**Krótko o SSL / TLS**

Protokół SSL (i jego następca, TLS) – to protokół, który działa bezpośrednio na połączeniu TCP (chociaż istnieją implementacje protokołów na podstawie datagramów, takich jak UDP). W ten sposób, protokoły wyższego poziomu (np. HTTP) mogą zostać bez zmian zapewniając bezpieczne połączenie. Pod warstwą SSL – HTTP wygląda identycznie jak protokół HTTPS.

Podczas poprawnego korzystania z protokołu SSL/TLS, wszystkie osoby atakujące mogą zobaczyć adres IP i port, do którego podłączone są, oraz ile danych zostało przesłano i czy były dane zaszyfrowanie i skompresowane. Ta osoba również może przerwać połączenie, ale obie strony (serwer i klient) będą wiedzieć, że połączenie zostało przerwane przez osobę trzecią.

Standardowo osoba atakująca jest w stanie dowiedzieć się, jaka jest nazwa hosta, z którym się łączysz (ale pozostałą część adresu URL już nie): chociaż sam HTTPS nie naraża nazwę hosta na niebezpieczeństwo, po prostu przeglądarka musi zrobić zapytanie do serwerów DNS, aby dowiedzieć się na jaki adres IP wysłać zapytanie.

**Opis działania protokołu SSL**

**Krok 1.** Po utworzeniu połączenia TCP uruchamia się SSL handshake poprzez klienta. Klient (który może być przeglądarką, a także innym programem, takim jak Windows Update lub PuTTY) wysyła szereg specyfikacji:  jaka wersja protokołu SSL czy TLS jest uruchomiona,  jakich szyfrów chce wykorzystać połączenie, i jakie metody kompresji chce wykorzystać klient.

**Krok 2.** Serwer sprawdza najwyższą wersję protokółu SSL/TLS którą mogą obsłużyć obie strony, wybiera jedną z możliwych opcji szyfrowania u klienta (jeśli obsługuje), i w razie potrzeby wybiera metodę kompresji.

**Krok 3.** Po tej podstawowej konfiguracji serwer wysyła swój certyfikat. Certyfikat ten musi być zaufany albo poprzez samego klienta albo stronę trzecią, któremu ufa klient. Na przykład, jeżeli klient ufa GeoTrust to on może również zaufać certyfikatowi Google.com, ponieważ certyfikat ze strony Google.com jest kryptograficznie podpisany firmą GeoTrust.

**Krok 4.** Po potwierdzeniu że certyfikat jest ważny i serwer jest naprawdę tym, jakim się wydaje (a nie „człowiekiem po środku”), odbywa się wymiana kluczami. To może być klucz publiczny, „PreMasterSecret” albo po prostu nic, w zależności od wybranej metody szyfrowania. Zarówno serwer, jak i klient teraz może określić klucz do symetrycznego szyfrowania. Klient informuje serwer, że od teraz cały ruch będzie zaszyfrowany i wysyła te zaszyfrowane i podpisane żądania na serwer.

**Krok 5.** Serwer sprawdza, czy adres MAC (służy do uwierzytelniania) jest poprawny i czy ta wiadomość/żądanie może być poprawnie rozszyfrowane. Następnie serwer zwraca komunikat, który sprawdza się przez klienta.

Handshake w tej chwili się skończył i dwa urządzenia mogą bezpiecznie się komunikować pomiędzy sobą. Więcej informacji można znaleźć w sekcji technet.microsoft.com/en-us/library/cc785811 i pl.wikipedia.org/wiki/Transport\_Layer\_Security.

Aby zamknąć połączenie, wysyła się powiadomienie close\_notify. Jeśli hacker spróbuje przerwać połączenie zakończeniem połączenia TCP (dodaniem pakietu FIN), obie strony będą wiedzieć, że połączenie zostało poprawnie zamknięte. Połączenie nie może być uszkodzone, ale może być przerwane.