SSL-Einführung mit Beispiel-Transaktion und Packet Exchange

Inhalt

Einführung

Dieses Dokument beschreibt die grundlegenden Konzepte des Secure Sockets Layer (SSL)-Protokolls und enthält eine Beispieltransaktion und Paketerfassung.

Übersicht über SSL-Datensätze

Die grundlegende Dateneinheit in SSL ist ein Datensatz. Jeder Datensatz besteht aus einem Datensatz-Header mit 5 Byte, gefolgt von Daten.

Datensatzformat

- **Typ**: uint8 - Werte aufgelistet
- **Version**: Uint16
- **Länge**: Uint16

Typ	Version		Länge	
T	VH	VL	LH	LL

Datensatztyp

In SSL gibt es vier Datensatztypen:

- **Handshake** (22, 0 x 16)
- **Cipher-Spez. ändern** (20, 0x14)
- **Alarm** (21, 0 x 15)
- **Anwendungsdaten** (23, 0 x 17)

Datensatzversion

Die Datensatzversion ist ein 16-Bit-Wert und in der Netzwerkreihenfolge formatiert.

> **Hinweis**: Für SSL Version 3 (SSLv3) ist die Version 0x0300. Für Transport Layer Security Version 1 (TLSv1) ist die Version 0x0301. Die Cisco Adaptive Security Appliance (ASA) unterstützt nicht SSL Version 2 (SSLv2), die die Version 0x0002 oder eine Version von TLS größer als TLSv1 verwendet.

Datensatzlänge

Die Datensatzlänge ist ein 16-Byte-Wert und wird in der Netzwerkreihenfolge formatiert.

Theoretisch bedeutet dies, dass ein einzelner Datensatz bis zu 65.535 (2^16-1) Byte lang sein kann. Der TLSv1 RFC2246 gibt an, dass die maximale Länge 16.383 (2^14-1) Byte beträgt. Microsoft-Produkte (Microsoft Internet Explorer und Internet Information Services) überschreiten diese Grenzen.

Arten von Datensätzen

In diesem Abschnitt werden die vier Typen von SSL-Datensätzen beschrieben.

Handshake-Datensätze

Handshake-Datensätze enthalten eine Reihe von Nachrichten, die zum Handshake verwendet werden. Dies sind die Meldungen und ihre Werte:

- **Hello-Anfrage** (0, 0x00)
- **Client Hello** (1, 0x01)
- **ServerHello** (2, 0x02)
- **Zertifikat** (11, 0x0B)
- **Server Key Exchange** (12, 0x0C)
- **Zertifikatsanforderung** (13, 0x0D)
- **Server Hello Done** (14, 0x0E)
- **Zertifikatüberprüfung** (15, 0x0F)
- **Client Key Exchange** (16, 0x10)
- **Fertig** (20, 0 x 14)

Im einfachen Fall werden Handshake-Datensätze nicht verschlüsselt. Ein Handshake-Datensatz, der eine fertige Nachricht enthält, wird jedoch immer verschlüsselt, wie es immer nach einem Change Cipher Spec (CCS)-Datensatz geschieht.

CCS-Datensätze

CCS-Datensätze werden verwendet, um auf eine Änderung der kryptografischen Verschlüsselung hinzuweisen. Unmittelbar nach dem CCS-Datensatz werden alle Daten mit der neuen Verschlüsselung verschlüsselt. CCS-Datensätze werden möglicherweise verschlüsselt oder nicht verschlüsselt. In einer einfachen Verbindung mit einem Handshake wird der CCS-Datensatz nicht verschlüsselt.

Alert-Datensätze

Mithilfe von Warndatensätzen wird dem Peer mitgeteilt, dass eine Bedingung aufgetreten ist. Einige Warnmeldungen sind Warnungen, während andere schwerwiegend sind und die Verbindung zum Ausfall führen. Warnmeldungen können verschlüsselt sein oder nicht, und sie können während eines Handshakes oder während der Datenübertragung auftreten. Es gibt zwei Arten von Warnmeldungen:

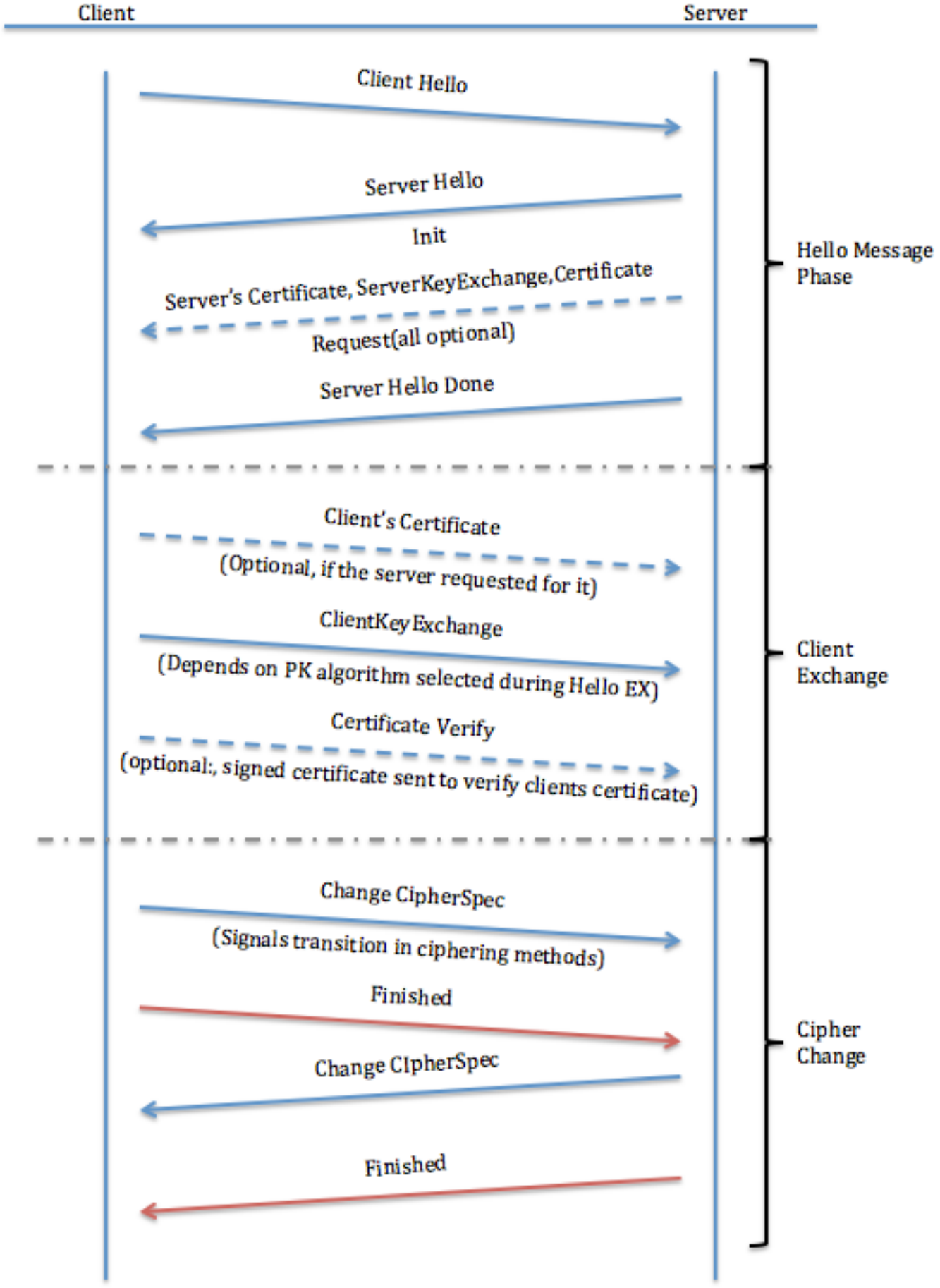
- **Abschlusswarnungen:** Die Verbindung zwischen Client und Server muss ordnungsgemäß geschlossen sein, um Abspaltungsangriffe zu vermeiden. Eine **close_notify-**Nachricht wird an den Empfänger gesendet, dass der Absender in dieser Verbindung keine Nachrichten mehr sendet.
- **Fehlerwarnungen:** Wenn ein Fehler erkannt wird, sendet der Erkenner eine Nachricht an den anderen Teilnehmer. Nach der Übermittlung oder dem Empfang einer tödlichen Warnmeldung schließen beide Parteien die Verbindung sofort. Einige Beispiele für Fehlerwarnungen sind:
  - **unerwartete_Nachricht** (fatal)
  - **Dekomprimierung_Ausfall**
  - **Handshake_Failure**

Anwendungsdatenaufzeichnung

Diese Datensätze enthalten die tatsächlichen Anwendungsdaten. Diese Nachrichten werden von der Datensatzschicht übertragen und sind je nach aktuellem Verbindungsstatus fragmentiert, komprimiert und verschlüsselt.

Beispieltransaktion

In diesem Abschnitt wird eine Beispieltransaktion zwischen dem Client und dem Server beschrieben.



Hello Exchange

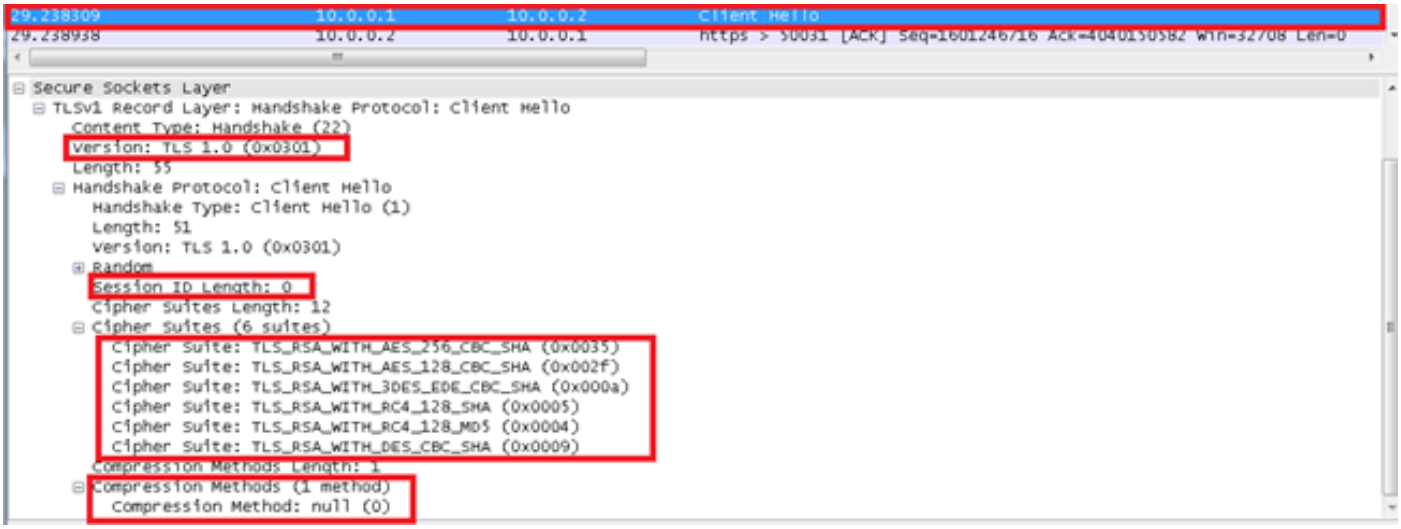
Wenn ein SSL-Client und -Server mit der Kommunikation beginnen, vereinbaren sie eine Protokollversion, wählen Verschlüsselungsalgorithmen aus, authentifizieren sich optional gegenseitig und verwenden Verschlüsselungstechniken für öffentliche Schlüssel, um gemeinsam genutzte Geheimnisse zu generieren. Diese Prozesse werden im Handshake-Protokoll ausgeführt. Zusammenfassend sendet der Client eine Client Hello-Nachricht an den Server, die mit einer Server Hello-Meldung reagieren muss, oder es tritt ein schwerwiegender Fehler auf, und die Verbindung schlägt fehl. Der Client Hello und der Server Hello dienen zur Einrichtung von Funktionen zur Erhöhung der Sicherheit zwischen dem Client und dem Server.

Client-Hello

Der Client Hello sendet diese Attribute an den Server:

- **Protokollversion:** Die Version des SSL-Protokolls, über das der Client während dieser Sitzung kommunizieren möchte.
- **Sitzungs-ID:** Die ID einer Sitzung, die der Client für diese Verbindung verwenden möchte. Im ersten Client Hello des Austauschs ist die Session-ID leer (siehe Screenshot der Paketerfassung nach dem Hinweis).
- **Cipher Suite:** Diese wird in der Client Hello-Nachricht vom Client an den Server übergeben. Es enthält die Kombinationen von kryptografischen Algorithmen, die vom Client unterstützt werden, in der Reihenfolge der vom Client bevorzugten Einstellungen (erste Wahl). Jede Verschlüsselungssuite definiert sowohl einen Schlüsselaustauschalgorithmus als auch eine Verschlüsselungsspez. Der Server wählt eine Verschlüsselungssuite aus oder gibt, wenn keine akzeptable Auswahl angezeigt wird, eine Handshake-Fehlerwarnung zurück und schließt die Verbindung.
- **Komprimierungsmethode:** Enthält eine Liste der vom Client unterstützten Komprimierungsalgorithmen. Wenn der Server keine vom Client gesendete Methode unterstützt, schlägt die Verbindung fehl. Die Komprimierungsmethode kann auch NULL sein.

  Hinweis: Die Server-IP-Adresse in den Erfassungen lautet 10.0.0.2, die Client-IP-Adresse 10.0.0.1.

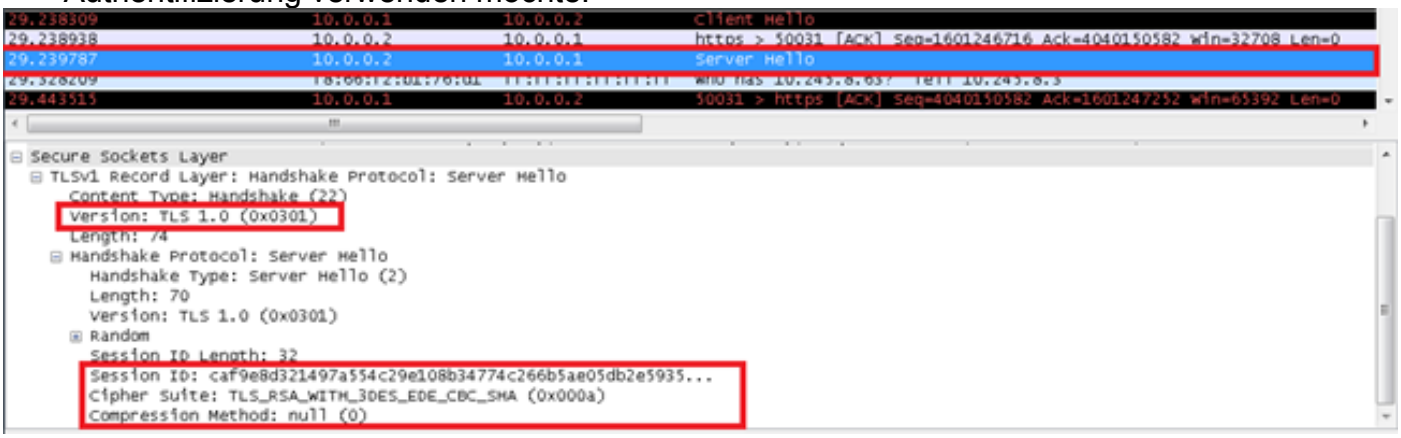


ServerHello

Der Server sendet diese Attribute an den Client zurück:

- **Protokollversion:** Die ausgewählte Version des SSL-Protokolls, das der Client unterstützt.

- **Sitzungs-ID:** Dies ist die Identität der Sitzung, die dieser Verbindung entspricht. Wenn die vom Client im Client Hello gesendete Sitzungs-ID nicht leer ist, sucht der Server im Sitzungscache nach einer Übereinstimmung. Wenn eine Übereinstimmung gefunden wird und der Server bereit ist, die neue Verbindung mithilfe des angegebenen Sitzungszustands herzustellen, antwortet der Server mit dem gleichen Wert, der vom Client bereitgestellt wurde. Dies weist auf eine wiederaufgenommene Sitzung hin und schreibt vor, dass die Parteien direkt zu den fertig gestellten Nachrichten fortfahren müssen. Andernfalls enthält dieses Feld einen anderen Wert, der die neue Sitzung identifiziert. Der Server kann eine leere **session_id** zurückgeben, um anzugeben, dass die Sitzung nicht zwischengespeichert wird und daher nicht wiederhergestellt werden kann.
- **Cipher Suite:** Wie vom Server aus der Liste ausgewählt, die vom Client gesendet wurde.
- **Komprimierungsmethode:** Wie vom Server aus der Liste ausgewählt, die vom Client gesendet wurde.
- **Zertifikatsanforderung:** Der Server sendet dem Client eine Liste aller auf ihm konfigurierten Zertifikate und ermöglicht dem Client, das Zertifikat auszuwählen, das er für die Authentifizierung verwenden möchte.

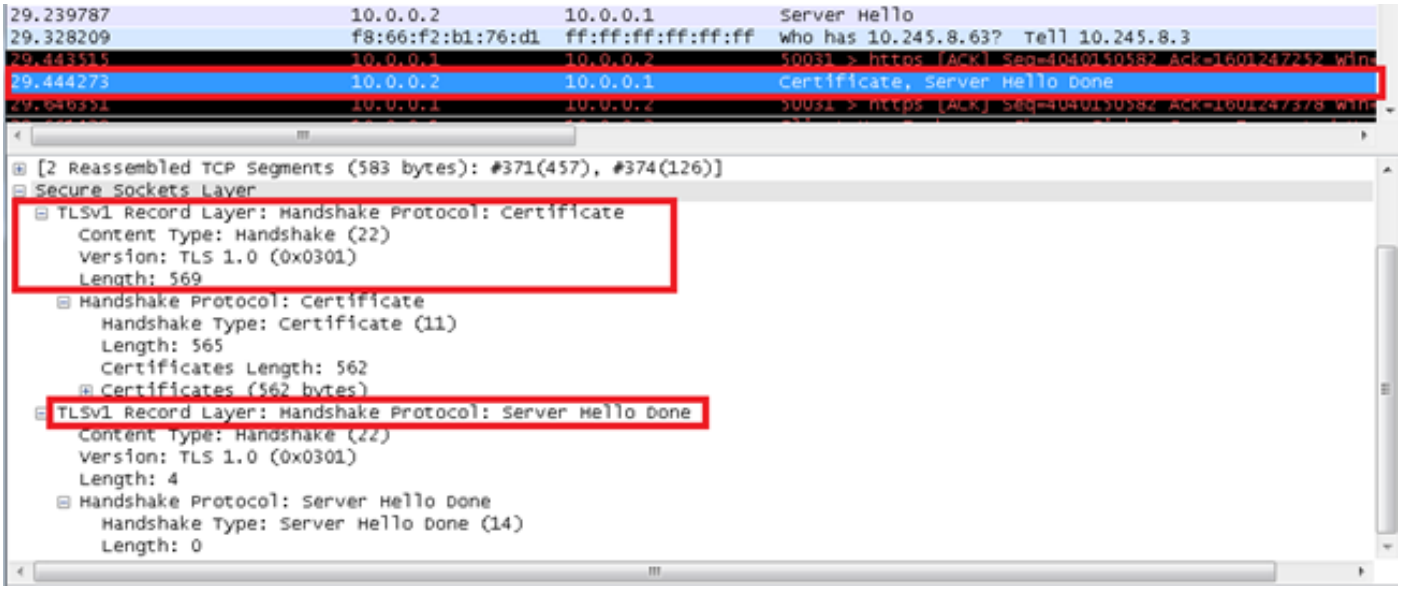


Für Anforderungen zur Wiederaufnahme einer SSL-Sitzung:

- Der Server kann auch eine Hello-Anfrage an den Client senden. Dies dient lediglich dazu, den Kunden daran zu erinnern, dass er die Neuverhandlung mit einer Client Hello-Anfrage beginnen sollte, wenn es zweckmäßig ist. Der Client ignoriert die Hello-Anforderung vom Server, wenn der Handshake-Prozess bereits ausgeführt wird.
- Die Handshake-Meldungen haben mehr Vorrang vor der Übertragung von Anwendungsdaten. Die Neuverhandlung darf nicht mehr als ein- oder zweimal so lange beginnen wie eine Datenmeldung mit einer maximalen Länge von Anwendungen.

Server Hello Fertig

Die Meldung Server Hello Done (ServerHello-Fertig) wird vom Server gesendet, um das Ende des Serverbetriebs und der zugehörigen Nachrichten anzugeben. Nachdem er diese Nachricht gesendet hat, wartet der Server auf eine Client-Antwort. Nach Erhalt der Meldung Server Hello Done (ServerHello abgeschlossen) überprüft der Client, ob der Server ggf. ein gültiges Zertifikat bereitgestellt hat, und ob die Parameter Server Hello akzeptiert werden können.



Serverzertifikat, Server Key Exchange und Zertifikatsanforderung (optional)

- **Serverzertifikat:** Wenn der Server authentifiziert werden muss (was im Allgemeinen der Fall ist), sendet der Server sein Zertifikat sofort nach der Meldung "Server Hello". Der Zertifikatstyp muss für den ausgewählten Schlüsselaustauschalgorithmus der Verschlüsselungssuite geeignet sein und ist in der Regel ein X.509.v3-Zertifikat.
- **Server Key Exchange:** Die Exchange-Nachricht für den Serverschlüssel wird vom Server gesendet, wenn er kein Zertifikat hat. Wenn die DH-Parameter (Diffie-Hellman) im Serverzertifikat enthalten sind, wird diese Meldung nicht verwendet.
- **Zertifikatsanforderung:** Ein Server kann optional ein Zertifikat vom Client anfordern, falls dies für die ausgewählte Verschlüsselungssuite erforderlich ist.

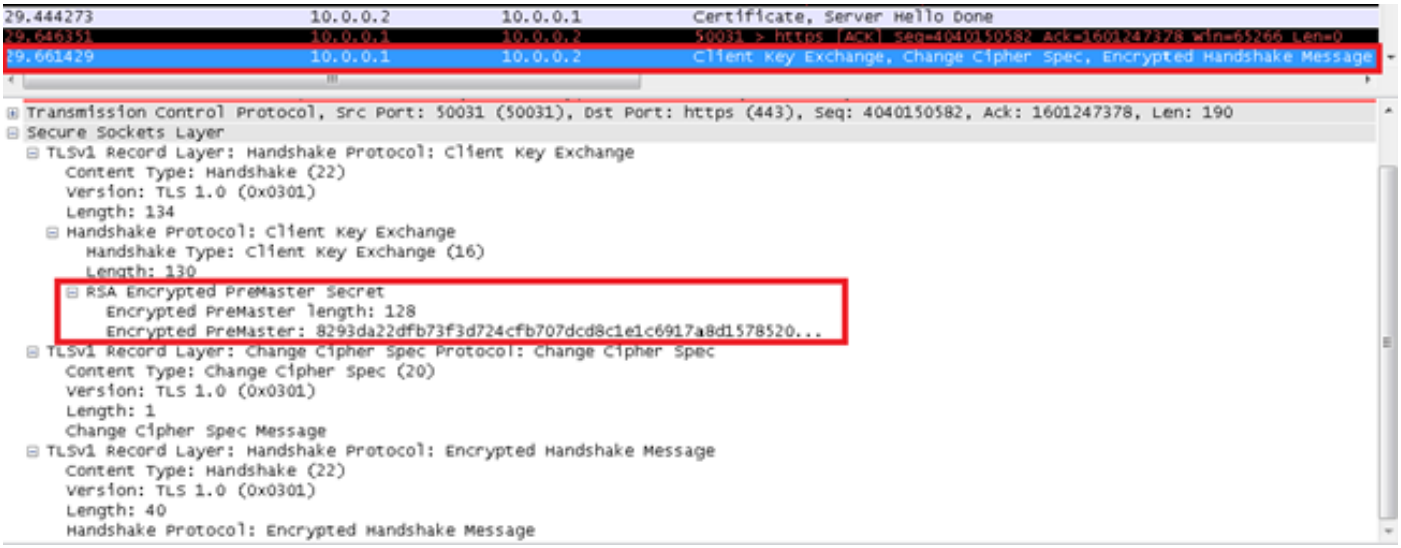
Client-Exchange

Client-Zertifikat (optional)

Dies ist die erste Meldung, die der Client sendet, nachdem er eine "Server Hello Done"-Nachricht erhält. Diese Meldung wird nur gesendet, wenn der Server ein Zertifikat anfordert. Wenn kein geeignetes Zertifikat verfügbar ist, sendet der Client stattdessen eine **no_certificate**-Warnung. Diese Warnung ist nur eine Warnung. Wenn eine Client-Authentifizierung erforderlich ist, kann der Server jedoch mit einer schwerwiegenden Handshake-Fehlermeldung reagieren. Client-DH-Zertifikate müssen mit den vom Server angegebenen DH-Parametern übereinstimmen.

Client Key Exchange

Der Inhalt dieser Nachricht hängt vom Algorithmus des öffentlichen Schlüssels ab, der zwischen den Meldungen Client Hello und Server Hello ausgewählt wurde. Der Client verwendet entweder einen durch den Rivest-Shamir-Addleman (RSA)-Algorithmus verschlüsselten Vormaster-Schlüssel oder DH für die Schlüsselvereinbarung und Authentifizierung. Wenn RSA für die Serverauthentifizierung und den Schlüsselaustausch verwendet wird, wird ein 48-Byte-**Pre_master_secret** vom Client generiert, unter dem öffentlichen Serverschlüssel verschlüsselt und an den Server gesendet. Der Server verwendet den privaten Schlüssel, um den **pre_master_secret** zu entschlüsseln. Beide Parteien konvertieren dann den **pre_master_secret** in den **master_secret**.



Zertifikatüberprüfung (optional)

Wenn der Client ein Zertifikat mit Signierbarkeit sendet, wird eine digital signierte Certificate Verify-Nachricht gesendet, um das Zertifikat explizit zu überprüfen.

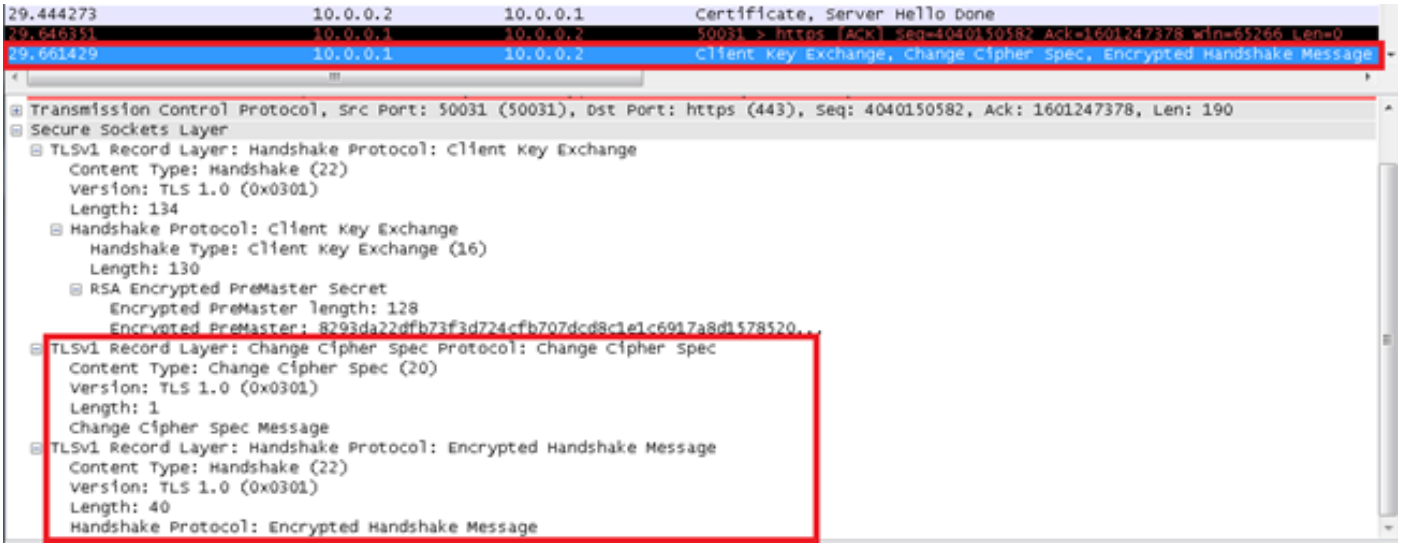
Verschlüsselungsänderung

Cipher-Spec-Meldungen ändern

Die Nachricht "Cipher Spec ändern" wird vom Client gesendet, und der Client kopiert die ausstehende Cipher Spec (die neue) in die aktuelle Cipher Spec (die vorher verwendete Cipher Spec). Das Protokoll "Cipher Spec ändern" existiert, um Übergänge in Verschlüsselungsstrategien zu signalisieren. Das Protokoll besteht aus einer einzelnen Nachricht, die verschlüsselt und unter der aktuellen (nicht ausstehenden) Cipher-Spez komprimiert wird. Die Nachricht wird sowohl vom Client als auch vom Server gesendet, um den Empfänger darüber zu informieren, dass nachfolgende Datensätze unter den zuletzt ausgehandelten Cipher Spec und Schlüsseln geschützt sind. Beim Empfang dieser Nachricht kopiert der Empfänger den ausstehenden Lesezustand in den aktuellen Zustand. Der Client sendet nach dem Austausch von Handshake-Schlüsseln und (falls zutreffend) Zertifikatsüberprüfungsmeldungen eine Change Cipher Spec-Nachricht, und der Server sendet eine Nachricht, nachdem er die vom Client erhaltene Schlüsselaustauschmeldung erfolgreich verarbeitet hat. Wenn eine vorherige Sitzung wiederhergestellt wird, wird nach den Hello-Nachrichten die Meldung Change Cipher Spec (Spezifikation ändern) gesendet. In den Captures werden die Nachrichten Client Exchange, Change Cipher und Beendet als eine einzige Nachricht vom Client gesendet.

Abgeschlossene Nachrichten

Eine Fertig gestellte Nachricht wird immer sofort nach einer Change Cipher Spec Nachricht gesendet, um zu überprüfen, ob der Schlüsselaustausch- und Authentifizierungsprozess erfolgreich war. Die "Fertig gestellt"-Nachricht ist das erste geschützte Paket mit den zuletzt ausgehandelten Algorithmen, Schlüsseln und Geheimnissen. Es ist keine Bestätigung der fertig gestellten Nachricht erforderlich. Die Parteien können sofort nach dem Senden der fertigen Nachricht mit dem Versenden verschlüsselter Daten beginnen. Empfänger von fertig gestellten Nachrichten müssen überprüfen, ob der Inhalt korrekt ist.



Zugehörige Informationen

- RFC 6101 - Secure Sockets Layer Protocol Version 3.0
- Wireshark SSL Wiki - Entschlüsseln von SSL-Paketen mit Wireshark
- Technischer Support und Dokumentation - Cisco Systems