|  |
| --- |
| Politechnika Wrocławska, Informatyka Stosowana |
| Ochrona komunikacji |
| Cyberbezpieczeństwo, Laboratorium nr.9 - raport |

|  |
| --- |
| Autor: Aleksander Stepaniuk  Nr. Indeksu: 272644 |

**1. Zadania i pytania (TLS)**

**Zadania 1.1-1.3;**

Przebieg zadań na zrzutach ekranu poniżej:

**Pytanie 1.4;**

Zasadniczo tak, oba protokoły służą do szyfrowania danych i uwierzytelniania stron w celu

zapewnienia bezpieczeństwa połączenia internetowego. Zasada działania SSL (Secure

Sockets Layer) oraz TLS (Transport Layer Security) jest taka sama – oba protokoły

wykorzystują szyfrowanie przesyłanych danych, aby ochronić je przed odczytaniem przez

niepożądane osoby. TLS jest jedynie nowocześniejszą i ulepszoną wersją protokołu SSL,

uzupełniającą jego luki w zabezpieczeniach – przy szyfrowaniu korzysta z silniejszych

algorytmów oraz minimalnie różni się procesem uzgadniania (handshake).

* Head

Nie, TLS jest następcą SSL. TLS wprowadza ulepszenia w zakresie bezpieczeństwa i wydajności.

**Pytanie 1.5;**

Jest to proces uzgadniania parametrów używanych do nawiązywania połączenia SSL/TLS

pomiędzy dwoma urządzeniami lub aplikacjami, np. serwerem i przeglądarką internetową.

Podczas tego procesu następuje wymiana informacji, takich jak wersja protokołu,

certyfikaty szyfrujące czy użyty algorytm. Jest to niezbędne, aby bezpiecznie przesyłać

dane za pomocą zaszyfrowanego połączenia.

* Head

To proces negocjacji między klientem a serwerem, w którym ustalane są szczegóły połączenia (np. algorytmy szyfrowania, klucze sesji).

**Pytanie 1.6;**

Protokół TLS/SSL zapewnia skuteczną ochronę transmisji poszczególnych danych poprzez

ich szyfrowanie i uwierzytelnianie stron. Dzięki temu możliwe jest bezpieczne przesyłanie

danych poufnych, takich jak dane osobowe, dane logowania czy numery kart kredytowych,

bez obaw o ujawnienie ich niepowołanym osobom.

Przykładowe zastosowania:

* zabezpieczanie połączeń z serwerem bankowym podczas logowania do konta bankowego online,
* szyfrowanie połączenia pomiędzy aplikacją mobilną a serwerem podczas przesyłania danych osobowych, np. przy rejestracji w serwisach, takich jak YouTube,
* bezpieczne połączenia z serwerem sieci VPN (Virtual Private Network) w celu ochrony prywatności podczas przeglądania stron internetowych,
* zaszyfrowanie połączenia pomiędzy aplikacją do prowadzenia połączeń głosowych lub wideo a serwerem, aby je zabezpieczyć i ochronić poufne dane przed potencjalnymi intruzami.

🡪   
Zapewnia:

* Szyfrowanie (ochrona danych przed podsłuchem)
* Integralność danych (ochrona przed ich modyfikacją)
* Uwierzytelnienie stron.  
  Przykłady: HTTPS (strony internetowe), SMTP (szyfrowanie e-maili).

**Pytanie 1.7;**

Obecnie najczęściej stosowane są wersje TLS 1.3 i TLS 1.2. Wersja 1.3 jest uznawana

obecnie jako standard i używana powszechnie w większości systemów. W niektórych

przypadkach można jeszcze spotkać starsze wersje protokołu, takie jak TLS 1.1 lub TLS

1.0, jednak jest to bardzo rzadkie i uważane za przestarzałe.

🡪  
TLS 1.2 i TLS 1.3.

**Pytanie 1.8;**

Wersja 1.3 jest uważana za bezpieczniejszą ze względu na użycie lepszych mechanizmów

uwierzytelniania i szyfrowania. W TLS 1.3 szyfrowanie obejmuje większą część procesu

handshake’u niż w TLS 1.2 oraz stosowane są nowsze, ulepszone metody szyfrowania

i uwierzytelnienia (podpisu cyfrowego), które łatają luki w bezpieczeństwie przesyłania

danych.

🡪

TLS 1.3, ponieważ eliminuje przestarzałe algorytmy, redukuje liczbę podatnych funkcji i usprawnia proces handshake.

**Pytanie 1.9;**

Pod tym względem również lepsza jest wersja TLS 1.3, ponieważ korzysta

z wydajniejszych algorytmów szyfrowania i uwierzytelniania. TLS 1.3 wykorzystuje

głównie algorytm AES, ze względu na szybszy proces szyfrowania, podczas gdy TLS 1.2

korzysta z dużo wolniejszego, asymetrycznego RSA. W TLS 1.3 skrócono również proces

handshake’u (wyeliminowano wszystkie wymiany kluczy niekorzystające z protokołu

Diffiego-Hellmana), dzięki czemu proces ten przebiega dużo szybciej i zużywa mniej

energii, co jest istotne m.in. w systemach wbudowanych czy Internetu Rzeczy. Należy

jednak pamiętać, że aby korzystać z nowszych wersji protokołu TLS, obie strony połączenia

muszą obsługiwać tę samą wersję protokołu.

🡪

TLS 1.3, dzięki krótszemu handshake (1-RTT zamiast 2-RTT).

**Pytanie 1.10;**

Program Wireshark zarejestrował dwie wersje protokołu: TLS 1.2 oraz TLS 1.3

🡪

Odpowiedź zależy od analizy przechwyconego ruchu w Wireshark.

**Pytanie 1.11;**

Są to wiadomości:

* ClientHello – klient wysyła do serwera parametry potrzebne do nawiązania połączenia przez protokół TLS (m.in. obsługiwana wersja protokołu, algorytmu szyfrującego i kompresji danych),
* ServerHello – serwer odpowiada podobnym komunikatem, w którym zwraca klientowi wybrane parametry połączenia,
* Certificate – serwer wysyła swój certyfikat, umożliwiając klientowi sprawdzenie swojej tożsamości
* ServerKeyExchange – serwer wysyła informację o swoim kluczu publicznym, którego rodzaj i długość określony jest przez typ algorytmu przesłany w poprzednim komunikacie,
* ServerHelloDone – serwer zawiadamia jest gotowy na odpowiedź klienta
* ClientKeyExchange – klient wysyła serwerowi wstępny klucz sesji, zaszyfrowany za pomocą klucza publicznego serwera. Na podstawie ustalonych w poprzednich komunikatach parametrów obie strony generują klucz sesji używany do wymiany danych,
* ChangeCipherSpec – klient zawiadamia, że serwer może rozpocząć komunikację szyfrowaną za pomocą parametrów ustalonych w poprzednich komunikatach
* Finished – klient zawiadamia, że jest gotowy do odbierania danych w postaci zaszyfrowanej,
* ChangeCipherSpec – serwer powiadamia, że od tej pory wysyłać będzie tylko zaszyfrowane dane,
* Finished – komunikat o pomyślnym procesie handshake’u, wysyłany bezpiecznym kanałem.

🡪

Zwykle z: *ClientHello*, *ServerHello*, wymiany certyfikatów i kluczy sesji.

**Pytanie 1.12;**

Wersja protokołu TLS, która jest używana, zależy od wersji oprogramowania, które jest

zainstalowane na obu komputerach uczestniczących w komunikacji. Proces komunikacji

przy pomocy TLS przebiegnie pomyślnie tylko wtedy, gdy obie strony poprawnie uzgodnią

wersję protokołu używaną do przesyłania informacji – ustalane jest to w procesie

handshake’u. Jeśli jedna ze stron używa wersji, która obsługuje „niższą”, wcześniejszą

wersję protokołu TLS, wówczas połączenie będzie nawiązane przy użyciu starszej wersji

protokołu. Dlatego ważne jest, by regularnie aktualizować oprogramowanie, po to aby mieć

dostęp do najnowszych i najbezpieczniejszych wersji protokołu TLS.

🡪

Od wsparcia po stronie klienta i serwera oraz konfiguracji ich oprogramowania.

**2. Zadania (OpenVPN)**

**Zadania 2.1-2.4;**

Przebieg zadań na zrzutach ekranu poniżej:

**3. Zadania i pytania (OpenVPN)**

**Zadania 3.1-3.4;**

Przebieg zadań na zrzutach ekranu poniżej:

**Pytanie 3.5;**

Ataki

**Pytanie 3.6;**

Ataki

**Pytanie 3.7;**

Ataki

**4. Zadania i pytania**

**Zadania 4.1-4.3;**

Przebieg zadań na zrzutach ekranu poniżej:

**Pytanie 4.4;**

Ataki

**Pytanie 4.5;**

Ataki

**Pytanie 4.6;**

Ataki