.

인증서를 확인하는 이유가 클라이언트가 서버의 인증서를 확인하는 것과 다르다는 것이다. 서버는 보통 CA에서 발급한 인증서를 사용하기 때문에 클라이언트는 디바이스에 저장된 CA리스트를 이용하여 해당 서버가 신뢰할 수 있는 서버인지를 확인한다. 하지만 서버는 클라이언트가 허용된 클라이언트인지를 검증하기 위해 미리 가지고있는 인증서 정보를 활용하여 클라이언트의 인증서를 확인하는 것이다. 즉 클라이언트는 서버가 신뢰할 수 있는 정상적인 서버인지 무결성을 확인하기 위해, 서버는 이 클라이언트가 자신에게 접속하고 정보를 제공해도 되는 클라이언트인지 검증하여 기밀성을 지키기 위해 인증서를 확인하는 것이다. ②-3 이제 다음으로 서버 사이드에서의 세션 연결을 살펴보자. 서버 사이드의 TLS 세션 연결은 클라이언트 사이드때와는 다르게 웹방화벽이 서버의 프록시 서버로써 이미 서버를 신뢰하고 있기 때문에 서버로부터 받은 CA인증서를 확인하는 과정을 별도로 수행하지 않고(제로트러스트 환경을 구축하는 기업이라면 웹방화벽에서도 서버의 CA인증서를 확인해야한다.), 대칭키 생성 및 암호화를 수행하여 세션을 맺게 된다. 그러면 여기에 mTLS를 적용하면 어떻게 달라질까? 서버사이드에 mTLS를 적용하면 클라이언트 사이드때와 마찬가지로 서버는 자신과 세션을 맺으려는 웹방화벽에게 인증서를 요구하고 웹방화벽은 세션을 맺기 위해 준비한 사설 인증서를 서버에 전달한다. 이때 역시 서버는 해당 인증서가 허용된 인증서인지 검증하기 위한 인증서 리스트 및 인증서 정보를 가지고 있어야한다. 이처럼 mTLS를 적용하여 세션을 맺기 위해서는 각 사이드에서 클라이언트를 검증하는 과정이 추가되어야 한다.

그리고 이런 연결 과정 때문에 고려해야하는 사항이 생기는데 그것이 바로 두번째로 살펴볼 내용이다. 혹시 위의 연결 과정을 보면서 서버는 클라이언트를 어떻게 확인할 수 있을까? 라는 의문을 던진 분들도 있을 것이다. 맞다. 위의 방법을 통해 세션이 연결되면 서버는 클라이언트 정보를 직접 받지 않기 때문에 서버에서는 클라이언트를 구분할 수 없다. 특히 마이데이터 서비스의 경우 서비스를 이용하는 기관과 기업 모두 상호인증을 위해 SERIALNUMBER(사업자등록번호)가 포함된 TLS 인증서 정보를 마이데이터 종합포털에 등록해야한다. 그리고 이 정보들을 바탕으로 정보 제공자는 정보를 요청한 클라이언트가 마이데이터 종합포털에 등록된 사업자가 맞는지 확인하게 된다. 결국 마이데이터 서비스를 위해서는 클라이언트 사이드에서 얻은 클라이언트 인증서 정보를 서버로 전달할 필요가 있는 것이다. 즉, 아래 [그림 5]처럼 웹방화벽에서 서버로 클라이언트 요청을 프록시할 때 클라이언트를 구분할 수 있도록 요청 헤더에 클라이언트 사이드에서 획득한 클라이언트 인증서 정보를 포함시켜 서버에게 전달해줘야 하는 것이다.



[그림 5] 파이오링크 WEBFRONT-K GUI에서 Mutual TLS 기능을 통한 헤더값 변경

이처럼 헤더에 클라이언트의 인증서 정보가 포함된다면 서버에서 클라이언트를 식별할 수 있게 되어 마이데이터 종합포털에 등록된 사업자인지 확인할 수 있게 되고, 필요시 클라이언트 정보를 추출하거나 색인 할 수도 있다.

지금까지 mTLS 적용을 위해 고려되어야 할 사항을 알아보았다. 물론 더 많은 고려사항이 있을 수 있지만, 위의 두가지 사항이 실제 mTLS를 적용하면서 많은 담당자들이 고민하고 어려워하는 대표적인 사항이라 생각된다. 이 밖에도 파이오링크에서는 마이데이터 서비스를 위해 mTLS적용을 고민하던 많은 고객들과 함께 문제를 해결하며 얻은 다양한 레퍼런스를 가지고있으니 mTLS와 관련된 고민이 있다면 언제든 파이오링크의 전문가들과 상담하여 문제를 해결하길 바란다.

(이메일: waf@piolink.com)