샘플 트랜잭션 및 패킷 교환을 통한 SSL 소개

목차

소개

이 문서에서는 SSL(Secure Sockets Layer) 프로토콜의 기본 개념을 설명하고 샘플 트랜잭션 및 패킷 캡처를 제공합니다.

SSL 레코드 개요

SSL의 기본 데이터 단위는 레코드입니다.각 레코드는 5바이트 레코드 헤더와 그 뒤에 데이터가 옵니다.

레코드 형식

- 유형:uint8 - 나열된 값
- **버전**:uint16
- **길이**:uint16

유형	**버전**		**길이**	
T	VH	VL	LH	LL

레코드 유형

SSL에는 4가지 레코드 유형이 있습니다.

- **핸드셰이크**(22, 0x16)
- **암호 사양 변경**(20, 0x14)

- **경고**(21, 0x15)
- **애플리케이션 데이터**(23, 0x17)

레코드 버전

레코드 버전은 16비트 값이며 네트워크 순서로 포맷됩니다.

> **참고:**SSL 버전 3(SSLv3)의 경우 버전은 0x0300입니다. TLSv1(Transport Layer Security Version 1)의 경우 버전은 0x0301입니다. Cisco ASA(Adaptive Security Appliance)는 0x002 버전을 사용하는 SSL 버전 2(SSLv2)를 지원하지 않습니다. TLS 버전 0보다 큰 버전은 0입니다. v1.

레코드 길이

레코드 길이는 16바이트 값이며 네트워크 순서로 포맷됩니다.

이론적으로 단일 레코드의 길이는 최대 65,535(2^16 -1)바이트임을 의미합니다.TLSv1 RFC2246은 최대 길이가 16,383(2^14-1)바이트임을 나타냅니다.Microsoft 제품(Microsoft Internet Explorer 및 Internet Information Services)은 이러한 제한을 초과하는 것으로 알려져 있습니다.

레코드 유형

이 섹션에서는 네 가지 유형의 SSL 레코드에 대해 설명합니다.

핸드셰이크 레코드

핸드셰이크 레코드는 핸드셰이크에 사용되는 메시지 집합을 포함합니다.다음은 메시지와 그 값입니다.

- **Hello 요청**(0, 0x00)
- **클라이언트 hello**(1, 0x01)
- **서버 hello**(2, 0x02)
- **인증서**(11, 0x0B)
- **서버 키 교환**(12, 0x0C)
- **인증서 요청**(13, 0x0D)
- **서버 hello 완료**(14, 0x0E)
- **인증서 확인**(15, 0x0F)
- **클라이언트 키 교환**(16, 0x10)
- **완료**(20, 0x14)

간단한 경우 핸드셰이크 레코드는 암호화되지 않습니다.그러나 완료된 메시지를 포함하는 핸드셰이크 레코드는 항상 암호화됩니다. 이는 항상 CCS(Change Cipher Spec) 레코드 이후에 발생합니다.

CCS 레코드

CCS 레코드는 암호화 암호가 변경되었음을 나타내기 위해 사용됩니다.CCS 레코드 직후 모든 데이터는 새 암호로 암호화됩니다.CCS 레코드가 암호화되거나 암호화되지 않을 수 있습니다.단일 핸드

셰이크와의 간단한 연결에서는 CCS 레코드가 암호화되지 않습니다.

경고 레코드

경고 레코드는 상태가 발생했음을 피어에 나타내기 위해 사용됩니다.일부 경보는 경고이며, 다른 경보는 치명적이며 연결이 실패합니다.경고는 암호화되거나 암호화되지 않을 수 있으며 핸드셰이크 또는 데이터 전송 중에 발생할 수 있습니다.두 가지 유형의 알림이 있습니다.

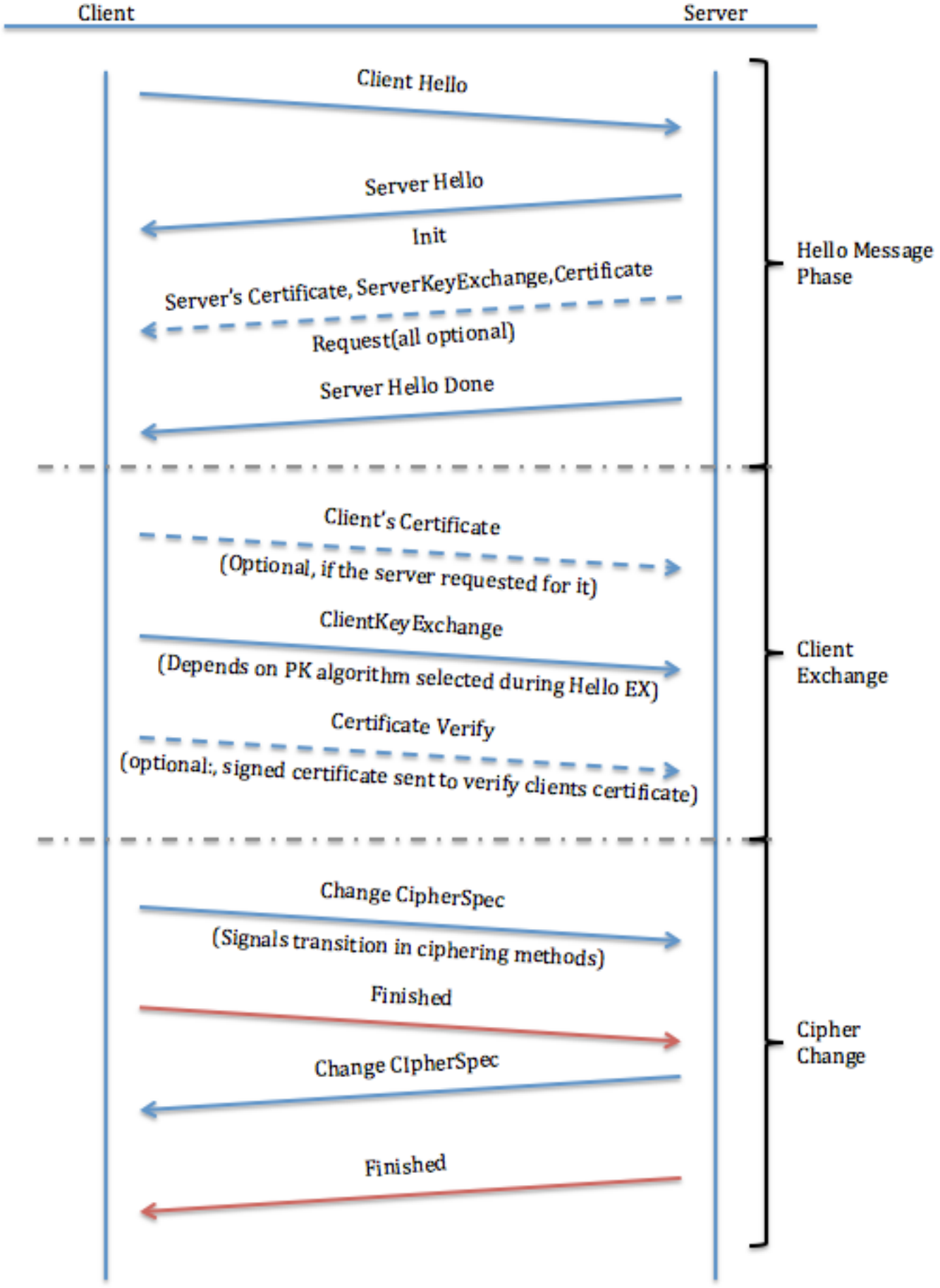
- **마감 경고:**어떤 종류의 잘림 공격을 방지하려면 클라이언트와 서버 간의 연결을 올바르게 닫아야 합니다.**close_notify** 메시지가 수신자에게 전송되어 발신자가 해당 연결에 대해 더 이상 메시지를 보내지 않음을 나타냅니다.
- **오류 경고:**오류가 감지되면 탐지 대상은 상대방에게 메시지를 보냅니다. 치명적인 경보 메시지를 전송하거나 수신하면 양 당사자는 즉시 연결을 닫습니다.오류 알림의 몇 가지 예는 다음과 같습니다.
  - **unexpected_message**(치명적)
  - **압축 해제_실패**
  - **handshake_failure**

애플리케이션 데이터 레코드

이러한 레코드에는 실제 애플리케이션 데이터가 포함됩니다.이러한 메시지는 레코드 레이어에서 전달되며 현재 연결 상태를 기반으로 조각화, 압축 및 암호화됩니다.

샘플 트랜잭션

이 섹션에서는 클라이언트와 서버 간의 샘플 트랜잭션에 대해 설명합니다.



Hello Exchange

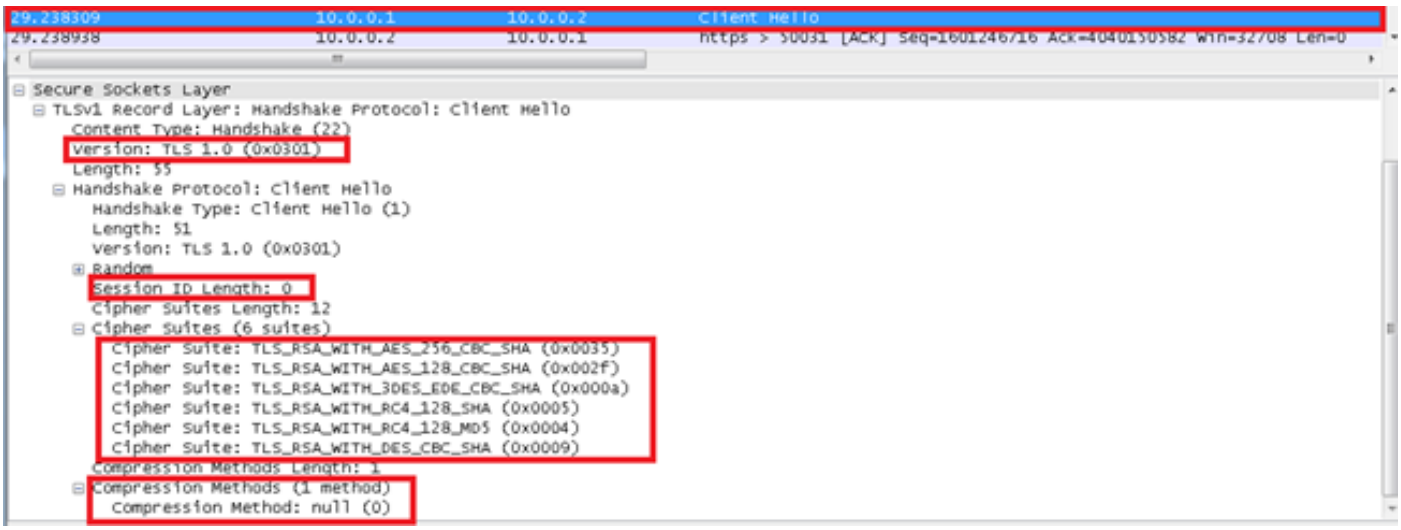
SSL 클라이언트와 서버가 통신을 시작하면 프로토콜 버전에 동의하고, 암호화 알고리즘을 선택하고, 선택적으로 서로를 인증하며, 공유 암호를 생성하기 위해 공개 키 암호화 기술을 사용합니다.이러한 프로세스는 핸드셰이크 프로토콜에서 수행됩니다.요약하면, 클라이언트는 서버에 Client Hello 메시지를 보냅니다. 이 메시지는 Server Hello 메시지로 응답해야 합니다. 그렇지 않으면 치명적인 오류가 발생하여 연결이 실패합니다.Client Hello 및 Server Hello는 클라이언트와 서버 간에 보안 개선 기능을 설정하는 데 사용됩니다.

클라이언트 hello

Client Hello는 다음 특성을 서버로 전송합니다.

- **프로토콜 버전:**이 세션 동안 클라이언트가 통신하려는 SSL 프로토콜의 버전입니다.
- **세션 ID:**클라이언트가 이 연결에 사용할 세션의 ID입니다.exchange의 첫 번째 Client Hello에서 세션 ID가 비어 있습니다(참고 후 패킷 캡처 화면 샷 참조).
- **암호 그룹:**클라이언트에서 Client Hello 메시지의 서버로 전달됩니다.클라이언트의 기본 설정(첫 번째 선택)에 따라 클라이언트에서 지원하는 암호화 알고리즘의 조합을 포함합니다. 각 암호 그룹은 키 교환 알고리즘과 암호 사양을 모두 정의합니다.서버는 암호 그룹을 선택하거나 허용 가능한 선택 항목이 표시되지 않으면 핸드셰이크 실패 알림을 반환하고 연결을 닫습니다.
- **압축 방법:**클라이언트에서 지원하는 압축 알고리즘 목록을 포함합니다.서버가 클라이언트에서 보낸 방법을 지원하지 않으면 연결이 실패합니다.압축 방법은 null일 수도 있습니다.

  참고:캡처의 서버 IP 주소는 10.0.0.2이고 클라이언트 IP 주소는 10.0.0.1입니다.

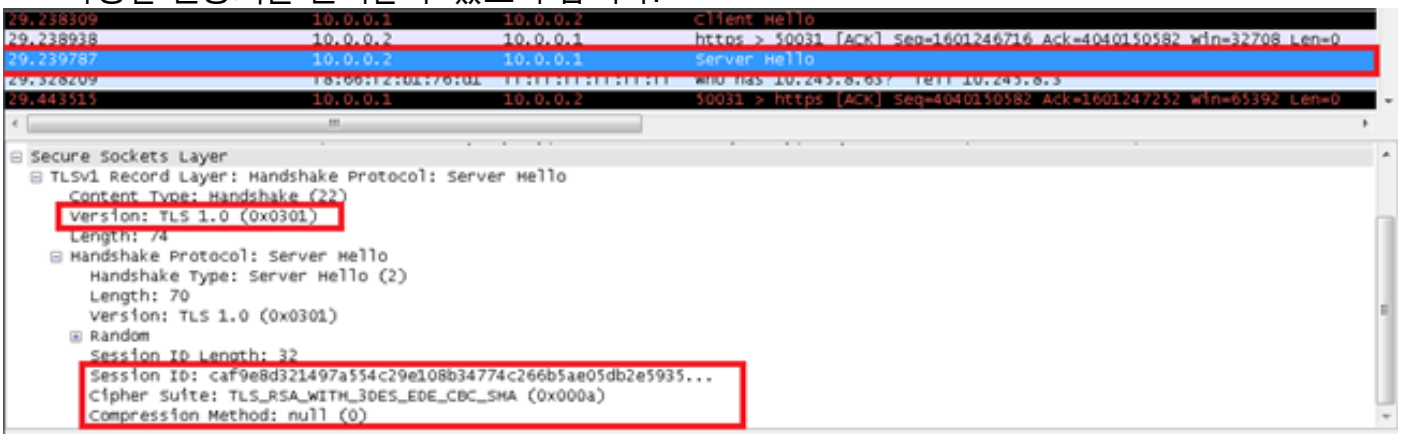


서버 Hello

서버는 다음 특성을 클라이언트로 다시 전송합니다.

- **프로토콜 버전:**클라이언트가 지원하는 SSL 프로토콜의 선택한 버전.
- **세션 ID:**이 연결에 해당하는 세션의 ID입니다.Client Hello에서 클라이언트가 보낸 세션 ID가 비어 있지 않으면 서버는 세션 캐시에서 매칭을 검색합니다.일치하는 항목이 발견되고 서버가 지정된 세션 상태를 사용하여 새 연결을 설정하려는 경우 서버는 클라이언트에서 제공한 것과 동일한 값으로 응답합니다.이는 재개된 세션을 나타내며, 당사자가 완료된 메시지로 직접 진행해야 함을 나타냅니다.그렇지 않으면 이 필드에는 새 세션을 식별하는 다른 값이 포함됩니다.서버가 빈 **session_id를 반환하여** 세션이 캐시되지 않으므로 다시 시작할 수 없음을 나타낼 수 있습니다.
- **암호 그룹:**클라이언트에서 보낸 목록에서 서버에서 선택한 대로

- **압축 방법:**클라이언트에서 보낸 목록에서 서버에서 선택한 대로
- **인증서 요청:**서버는 클라이언트에 구성된 모든 인증서 목록을 전송하며 클라이언트가 인증에 사용할 인증서를 선택할 수 있도록 합니다.

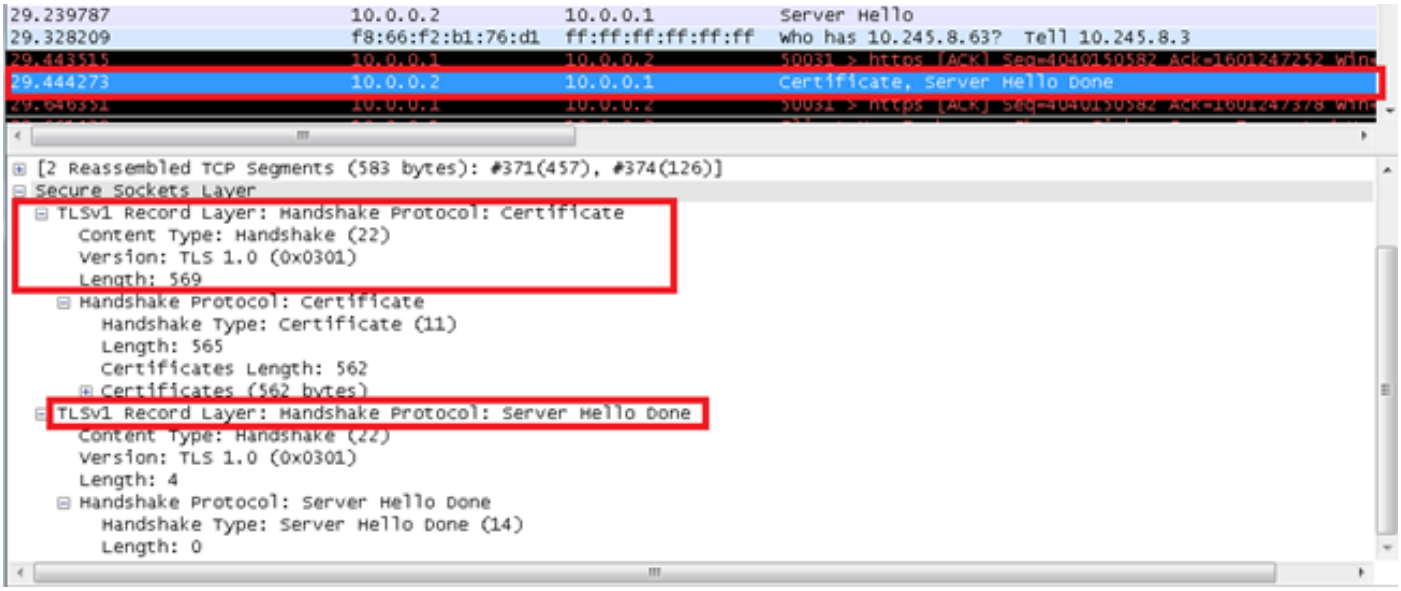


SSL 세션 재개 요청의 경우:

- 서버는 클라이언트에도 Hello 요청을 보낼 수 있습니다.이는 편의상 클라이언트 Hello 요청을 사용하여 재협상을 시작해야 한다는 것을 클라이언트에게 알리는 것입니다.핸드셰이크 프로세스가 이미 진행 중인 경우 클라이언트는 서버의 Hello 요청을 무시합니다.
- 핸드셰이크 메시지는 애플리케이션 데이터 전송보다 우선합니다.재협상은 최대 길이 응용 프로그램 데이터 메시지의 전송 시간의 1~2배까지만 시작해야 합니다.

서버 hello 완료

서버 hello 완료 메시지는 서버 hello 및 관련 메시지의 끝을 나타내기 위해 서버에서 전송합니다.이 메시지를 전송하면 서버는 클라이언트 응답을 기다립니다.Server Hello Done(서버 hello 완료) 메시지를 수신하면 클라이언트는 필요한 경우 서버가 유효한 인증서를 제공했는지 확인하고 Server Hello 매개변수가 허용되는지 확인합니다.



서버 인증서, 서버 키 교환 및 인증서 요청(선택 사항)

- **서버 인증서:** 서버를 인증해야 하는 경우(일반적으로 예:) 서버는 Server Hello 메시지 바로 뒤에 인증서를 보냅니다.인증서 유형은 선택한 암호 그룹 키 교환 알고리즘에 적합해야 하며 일반적으로 X.509.v3 인증서입니다.
- **서버 키 교환:** 인증서가 없는 경우 서버에서 서버 키 교환 메시지를 보냅니다.서버 인증서에

Diffie-Hellman(DH) 매개변수가 포함된 경우 이 메시지는 사용되지 않습니다.

- **인증서 요청:** 선택한 암호 그룹에 적합한 경우 서버는 선택적으로 클라이언트에서 인증서를 요청할 수 있습니다.

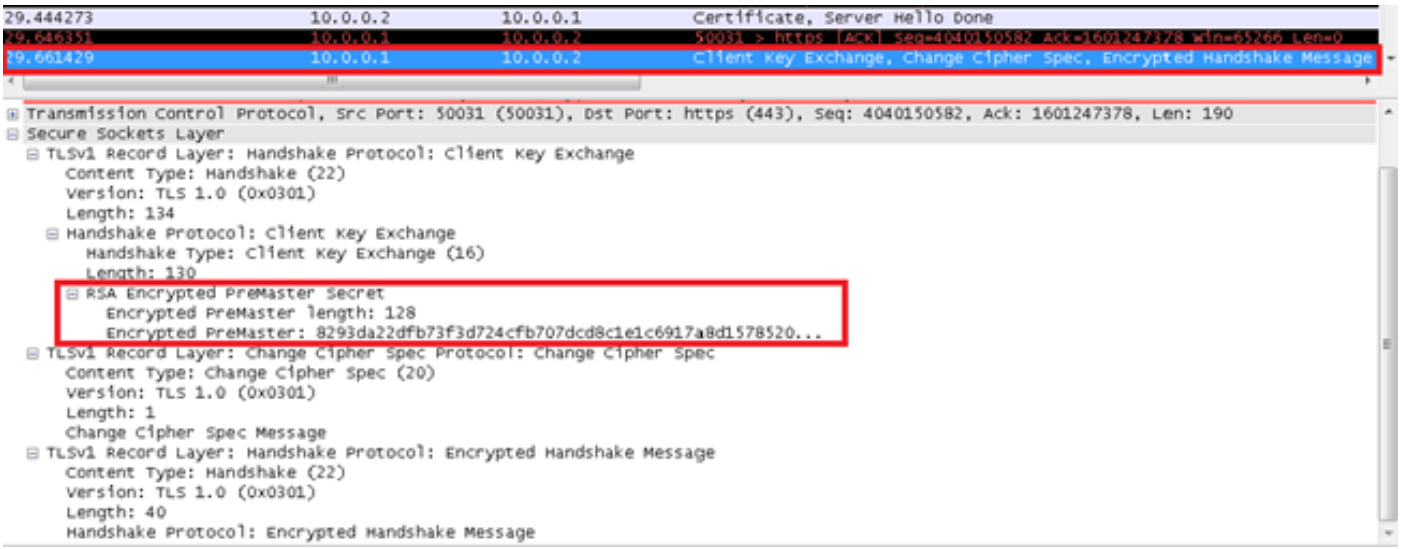
클라이언트 교환

클라이언트 인증서(선택 사항)

이것은 클라이언트가 Server Hello Done 메시지를 받은 후 보내는 첫 번째 메시지입니다.이 메시지는 서버가 인증서를 요청하는 경우에만 전송됩니다.적합한 인증서를 사용할 수 없는 경우 클라이언트는 **no_certificate** 알림**을** 대신 전송합니다.이 경고는 경고일 뿐입니다.그러나 클라이언트 인증이 필요한 경우 서버는 치명적인 핸드셰이크 실패 알림으로 응답할 수 있습니다.클라이언트 DH 인증서는 서버에서 지정한 DH 매개변수와 일치해야 합니다.

클라이언트 키 교환

이 메시지의 내용은 Client Hello 메시지와 Server Hello 메시지 간에 선택한 공개 키 알고리즘에 따라 달라집니다.클라이언트는 Rivest-Shamir-Addleman(RSA) 알고리즘에 의해 암호화된 프리마스터 키 또는 키 계약 및 인증에 DH를 사용합니다.RSA가 서버 인증 및 키 교환에 사용되는 경우 48바이트 **pre_master_secret**이 클라이언트에 의해 생성되고 서버 공개 키로 암호화되어 서버로 전송됩니다.서버는 **pre_master_secret**을 해독하기 위해 개인 키를 **사용합니다**.그런 다음 두 당사자 모두 pre_**master_secret을 master_secret**으로 **변환합니다**.



인증서 확인(선택 사항)

클라이언트가 서명 기능이 있는 인증서를 보낼 경우 인증서를 명시적으로 확인하기 위해 디지털 서명 인증서 확인 메시지가 전송됩니다.

암호 변경

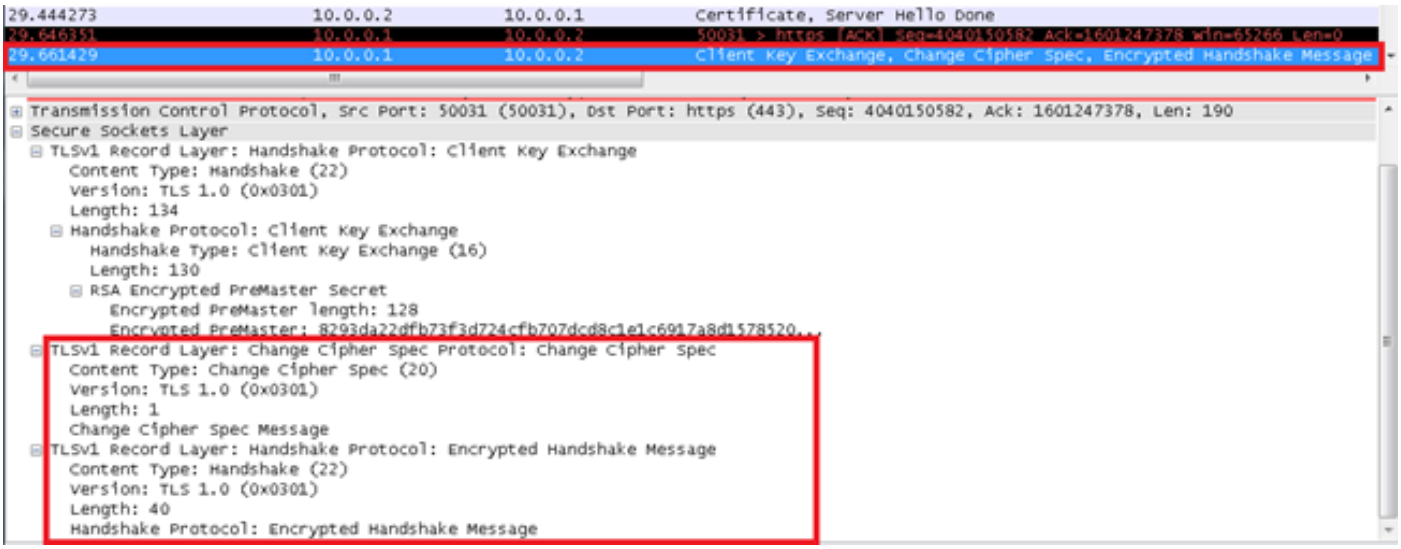
암호 사양 메시지 변경

암호 사양 변경 메시지가 클라이언트에 의해 전송되고, 클라이언트는 보류 중인 암호 사양(새 항목)을 현재 암호 사양(이전에 사용된 항목)으로 복사합니다. 암호화 전략의 전환을 신호로 보내기 위해 암호 사양 변경 프로토콜이 있습니다. 프로토콜은 단일 메시지로 구성되며, 이 메시지는 현재(보류 중이 아님) 암호 사양 아래에 암호화되어 압축됩니다. 후속 레코드가 가장 최근에 협상된 암호 사

양 및 키 아래에서 보호됨을 수신자에게 알리기 위해 클라이언트와 서버 모두에 의해 전송됩니다.이 메시지를 수신하면 수신자가 읽기 보류 상태를 읽기 현재 상태로 복사합니다.클라이언트는 핸드셰이크 키 교환 및 인증서 확인 메시지(있는 경우) 후에 암호 사양 변경 메시지를 전송하고, 서버는 클라이언트에서 받은 키 교환 메시지를 성공적으로 처리한 후 이를 전송합니다.이전 세션이 다시 시작되면 Hello 메시지 후에 Change Cipher Spec 메시지가 전송됩니다.캡처에서 Client Exchange, Change Cipher 및 Finished 메시지는 클라이언트에서 단일 메시지로 전송됩니다.

완료된 메시지

키 교환 및 인증 프로세스가 성공했는지 확인하기 위해 Change Cipher Spec(암호 사양 변경) 메시지 직후 Finished(완료) 메시지가 항상 전송됩니다.[완료] 메시지는 가장 최근에 협상된 알고리즘, 키 및 암호를 가진 첫 번째 보호된 패킷입니다. 완료 메시지에 대한 승인이 필요하지 않습니다.당사자가 Finished 메시지를 보낸 후 즉시 암호화된 데이터를 전송할 수 있습니다.완료된 메시지의 수신자는 내용이 올바른지 확인해야 합니다.



관련 정보

- RFC 6101 - The Secure Sockets Layer Protocol 버전 3.0
- Wireshark SSL Wiki - Wireshark를 사용하여 SSL 패킷 해독
- 기술 지원 및 문서 – Cisco Systems