SSL/TLS 보안 및 취약점 동향

윤재웅*, 서화정**

*한성대학교 컴퓨터공학부

**한성대학교 IT 융합학과

e-mail : yjaewoong@naver.com

Security Technology and Vulnerability Trends of SSL / TLS

Jae-Woong Yun*

*School of Computer Engineering, Hansung University.

Hwa-Jeong Seo**

**Division of IT convergence Engineering, Hansung University.

요 약

우리가 사용하는 서비스 중에서 인터넷 연결이 필요 없는 서비스는 거의 없다고 봐도 무방할 정도로 많은 분야에서 네트워크 기술이 활용되고 있다. 이와 더불어 인터넷 연결에 활용되는 네트워크 통신 간의 해킹이나 바이러스, 그리고 개인 정보 유출도 덩달아 증가하고 있다. 따라서 이러한 보안 취약점을 이용한 공격을 방지하기 위한 네트워크 보안의 중요성이 점차 증가하고 있다. 본 논문에서는 네트워크 통신환경에서 주고받는 데이터를 보호하기 위해서 암호화를 해주는 통신 규약으로 널리 사용되고 있는 SSL (Secure Socket Layer)과 TLS (Transport Layer Security)의 변천과 구조를 확인해 보고 이에 따른 동작과 취약점 동향에 대해 확인해 보도록 한다.

I. 서론

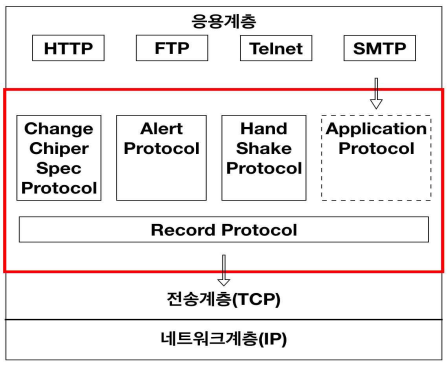
인터넷이 점차 발전에 따라 성별과 나이를 불문하고 네트워크를 사용하고 있다. 가정이나 회사에서 PC는 물론 핸드폰, 태블릿 같은 유무선 기기를 사용하여 언제 어디서나 쉽게 인터넷 서비스를 통해 네트워크를 생성할 수 있게 되었다. 인터넷은 사용자간의 정보를 자유롭게 상호 접근 및 공유를 가능하게 한다. 그러나 개방성, 공유성의 특징을 가지고 있는 인터넷은 이러한 순기능만 가지고 있는 것은 아니다. 인터넷 사용이 증가함에 따라 네트워크를 악의적인 목적으로 사용하여 해킹과 바이러스, 개인정보 유출 등의 발생 빈도도 점차 증가하고 있다. 이러한 사고들을 방어하기 위한 네트워크 보안의 중요성도 함께 높아지고 있다. 본 논문에서는 클라이언트와 서버 사이에 교환되는 데이터를 안전하게 보호하기 위하여 공개키 인증서를 통한 사용자 인증, 비밀키 암호 시스템을 통한 데이터의 기밀성을 제공해주는 SSL(Secure Socket Layer)과 TLS (Transport Layer Security)의 변천과 구조를 살펴보고, 동작 과정 및 취약점에 대한 동향에 대해 확인해 보도록 한다.

II. SSL/TLS

2.1 개요

SSL(Secure Sokets Layer)은 1994년 Netscape사에 의해서 Netscape 웹브라우저를 통한 안전한 통신을 위하여 최초로 제안되었으며, 1994년 SSL v2.0, 1996년 Internet Engineering Task Force(IETF)에서 SSL v3.0을 제안했다. 이후에도 SSL v3.0은 지속적으로 수정 및 보안 되었으며 그 당시 사실상의 웹 보안의 표준 이었다. 1999년에는 SSL v3.0을 참고로 하여 RFC 2246으로 표준화 된 것이 TLS(Transport Layer Security)이다. 1999년 TLS v1.0(RFC 2246 : SSL v3.1에 해당)을 시작으로 TLS v1.1 (RFC 4346, 2006), TLS v1.2 (RFC 5246, 2008) 그리고 최근 TLS v1.3도 등장했다. SSL/TLS는 TCP(UDP 상에서도 가능한 버전 존재 - DTLS(Datagram Transport Layer Security) RFC 6347, 2012) 위에서 Record Protocol을 통해 실직적인 보안 서비스를 제공하고, Handshake Protocol, Change Cipher Spec Protocol, Alert Protocol을 통해 SSL/TLS 동작에 관한 관리를 하게 된다.

2.2 구조



[그림 1] SSL/TLS 프로토콜 계층 구조

[그림 1]의 응용계층과 전송계층(TCP) 사이에서 SSL/TLS는 독립적인 프로토콜 계층으로 동작한다. 응용계층의 프로토콜들은 외부로 보내는 데이터를 전송계층이 아닌 SSL/TLS 계층으로 보내게 된다. SSL/TLS 계층은 받은 데이터를 암호화 하여 전송계층으로 보내서 외부 인터넷에 전송한다.

① Record Protocol

Record Protocol은 데이터의 압축을 수행하여 안전한 TCP 패킷으로 변환하고, 데이터 암호화 및 무결성을 위한 메시지 인증을 수행하는 프로토콜로, Handshake Protocol, Change Cipher Spec Protocol, Alert Protocol, 그리고 Application Protocol을 감싸는 역할을 한다.

[그림 2] Change Cipher Spec Protocol을 감싸고 있는 Record Protocol의 예시

[그림 2]의 예시를 보면 Protocol 필드에는 Cipher Spec을 나타내는 20이 들어간다. 그리고 data를 보내기 좋게 자르거나 붙이고 선택적으로 압축하여 MAC(Message Authentication Code)을 적용하고 암호화하여 이를 TCP로 전달한다.

② Change Cipher Spec Protocol

암호화 알고리즘과 보안 정책을 송수신 측간에 조율하기 위해 사용하는 프로토콜로, 프로토콜의 내용에는 단 하나의 바이트, 언제나 1이라는 값이 들어가게 된다. ([그림1] 에서 Change Cipher Spec Protocol을 감싸고 있는 Record Protocol을 참고)

③ Alert Protocol

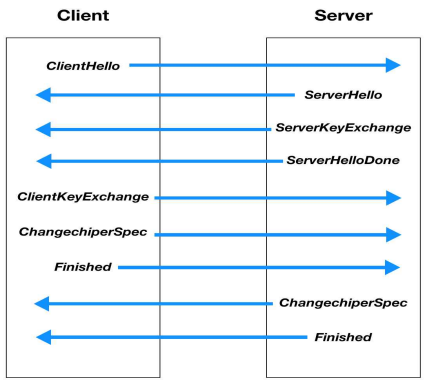
2바이트로 구성되며, 첫 번째 바이트에는 warning(주의를 해야 하는 문제, 연결 미종료) 또는 fata(매우 중대한 문제, 연결 종료)이 들어가고 두 번째 바이트에는 Handshake, Change Cipher Spec, Record Protocol 수행 중 발생하는 오류메시지가 들어가게 된다.

[그림 3] Alert Protocol을 감싸고 있는 Record Protocol 예시

④ Handshake Protocol

대부분의 메시지가 여기에 해당하며, 암호 알고리즘 결정, 키 분배, 서버 및 클라이언트 인증을 수행하기 위해 사용되는 프로토콜이다. 레코드 계층 상위에서 클라이언트는 Handshake Protocol 을 이용하여 보안 파라미터를 서버에게 요청하게 된다. 이때 서버는 클라이언트의 요청에 응답하여 통신에 필요한 파라미터들을 설정한다. Handshake에 의하여 설정된 파라미터들은 Change Cipher Spec Protocol 로 사용할 수 있도록 활성화되고, data들은 Application data Protocol 을 통하여 SSL/TLS에 의해 보호되어 전송된다. 통신과정에서 발생한 오류들은 Alert Protocol 이 처리하게 된다. Handshake Protocol 수행 과정에서 클라이언트와 서버를 상호인증하며 암호 알고리즘, 암호키, MAC 알고리즘 등의 세션 상태를 유지할 수 있는 요소들을 설정하게 된다.

2.3 동작



[그림 4] Handshake Protocol의 동작

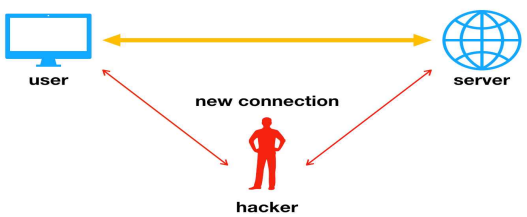
Handshake Protocol 동작 과정은 [그림 4]와 같이 진행되며 다음과 같이 동작하며 네 가지 단계로 나눌 수 있다. 첫 번째, 초기 협상 단계로 통신 시작 알림으로 클라이언트는 서버에게 Client-Hello 메시지를 전송하고 서버의 응답을 기다린다. 만약 서버가 바로 Server-Hello 메시

지를 보내지 않으면 에러가 발생하고 커넥션이 중단된다. 그리고 클라이언트가 서버에게 CipherSuite(사용 가능 암호화, 해싱 방식 등)을 보내고 서버 인증서를 요구한다. 두 번째, 인증 단계로 서버에서 공개키, 서버명, 인증기관 주소 등을 포함한 인증서를 서버가 클라이언트에게 ServerKEyExchang 메시지를 전송한다. 이때, 서버는 클라이언트가 제시한 것 중 자신이 선택한 암호화 방식 및 인증서를 보낸다. 필요시 클라이언트는 인증서를 발급한 인증기관 서버에 접속하여 서버 인증서의 유효성을 확인한고, ServerHelloDone 메시지로 초기화 협상 단계 마무리를 클라이언트에게 알린다. 세 번째, 보안 채널 형성으로 클라이언트는 보안 채널 형성에 필요한 세션키를 만들기 위해 서버의 공개키를 이용하여 임의의 수(Pre Master Key)를 암호화 시켜 서버에게 ClientKeyExchange 메시지를 전송하고 서버는 자신의 비밀키(개인키)로 이를 해독(역암호화)하게 된다. 이때 임의의 수로부터 Master Key를 유도하고 이 Master Key로부터 양측의 암호화, 복호화에 필요한 세션키를 생성한다. 지금까지 정한 암호화 파라미터를 사용 한다는 ChangeCipherSpec 메시지와 암호화 파라미터들의 협상 종료 Finished 메시지를 서버에게 보낸다. 마지막으로, 서버는 클라이언트에게 ChangeCipherSpec 메시지를 보냄으로서 지금 까지 정해진 암호화 파라미터들을 실제로 적용 한다. 그리고 암호화 파라미터들의 협상을 종료 하겠다는 Finished 메시지를 클라이언트에게 보냄으로서 상호 암호화 통신 시작된다. 즉 보안성이 확립된 SSL/TLS 터널 내에서 상호 통신이 가능해 진다[1].

III. 취약점

3.1 취약점 동향

일반적인 네트워크 환경에서 사용자가 웹 서버와 통신을 하려면 무수히 많은 라우터 혹은 스위치를 거쳐야 하는데 이 장비들은 통신의 내용을 알 수 있다. 만약 공격자가 통신장비처럼 클라이언트와 웹 서버와의 경로 상에 들어가게 되면 모든 내용을 감청 및 위변조가 가능하게 되는데 이것을 중간자 공격(Man-in-the-middle attack)이라고 부른다[2]. [그림 5]은 일반적인 네트워크 환경과 중간자 공격을 받는 환경을 나타낸 것이다.

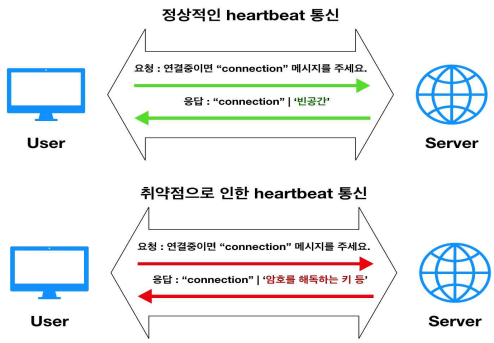


[그림 5] Man-in-the-middle

SSL/TLS는 중간자 공격에 취약하여, Handshake 단계에서 암호화 알고리즘 및 SSL/TLS 버전 등을 협상하는 과정에 공격자가 개입하여 협상 내용을 보안에 취약한 것으로 변경하는 방식(다운 그레이드)의 공격이 많이 발생한다. 다운그레이드 공격 외에도 각 버전에서 지원하는 암호화 방식이나 암호화 알고리즘의 취약성을 이용한 공격도 자주 발생하며, BEAST, FREAK, Logjam, DROWN, POODLE 취약점이 있다.

프로토콜 자체의 취약점 외에도 이를 구현한 오픈 소스 라이브러리인 OpenSSL은 정해진 규격의 네트워크 보안 프로토콜을 범용 라이브러리로 구현하기 위한 목적으로 만들어졌으며, SSL/TLS를 이용한 암호화를 구현할 수 있다. 강력한 암호화 기능을 제공하기 때문에, 보안이 중요한 대형 포털서비스, 이메일 서비스, 금융권 등에서 데이터 통신 시 사용되고 있다. 몇 년 전에 이슈가 되었던 Heartbleed[3]는 OpenSSL의 취약점(OpenSSL 1.0.1~1.0.1f, 1.0.2-beta, 1.0.2-beta1 버전)으로, 잘못된 프로토콜 구현이 원인이 되어 발생하는 취약점이다. SSL/TLS에는 상대와 연결이 계속 유지되는지 확인할 수 있도록 Heartbeat 프로토콜이 존재한다. [그림 6] Heartbeat는 어느 한 쪽이 메시지를 전송하면 상대방이 수신된 메시지를 그대로 다시 전송하는데 악의적으로 조작된 메시지를 받게 되면, 그 뒤의 메모리 내용을 상대방에게 전송하게 된다. 따라서 공격자는 Heartbeat 프로토콜이 허용하는 최대 메시지 길이인 64KB씩 서버 메모리의 데이터를 탈취할 수 있다.

해당 취약점을 이용하여 시스템 메모리에 저장되어 있는 무의미한 작은 정보들을 지속적으로 유출시키면, 이러한 무의미한 정보들이 모여 하나의 완전한 유의미한 정보가 될 수 있다. HeartBleed 명칭의 유래는 해당 취약점으로 공격할 때마다 작은 정보들이 새어 나오는 것을, 심장이 한 번씩 뛸 때마다(Heartbeat) 심장에서 피가 한 방울씩 떨어지는 치명적인 심장출혈(Heartbeat)로 비유하여 명명한 것이다. 대응방안으로는 시스템 측면, 네트워크 보안장비 측면 서비스 관리 측면에서 보안이 가능하다.



[그림 6] heartbeat 통신

최근 발생한 취약점으로는 사이퍼 스턴팅(Cipher Stunting)이 있다[4]. 공격자는 사이퍼 스턴팅 취약점을 이용해 암호화된 트래픽을 조작함으로써 탐지 장치들을 피해간다. SSL/TLS로 암호화 된 통신의 디지털 지문을 조작하는 방식이다. 최초 Handshake 요청인 클라이언트 헬로(Client Hello) 패킷(TLS버전, 세션 ID, 암호화 관련 옵션, 확장자, 압축 방법과 관련된 정보)의 변종 수가 갑자기 폭발적으로 늘어나는 현상이 일어난다. 연결이 성립되는 동안 패킷 정보를 확인하고 매칭시킴으로써 증가한 트래픽 뒤에 숨어 공격자와 정상 사용자를 구분하지 못하게 한다. 암호화 된 트래픽은 내용을 들여다볼 수 없기 때문에 공격자들이 이 트래픽 안에 자신들의 공격툴들을 숨겨서 사용했다. 이러한 경로를 차단하기 위해 암호화된 트래픽의 내용이 아니라 정보를 확인하고 매칭시켜 공격자와 정상 사용자를 구분하는 디지털 지문을 조사해 트래픽을 검사하는 방법을 사용해왔다. 하지만 이에 공격자들은 디지털 지문을 조작하는 사이퍼 스턴팅 취약점을 이용했다. 공격자들의 최종 목표는 단일 컴퓨터나 단일 네트워크로부터 발생하는 트래픽을 수만 개 내지는 수천 만 개의 사용자 장비로 보이도록 둔갑시키는 것이다. 공격자들은 사이퍼 패킷의 각종 정보들을 무작위로 가져다 쓰면서 디지털 지문의 양을 늘리고 내용을 바꿨다. 이로 인해 클라인언트 헬로 패킷의 디지털 지문 수가 기하급수적으로 증가하기 시작 한다.

3.2 TLS 동향

TLS v1.3(RFC 8446, 2018)은 이전 버전인 TLS v1.2부터 유지되어 온 구조적인 취약점을 없애기 위해 새롭게 설계되었다[5]. 이에 따라 강화된 보안과 빨라진 성능을 제공한다. 주요 특징으로는 크게 세 가지가 존재한다. 첫 번째, 보안성 강화로 Handshake 단계에서 인증서를 암호화하고, 무결성을 검증함으로써 중간자 공격을 통해 협상 내용을 취약하게 변경하는 다운 그레이드 공격 방어가 가능하다. 두 번째, TLS v1.2 이하 버전 Handshake 과정에서 2-RRT(Round Trip Time)를 거쳐야 했으나, 이 과정을 단순화시켜 1-RRT로 감소시켜 성능을 향상시켰다. 세 번째, 프라이버시 강화로 TLS의 확장인 SNI(Server Name Indication, 2003) 정보를 암호화 하는 ESNI(Encrypted SNI) 드래프트가 나와 있는 상태이다[6]. ESNI를 적용하면 접속지 정보가 암호화되므로, 국가 검열 등의 이슈에서 자유로워 질 수 있으며 사용자에게는 큰 프라이버시를 제공하게 된다. 하지만 SSL/TLS의 이전 버전에 결함이 발견되어 새로운 버전이 발표되더라도, 호환성 문제나 관리의 문제 등으로 인해 이전 버전을 허용하는 경우가 많다. 실제로 TLS v1.1 이하의 경우 보안성 이슈로 지원을 중지하고 있다. PCI(Payment Card Industry) 협의회에서는 2018년 6월까지 사용을 중지하도록 했고, IETF에서도 TLS v1.0과 v1.1 사용 금지에 대한 드래프트를 내 놓았다. Chrome. Firefox, Edge, Safari, Explorer 등 주요 웹브라우저들도 2020년에 TLS v1.0과 v1.1에 대한 지원을 중지한다고 발표했다.

IV. 결론

본 논문에서는 SSL/TLS의 변천과 구조 그리고 동작 과정을 알아보았으며, 취약점에 대해서도 알아보았다. SSL/TLS로 암호화된 트래픽이 발전함에 따라 사용자에게는 안전을 제공하지만 공격자에게는 무기를 숨길 수 있는 안전한 통로가 되고 있다. 사용자는 보안성을 고려하여 가능한 높은 버전의 프로토콜을 유지해야 하며 보안을 위해 트래픽에 명확한 가시성이 필요하다.

[참고문헌]

[1] Sung-Min Kim, Jun-Sang Park, Sung-Ho Yoon, Jong-Hyun Kim, Sun-Oh Choi, Myung-Sup Kim, "Service Identification Method for Encrypted Traffic Based on SSL/TSL" The Journal of Korean Institute of Communications and Information Sciences 40(11), 2015.11, 2160-2168(9 pages)

[2] Seong-Min Cho, Hoon Lee, "A Countermeasure against the Abatement Attack to the Security Server", Journal of the Korea Institute of Information and Communication Engineering 20(1), 2016.1, 94-102(9 pages)

[3] Z. Durumeric, J. Kasten, F. Li, J.Amann, J. Beekman, M. Payer, N. Weaver, J. A. Halderman, V. Paxson,and M. Bailey. "The matter ofHeartbleed," Proceedings of the 2014 ACM Internet Measurement Conference,pp. 475-488, Nov. 2014.

[4] "사이버 공격자들, 사이퍼 스턴팅이라는 기술로 악성 트래픽 감춰", 2019.5.16, https://www.boannews.com/media/view.asp?idx=79570&kind=&sub_kin=

[5] “SSL/TLS의 이해와 TLS 1.3으로 업그레이드해야 하는 이유” 2018.12.12 http://www.itworld.co.kr/news/113007

[6] “TSL”, 2019.05.27. https://namu.wiki/w/TLS#s-1.35-tls 1.3