Kriptográfia - Transport Layer Security

Hamza Balázs

A Transport Layer Security (röviden: TLS) egy kriptográfiai protokoll, amely az Interneten keresztüli kommunkációhoz nyujt védelmet. Ez a protokoll gyakran használt olyan applikációknál, mint az email, instant üzenetküldés, de a leggyakoribb használata a HTTPS biztosítása.

A TLS protokoll első sorban kriptográfia segitségével teszi lehetővé a biztonságos kommunikációt, elhárítva a lehallgathatóságot és az esetleges hamisitást/módositást. Az Internet Engineering Task Force (IETF) ajánlotta, mint internetes szabvány, először 1999-ben. A kurrens verzió a TLS 1.3, amelyet 2018-ban definiáltak, a régi SSL specifikációkra épül fel.

Kliens-szerver applikácíok használják a TLS protokollt, hogy biztositsák a biztonságós kommunikációt. Az applikációknak kérnie kell a szervertől a TLS-kapcsolat létrehozását, mivel TLS/SSL nélkül is tudnak kommunikálni. Ennek elérésének egyik fő módja az, hogy külön port számót használunk a TLS kapcsolatoknak. Például a 80-as port számot általában titkosítatlan HTTP forgalomhoz használják, míg a 443-as port számot titkosított HTTP forgalomhoz használják. Egy másik mód a TLS-kapcsolat létrehozására az, hogy a kliens protokoll-specifikus kérést küld a szervernek, hogy kapcsolja át a kapcsolatot TLS-re. Ehhez leggyakrabban a STARTTLS kérést kezdeményezi, főképp email vagy hír-protokoll esetében

Amint a kliens és a szerver megállapodott, hogy TLS-kapcsolatot használjanak, létrehoznak egy munkamenetet a handshake protokoll segitségével. A handshake akkor kezdődik el, amikor a kliens kapcsolódik a TLS-engedélyezett szerverhez és a kliens előáll több kriptográfiai paraméterrel (rejtjelek, hash-funkciók). Ezután a szerver kiválaszt egy vagy több rejtjelt/hash-funkciót és visszajelez a kliensnek a döntéséről. Ezután a szerver biztosít egy digitális aláirást, amivel igazolja magát. Ez az aláirás tartalmazza a szerver nevét, egy tanusítványt és a szerver nyilvános kulcsát. A kliens megerősíti az aláirás érvényességét. Ha érvényes az aláirás, a kliens csatlakozik a szerverhez és elkészíti a munkafolyamat-azonosító kulcsot (session). Ezt a kulcsot kétfajtaképpen tudja létrehozni a kliens:

- Titkosít egy véletlenszerű számot a szerver publikus kulcsának segitségével, amit majd átad a szervernek. Ezt a számot a szerver visszatudja fejteni a privát kulcsa segitségével. Ezután a kliens és a szerver is felhasználja a véletlenszerű számot, hogy egyedi munkafolyamat kulcsokat generáljon a titkosítás és visszafejtéshez.

- Felhasználja a Diffie-Hellman kulcs cserét, hogy biztonságosán generáljon véletlenszerű és egyedi munkafolyamat kulcsokat, amiket a titkosítás és visszafejtéshez használ fel.

Ezek után sikeresen létrejött a biztonságos kapcsolat, a kliens és szerver biztonságban tud kommunikálni egymással. Ha akármelyik lépés sikertelen, a TLS-kapcsolat nem jön létre.

A TLS-kapcsolatnak több paramétere is van. Ezekben a paraméterekben van eltárolva a szerver és kliens random értéke (minden kapcsolatnál más random érték választódik ki), a szerver és kliens MAC értéke (küldött adatok esetében használt érték), a szerver és kliens titkos kulcsa (szimmetrikus titkositásnál használt, egyik fél titkosít, míg a másik visszafejt), az inicializálási vektor (kezdetben a TLS handshake protokoll inicializálja, minden kulcshoz külön vektor jár) és az elküldött és fogadott üzenetek sorszáma (szerver és kliens külön számozza ezeket a sorszámokat).

A TLS-munkamenetnek is több paramétere van, itt tároljuk el a munkamenet azonositóját (random bájtszekvencia, szerver választja), a peer tanusítványát (lehet null, leggyakrabban egy X509.v3 tanusítvány), a tömörítési módszert (adattömörítésre használt algoritmus), a titkosítás paramétereit (MAC-hez használt titkosítási módszer, hash függvény, hash méret), a mester titkot (a kliens és szerver között megosztott 48 bájtos titok) és a jelző értéket (jelzi, hogy a munkamenet használható-e új kapcsolatok indítására).

Ha a Handshake Protokoll sikerrel járt, sikerült elérni egy biztonságos kapcsolatot a kliens és szerver között, már tudnak üzenetet küldeni egymásnak. Az üzeneteket a TLS Record Protokoll titkosítja a Handshake Protokoll által létrehozott kulcsok segítségével.

A TLS Record Protokoll felépitése:

- Application data: Itt tárolódik el az üzenet, ezek az üzenetek lehetnek: ChangeCipherSpec, Alert, Handshake, Application, Heartbeat.

- Fragment : Az üzenet itt fragmentálódik több blokkba, ezek a blokkok nem haladhatják meg a 2^{14} bájt méretet.

- Compress : Itt történik a fragmentek tömörítése, veszteségmentesnek kell lennie és nem növelheti a tartalom méretét 1024 bájtnál többel (opcionális).

- Add MAC : A tartalomhoz hozzáadódik a Message Authentication Code (MAC) vagy HMAC.

- Encrypt : A tartalom titkosítódik, 128 bites kulcsméretü szimmetrikus titkosítással, nem növelheti a tartalom méretét 1024 bájtnál többel. Blokk titkosítás esetében a padding-et a MAC után végzi (AES-128/256, 3DES-168, RC4-128)

- Append TLS record header : Négy tipusu mező hozzáadását jelenti, ebből a négy mezőből három a fragment feldolgozására használt módszereket tartalmazza, a negyedik mező az üzenet hosszát tartalmazza.

Miután a TLS Record Protokoll befejeződik, a titkosított adat tovább küldődik a Transmission Control Protocol (TCP) layernek szállitásra.

A TLS Record Protokoll üzenetei:

- ChangeCipherSpec : egyetlen egybájtos üzenetből áll: jelzi, ha a munkamenet titkosítási módját egy másik rekord módosítja.

- Alert : Két bájtból áll, az első warning vagy fatal értéket jelent, a második a riasztás kódját tartalmazza. A warning érték jelzi, ha a kapcsolat vagy biztonság instabil, míg a fatal érték jelzi, ha a kapcsolat vagy biztonság veszélybe került, vagy javíthatatlan hiba történt.

- Heartbeat: Biztosítja a feladót, hogy a címzett még mindig "életben" van, akkor is ha egy ideig nem volt tevékenység. Ezen kivül aktivitást generál a kapcsolaton keresztül tétlenségi időszakokban, így elkerülhető a kapcsolat megszakadása tétlenség miatt. Ez az üzenet a TLS Record Protokollon felül fut.

- Handshake: Lehetővé teszi a szerver és a kliens kölcsönös hitelesítését. Ezen túl még lehetővé teszi, hogy a felek megállapodjanak egy titkosító- és egy MAC-algoritmus használatában, amit a kommunikáció során fognak használni.

A Handshake protokoll a legösszetettebb része a TLS-nek, itt jön létre a kommunikáció. A protokollnak négy fázisa van:

- Phase 1 (client_hello, server_hello): A kliens elküldi a client_hello üzenetet és még mellette megosztja a kliens random értékét és rejtjelkészletét. A szerver válaszol a server_hello üzenettel, és ő is megosztja a random értékét.

- Phase 2 (certificate, server_key_exchange, certificate_request, server_hello_done): A szerver elküldi a digitális aláírását, és kérhet kulcscserét és aláírást a klienstől. Miután sikeresen elküldte az üzeneteket, a szerver küld egy server_hello_done üzenetet, ezzel lezárva ezt a fázist.

- Phase 3 (certificate, client_key_exchange, certificate_verify): Ha a szerver kérvényezte a kliens aláírását és/vagy a kulcscserét, a kliens elküldi. Ezek után a kliens létrehoz egy elő-főkulcsot amit a szerver publikus kulcsával titkosít, majd elküldi a szervernek. Miután a szerver megkapja az elő-főkulcsot, a kliens és szerver külön-külön legenerálja a főkulcsot és a munkamenet-kulcsot az elő-főkulcs segitségével.

- Phase 4 (change_cipher_spec, finished): A kliens elküldi a change_cipher_spec üzenetet, amivel jelzi a szervernek, hogy elkezdi használni a munkamenet-kulcsokat titkosításra, majd elküldi a finished üzenetet. A szerver megkapja a change_cipher_spec üzenetet, átváltja a record réteg biztonsági állapotát szimmetrikus titkosításra, felhasználva a munkamenet-kulcsokat. Ezek után elküldi a szerver a finished üzenetet.

A négy fázis lejárta után létrejön a biztonságos kapcsolat, a kliens és szerver közötti üzenetek mostmár titkosítva vannak a munkamenet-kulcsok segitségével.

A TLS megjelenése óta számos sikeres támadás jelent meg. Ezeket a támadásokat négy kategóriába sorolhatjuk:

- a handshake protokoll elleni támadások: 1998-ban sikeres támadás a handshake protokoll ellen, az RSA séma támadásán alapult.

- record- és applikáció protokoll elleni támadások: 2011 szeptemberében mutatták be a BEAST-et (Browser Exploit Against SSL/TLS). Ez egy kliens oldali támadás, ami a man-in-the-middle technikát használja fel. A támadó kitudja találni az inicializálási vektort, és ezt a vektort feltudja használni ahhoz, hogy a titkosított üzenetet visszafejtse. Egy másik támadás a CRIME (Compression Ration Info-leak Made Easy), a támadó visszatudja állitani a webes cookie-k tartalmát. Ha egy hitelesítési cookie tartalmát állitja vissza, azzal átveheti a támadó az uralmat a munkamenet felett, így többfajta támadást is eltud indítani.

- PKI (Public Key Infrastructure) elleni támadások: Az X.509 tanusítványok érvényességének ellenőrzése sokfajta támadásnak van kitéve. Ilyen támadások lehetnek DoS (Denial-of-Service) támadások, a Certificate Authority feletti irányitás átvétel (a támadó hamis tanusítványokat állithat ki, így ki tudja adni magát mint akármelyik weboldal).

- egyéb támadások: 2011-ben bejelentett a The Hackers Choice német hackercsoport egy olyan DoS támadást, amely megterhelte a szervert handshake kérések sokaságával. Mivel a szerver minden handshake protokoll alatt egy új random értéket kell meghatározzon és annak segitségével új kulcsokat kell generáljon, így a sok handshake kérés nagyon leterheli a szervert.