Protokol TLS, zabezpečené připojení

Pozor, útok!(3. díl)

V rámci našeho seriálu o bezpečnosti se tentokrát zaměříme na zabezpečené připojení a na protokol TLS, který umožňuje bezpečné finanční transakce i zabezpečenou výměnu komerčních dat v prostředí internetu.

Úvod

V předchozích částech seriálu věnovaného zabezpečení přenosu dat přes internet jsme se dozvěděli, že HTTP protokol neposkytuje potřebné bezpečí. Jakýkoliv programátor, který má potřebné znalosti v oblasti počítačových sítí a volný přístup k těmto sítím, totiž může při troše snahy prohlížet či modifikovat přenášená data vzájemně komunikujících účastníků spojení.

Jednou z reakcí na tuto nemilou vlastnost HTTP transakcí byl zrod několika bezpečnostních protokolů, jejichž cílem bylo zajistit bezpečnost takto ohrožených dat.

Minule jsme se seznámili se základními informacemi o protokolu S-HTTP – o protokolu, který na širší využití a rozšíření teprve čeká. V dnešním dílu si naopak povíme o relativně novém protokolu TLS (Transport Layer Security), který je podobně jako protokol SSL (Secure Socket Layer) podporován většinou současných prohlížečů.

Co je TLS?

Primárním cílem TLS protokolu je umožnit komunikujícím aplikacím soukromé spojení a integritu přenášených dat.

TLS ve verzi 1.0 a SSL ve verzi 3.0 jsou v podstatě shodné protokoly. Novější TLS totiž vychází z protokolu SSL a obsahuje menší změny, které do něj byly zakomponovány. Nyní, když už jsme získali obecné znalosti o TLS, tedy konečně nahlédněme “pod pokrývku” vývojářům tohoto protokolu (IETF – viz infotipy) prozkoumáním jeho struktury.

Architektura TLS

Protokol TLS se skládá ze dvou oddělených vrstev: TLS Record Protocol a TLS Handshake Protocol. Na nižší úrovni, ležící na vhodném transportním protokolu – např. na TCP (Transmission Control Protocolu) – se nachází TLS Record Protocol (viz obrázek).

Chceme-li se o těchto dvou vrstvách dozvědět více informací a pochopit jejich funkce, je vhodné se nejprve seznámit s jejich charakteristickými vlastnostmi.

Veškerá komunikace (včetně TLS Handshake Protocolu a datových zpráv) probíhá pomocí nižší vrstvy – TLS Record Protocolu – a je navržena takovým způsobem, aby bylo navázané spojení **spojením soukromým**. To je zajištěno pomocí symetrického šifrování – data jsou zašifrována například pomocí DES. **Poznámka:** Existuje i možnost nepoužít šifrování pro Record Protocol.

Spolehlivost spojení je zajištěna kontrolou integrity zprávy pomocí MAC (Message Authentication Code), což je v podstatě kontrolní součet (checksum), který je odvozen z aplikovaného ověřovacího schématu a klíče na danou zprávu. Pro výpočet MAC jsou v tomto případě využívány bezpečné hashovací funkce, například MD5.

Je tedy zřejmé, že TLS Record Protocol je používán pro zapouzdření různých výše položených vrstev protokolů. Stejně tak je pomocí TLS Record Protocolu zapouzdřena i vyšší vrstva – TLS Handshake Protocol –, jejímž cílem je umožnit klientu a serveru jak vzájemnou autentizaci, tak vzájemnou domluvu na použití šifrovacích mechanismů ještě předtím, než jsou přenášeny první bajty dat.

Ustavení každého nového sezení – “session” – proběhne nejdříve pomocí úvodní výměny informací – “handshake” (potřesení rukou). Abychom si více objasnili poslání TLS Handshake Protocolu, vyjmenujeme si stejně jako u předchozí vrstvy jeho charakteristické vlastnosti.

První vlastností je skutečnost, že **identita uživatelů** (klientu a serveru) může být ověřena užitím asymetrického šifrování, např. RSA. Zjištění identity může být volitelné, ale obecně je požadováno alespoň u jednoho účastníka spojení. Další důležitou vlastností je, že ověřené **spojení nemůže být přivlastněno** tzv. slídilem ani v případě, že se nachází ve středu spojení.

Poslední významnou vlastností je skutečnost, že **ověření je spolehlivé**. Tato vlastnost je zajištěna detekcí jakéhokoliv pokusu o modifikaci ověřovacího spojení následným upozorněním účastníků spojení.

Průběh transakce

Jelikož je navázání bezpečného spojení podobné jako u SSL (1. díl seriálu), je komunikace mezi klientem a serverem v následujícím textu popsána zjednodušenou formou, která však obsahuje všechny podstatné informace o průběhu této úvodní transakce.

V případě, že se prohlížeč (= klient) snaží připojit k zabezpečenému serveru, nejdříve mu pošle zprávu zvanou “ClientHello”, což je obdoba žádosti na zřízení HTTP spojení. V okamžiku, kdy server zprávu ClientHello přijme, zpracuje informace v ní obsažené.

Pro lepší pochopení obsahu zprávy ClientHello je v následujícím textu uvedena její programová struktura:

```
enum { null(0), (255) } CompressionMethod;
struct {
ProtocolVersion client_version;
Random random;
SessionID session_id;
CipherSuite cipher_suites<2..2^16-1>;
CompressionMethod compression_methods<1..2^8-1>;
} ClientHello;
```

Pokud je vše v pořádku, zašle server klientu zprávu zvanou “ServerHello”, která zpravidla obsahuje certifikát, údaje o klíčích serveru a volitelně požadavek certifikace klientu.

Struktura zprávy ServerHello a certifikátu je následující:

```
struct {
ProtocolVersion server_version;
Random random;
SessionID session_id;
CipherSuite cipher_suite;
CompressionMethod compression_method;
} ServerHello;
opaque ASN.1Cert<1..2^24-1>;
struct {
ASN.1Cert certificate_list<0..2^24-1>;
} Certificate;
```

Poté, co klient obdrží zprávu ServerHello, zašle serveru údaje o svých klíčích, ověří pravost certifikátu serveru a v případě neshody ukončí spojení. Byl-li požadován jeho certifikát, zašle jej spolu s výběrem číslic, pomocí něhož je vytvořen tzv. klíč relace. Po obdržení této zprávy vytvoří server ze seznamu číslic klíč relace. Pak klient a server přenášejí data, která jsou zašifrována pomocí tohoto klíče relace.

Poznámka: Tvorba klíče relace může proběhnout jak na straně klientu, tak na straně serveru.

A jak se dozvíme, zda jsme připojeni na bezpečný server? Stačí se podívat na URL serveru, a pokud začíná **https://** (tedy stejně jako v případě SSL), můžeme být o něco klidnější, protože naše komunikace přes port 443 probíhá bezpečně.

Kompatibilita SSL a TLS

Jak již bylo zmíněno, oba protokoly jsou si velice podobné. Z historických důvodů, aby se zabránilo rozmařilé spotřebě rezervovaných čísel portů, užívají TLS 1.0, SSL 3.0 a SSL 2.0 stejný spojovací port. Komunikace klientu a serveru užívající protokoly ve verzích TLS 1.0 a SSL 3.0 je stručně popsána v následujícím textu.

Pokud se chce klient TLS 1.0 dohodnout se serverem SSL 3.0, pošle serveru zprávu ClientHello užitím record formátu SSL 3.0. Pokud server podporuje pouze SSL 3.0, odpoví SSL 3.0 ServerHello. Podporuje-li navíc i TLS, odpoví pomocí TLS ServerHello.

Podobně pokud TLS server komunikuje se SSL klientem, akceptuje ClientHello zprávu ve formátu SSL 3.0 a odpoví ServerHello také ve formátu SSL.

Cíle TLS

Po objasnění funkcí TLS protokolu si již můžeme říci, proč byl vlastně tento protokol navržen a jaké jsou jeho hlavní cíle. Nejdůležitějším cílem je zajistit ustavení bezpečného spojení mezi účastníky komunikačního procesu. Dalším cílem je umožnit programátorům vytvářet aplikace využívající vlastností TLS bez další znalosti šifrovacích mechanismů.

Protokol TLS byl navržen s ohledem na neustálý vývoj metod zvyšujících bezpečnost (šifrovacích i jiných). Umožňuje tedy implementaci aktualizovaných i zcela nových bezpečnostních knihoven. Dalším cílem bylo využít stále rostoucí výpočetní výkon CPU zejména pro operace s veřejnými klíči a urychlit tak proces výměny informací obsažených ve zprávách.

Závěr

TLS zahrnuje mechanismy pro zabezpečení finančních transakcí i pro bezpečnou výměnu komerčních dat přes web. Je tedy dalším protokolem, jehož pomocí se můžeme svěřit všeobjímajícímu internetu bez pocitu strachu o bezpečnost privátních dat.

Příště si povíme o dalších způsobech, jak zvýšit bezpečnost svých dat na internetu.

Ing. Milan Pinte (pinte@kpv.zcu.cz)