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

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

**0. Konfiguracja VPN**

**Konfiguracja;**

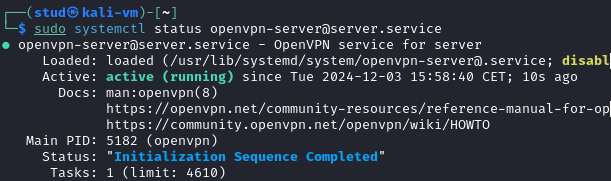
Przebieg konfiguracji na zrzutach poniżej:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie



Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, czarne i białe

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
  
**1. Zadania i pytania (TLS)**

**Zadania 1.1-1.3;**

Przebieg zadań na zrzutach ekranu poniżej:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

**Pytanie 1.4;**

Oba protokoły działają w głównej mierze tak samo (służą do szyfrowania i uwierzytelniania danych, aby chronić je przed odczytem przez osoby niepożądane), z tą różnicą że TLS jest następcą SSL, a więc wprowadza różnego rodzaju ulepszenia, między innymi w zakresie bezpieczeństwa i wydajności działania protokołu.

**Pytanie 1.5;**

TLS/SSL handshake to process negocjacji parametrów używanych do nawiązywania połączenia między klientem a serwerem. Podczas tego procesu następuje wymiana szczegółów połączenia takich jak wersja protokołu, klucze sesji czy użyte algorytmy szyfrujące. Jest niezbędnym elementem bezpiecznego przesyłania danych zaszyfrowaną drogą.

**Pytanie 1.6;**

Protokół TLS/SSL zapewnia ochronę transmisji danych poprzez ich szyfrowanie oraz uwierzytelnianie stron komunikacji. Zastosowania:

* Szyfrowanie połączenia między klientem a serwerem podczas przesyłania danych logowania przy rejestracji do wybranego serwisu (np. Netflix)
* Szyfrowanie połączenia między klientem a serwerem podczas prowadzenia połączenia głosowego przez aplikację typu Zoom (ochrona danych przed podsłuchem)
* Uwierzytelnianie stron internetowych HTTPS (np. facebook.com)
* Szyfrowanie e-maili (SMTP)
* Bezpieczne połączenie z serwerem VPN (ochrona prywatności)

**Pytanie 1.7;**  
TLS 1.2 i TLS 1.3. Wersja 1.3 uznawana jest obecnie jako standard i jest powszechnie używana w większości systemów.

**Pytanie 1.8;**

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

**Pytanie 1.9;**

TLS 1.3, ponieważ korzysta z wydajniejszych algorytmów oraz sam proces przebiega szybciej (oraz zużywa mniej energii) ze względu na:

* 1-RTT zamiast 2-RTT
* AES zamiast asymetrycznego RSA
* Krótszy handshake (każda wymiana kluczy korzysta z Diffiego-Hellmana)

**Pytanie 1.10;**

Wireshark przechwycił obydwie wersje protokołu: TLS 1.2 oraz TLS 1.3

**Pytanie 1.11;**

Zwykle z: *ClientHello*, *ServerHello*, wymiany certyfikatów i kluczy sesji. Proces ten różni się w zależności od wersji protokołu:

* **TLS 1.2**:
  1. *ClientHello*: Klient wysyła informacje o obsługiwanych algorytmach szyfrowania, wersji protokołu i danych do wygenerowania klucza.
  2. *ServerHello*: Serwer odpowiada wyborem algorytmu i wersji TLS.
  3. *Certificate*: Serwer wysyła swój certyfikat do uwierzytelnienia.
  4. *Key Exchange*: Klient i serwer wymieniają dane pozwalające na ustalenie wspólnego klucza sesji.
  5. *Finished*: Obie strony kończą proces, potwierdzając poprawność ustaleń.
* **TLS 1.3**:
  1. *ClientHello*: Klient wysyła propozycję algorytmów szyfrowania i dane do wygenerowania klucza (w tym dane dla szyfrowania asymetrycznego).
  2. *ServerHello*: Serwer akceptuje algorytm i generuje klucz sesji na podstawie danych klienta. Certyfikat serwera jest przesyłany w jednym kroku, skracając czas wymiany.
  3. *Finished*: Po weryfikacji certyfikatu połączenie jest szyfrowane, co zmniejsza liczbę rund komunikacji.

**Pytanie 1.12;**

Zależy od wersji oprogramowania zarówno po stronie klienta jak i serwera (wsparcie systemowe oraz konfiguracja oprogramowania mają znaczenie). Z tego powodu warto regularnie aktualizować oprogramowanie aby mieć dostęp do najnowszych, najszybszych i najbezpieczniejszych wersji protokołów takich jak TLS. Proces komunikacji przebiegać będzie po najnowszej wspólnej wersji protokołu uzgodnionej w handshaku (obie strony muszą obsługiwać daną wersję). Dodatkowo administrator sieci może wymusić użycie konkretnej wersji protokołu. Niektóre przeglądarki lub aplikacje pozwalają na wymuszenie używania nowszych wersji. Serwery proxy lub urządzenia pośredniczące mogą wymuszać starsze wersje w przypadku braku wsparcia.

**2. Zadania (OpenVPN)**

**Zadania 2.1-2.4;**

Przebieg zadań na zrzutach ekranu poniżej:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Dla open vpn (tun0) udało się wysłać wiadomość przez netcat, przechwycić pakiety TCP przesyłane między maszynami, dało się odczytać zawartość wiadomości z pakietu

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

**Zadania 3.1-3.4;**

Przebieg zadań na zrzutach ekranu poniżej:

Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, Strona internetowa

Opis wygenerowany automatycznie

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, zrzut ekranu

Opis wygenerowany automatycznie

Dla enp0s3 ramki OpenVPN przesyłane pomiędzy klientem a serwerem mają niewidoczny adres IP adresata oraz nie można było odczytać przesyłanej wiadomości w wireshark.

**Pytanie 3.5;**

Tak, jeśli wiadomość została przesłana w postaci nieszyfrowanej (dla ruchu z interfejsu sieci VPN), można ją odczytać, przechwytując ruch sieciowy. W przypadku analizy ruchu z interfejsu fizycznego widoczne pakiety OpenVPN przesyłane między klientami miały postać zaszyfrowaną i wymagały klucza dostępu.

**Pytanie 3.6;**

Dla ruchu na interfejsie hosta widoczny jest zarówno adres źródłowy jak i docelowy (jeśli nie użyto szyfrowania aplikacyjnego, np. TLS), natomiast na interfejsie sieci VPN pakiety posiadały jedynie adres klienta, który wysłał wiadomość oraz adres serwera, a więc odbiorca wiadomości nie mógł w tym przypadku sprawdzić z jakiego adresu wysłano wiadomość.

**Pytanie 3.7;**

Dla interfejsu hosta widać szczegóły transmisji (adresy IP źródłowy i docelowy, porty, dane aplikacyjne). Na interfejsie VPN widoczne są tylko zaszyfrowane pakiety i dane typu metadane (np. adres serwera VPN).

**4. Zadania i pytania**

**Zadania 4.1-4.2;**

Przebieg zadań na zrzutach ekranu poniżej:

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Zadanie 4.3;**

* Ustawienia 1: 27,73 sekund
* Ustawienia 2: 35,69 sekund
* Ustawienia 3: 42,11 sekund
* Ustawienia 4: 39,19 sekund

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Pytanie 4.4;**

Tak, szyfrowanie i dodatkowe trasy przez serwer VPN zwiększają opóźnienia przy przesyle pliku, jednak sama różnica jest raczej na tyle niewielka, że można założyć że wybór algorytmu nie ma większego znaczenia przy czasie przesyłania pliku. Te różnice mogą być spowodowane wieloma innymi czynnikami (jak to, że im dłużej pracuje się na wirtualnych maszynach tym bardziej zaczyna im brakować ramu i po godzinie pracy potrafią się zacinać przy 3 wirtualnych środowiskach (działają wolniej)).

**Pytanie 4.5;**

Algorytmy silne (np. AES-256-GCM) zapewniają lepsze bezpieczeństwo, ale mogą być wolniejsze w użytku. DES-EDE-CBC jest uznawany za bezpieczny, jednak niezalecany, bo zaleca się korzystanie z lepszego w każdym względzie AESa. Algorytm MD5 jest zdecydowanie przestarzały i nie gwarantuje współcześnie żadnego bezpieczeństwa w związku z tym nie zaleca się korzystanie z niego.

**Pytanie 4.6;**

Za najgorszy zestaw można uznać ten z algorytmem DES-CBC oraz MD5, ponieważ nie korzysta się z tych algorytmów we współczesnej kryptografii (posiadają wady pozwalające je prościej złamać) i zalecane jest zastąpienie je przez współczesne metody takie jak AES czy SHA256. Z kolei za najbezpieczniejszy zestaw można z kolei uznać użycie AES-256-GCM oraz SHA512, ponieważ długości klucza są długie i zapewniają bezpieczne szyfrowanie danych.