Krzysztof DUZINKIEWICZ
URZĄD KOMUNIKACJI ELEKTRONICZNEJ DELEGATURA W ZIELONEJ GÓRZE

Wirtualna Sieć Prywatna

Mgr Krzysztof DUZINKIEWICZ

Absolwent Wydziału Matematyki Uniwersytetu Zielonogórskiego oraz studiów podyplomowych na Wydziale Elektrotechniki Informatyki i Telekomunikacji UZ. Pracownik Urzędu Komunikacji Elektronicznej Delegatury w Zielonej Górze na stanowisku Starszy specjalista w zakresie administracji informatycznych zasobów lokalnych, badań w ramach kontroli rynku wyrobów oraz kontroli przedsiębiorców telekomunikacyjnych.

e-mail: k.duzinkiewicz@uke.gov.pl

Streszczenie

W artykule przedstawiono technologie wirtualnych sieci prywatnych (VPN) na przykładzie oprogramowania OpenVPN. Opisano instalację, konfigurację serwera VPN zaimplementowanego na linuksowym routerze oraz klientów pod systemem operacyjnym Windows. Opisano aspekty związane z bezpieczeństwem przesyłania otwartym kanałem zaszyfrowanych informacji i zasady działania protokołu SSL

Słowa kluczowe: VPN, SSL, algorytm DH, CA.

Virtual Private Network

Abstract

The technologies of virtual private nets (VPN) enabling the safe access to supplies and corporate applications from home unprotected internet nets on the example of software OpenVPN that are based on the open (GPL) General Public License are presented in the paper. In Section 2 there are described both the principle of SSL protocol and the theoretical bases of coding the symmetrical and asymmetrical connection called the mixed coding. The Diffiego-Hellman's algorithm is also introduced. It is responsible for save distribution of private keys by an open channel. There is presented the mechanism called a reliable institution of the keys storage - certificate authenticity. Section 3 is concentrated on the practical use of the technology called VPN. It presents the installation of free software - OpenVPN. Further on, the way of generating keys and certificates is introduced as well as the server and clients' configuration. The VPN server role is functioned by Linksys WRT54GL router where the original firmware is replaced by an alternative firmware called dd-wrt.v24_vpn. However, the VPN client works under the control of the Windows system. Finally, the topology of VPN net (tunnels) based on real distracted net is described.

Keywords: Virtual Private Network, Secure Socket Layer, algorithm Diffiego-Hellmana, Certificate Authority.

1. Wstęp

Technologia VPN – Wirtualna Sieć Prywatna (*ang. Virtual Private Network*) w założeniu powstała w celu umożliwienia bezpiecznego dostępu do zasobów i aplikacji korporacji pracownikom korzystającym z domowych niezabezpieczonych sieci internetowych [1]. Połączenie VPN określa się również jako wirtualny tunel łączący komputery, sieci lokalne LAN za pośrednictwem publicznych sieci internetowych. Przesyłane dane w ramach tunelu są szyfrowane za pomocą algorytmów szyfrujących w celu zwiększenia bezpieczeństwa. Sieć VPN istnieje tylko jako struktura logiczna w ramach sieci publicznych, ale stacje końcowe w ramach połączenia tunelowego działają tak, jakby były w jednej sieci lokalnej.

2. Podstawy teoretyczne

2.1. OpenVPN

Jednym z rozwiązań wirtualnych sieci prywatnych jest projekt OpenVPN stworzony przez Jamesa Yonana w ramach wolnego oprogramowania opartego na licencji GPL (*ang. General Public License*), oprogramowanie jest darmowe a kod programu otwarty [2].

Projekt jest dostępny dla wielu systemów operacyjnych i pod różne platformy, miedzy innymi również system operacyjny Windows XP. Cały pakiet składa się z jednego kodu binarnego dla klienta i serwera, pliku konfiguracyjnego, plików kluczy i certyfikatów. Pod względem topologii wirtualnej sieci pakiet OpenVPN oferuje dwa typy konfiguracji: tryb router - połączenie jeden do jednego (klient-serwer) oraz tryb bridge - połączenie wielu klientów do jednego serwera. Pakiet OpenVPN do szyfrowania połączeń nie wykorzystuje protokołów IPSec, ale korzysta z biblioteki OpenSSL opartej na protokole TLS (*ang. Transport Layer Security*) mającym na celu zapewnienie poufności i integralności transmisji danych. Protokół TLS przyjęty został jako standard w Internecie i jest rozwinięciem protokołu SSL (*ang. Secure Socket Layer*) stworzonego do bezpiecznego połączenia. Biblioteka OpenSSL zawiera mechanizmy kryptograficzne, a także zestaw narzędzi do generowania kluczy i certyfikatów.

2.2. Protokół SSL

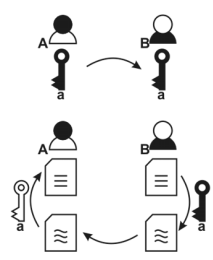
Algorytmy szyfrujące możemy podzielić ze względu na sposób użycia kluczy na dwa podstawowe typy [3]:

2.2.1. Szyfrowanie symetryczne

Algorytm posługujący się tym samym kluczem przy szyfrowaniu i deszyfrowaniu informacji. Jest to metoda szyfrowania znana od dawna i polega na uzgodnieniu przez strony tajnego klucza szyfrującego (schematu kodowania). Współcześnie tego typu metody kryptograficzne są bardziej zawansowane i używają skomplikowanych operacji bitowych; zaletą jest ich szybkość (w małym stopniu zmniejszają przepustowość kanału transmisyjnego), ale niewątpliwie wadą jest problem dystrybucji klucza prywatnego.

2.2.2. Szyfrowanie asymetryczne

Używana jest para kluczy - klucz publiczny jawny oraz klucz prywatny znany tylko właścicielowi (rys.1). Klucze te generowane są algorytmem, który opiera się na problemie faktoryzacji dużych liczb pierwszych, gdzie rozłożenie na czynniki iloczynu dużych liczb jest stosunkowo trudne obliczeniowo.



Użytkownik A generuje dwa klucze – prywatny i publiczny, klucz publiczny wysyła jawnym kanałem do użytkownika B.

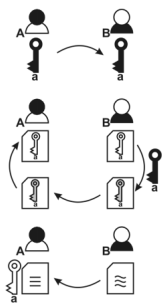
Użytkownik B wysyła zaszyfrowaną informację do użytkownika A za pomocą klucza publicznego otrzymanego od niego wcześniej.
Użytkownik A deszyfruje zakodowaną informację własnym kluczem prywatnym.

Rys. 1. Poglądowy schemat kryptografii asymetrycznej (źródło: autor)
Fig. 1. Diagram of asymmetric cryptography (source: the author)

Rozwiązany zostaje problem dystrybucji kluczy, ale kodowanie i dekodowanie informacji jest złożone i wymaga dłuższych czasów do przeprowadzenia obliczeń, niż w przypadku kryptografii symetrycznej.

2.2.3. Szyfrowanie mieszane

Jest to metoda kryptograficzna, która łączy zalety obu metod kryptografii symetrycznej i asymetrycznej (rys.2). Stosując odpowiednio długie klucze można zwiększyć dowolnie bezpieczeństwo zaszyfrowanych informacji algorytmami symetrycznymi, ale pozostaje w dalszym ciągu problem uzgodnienia kluczy, dlatego opracowano bezpieczny sposób ich dystrybucji. Przedstawiona metoda kryptograficzna sprowadza się do bezpiecznego uzgodnienia, dystrybucji klucza prywatnego i dalej przesyłania informacji zaszyfrowanych algorytmem symetrycznym.



Użytkownik A dystrybuuje klucz publiczny, który otrzymuje użytkownik B, proces może być odwrotny, istotny jest fakt uzgodnienia klucza publicznego przez obie strony.

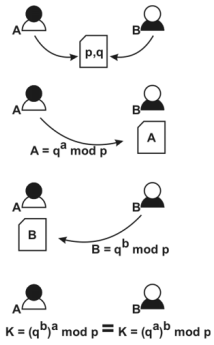
Użytkownik B wysyła swój klucz prywatny zakodowany uzgodnionym wcześniej kluczem publicznym i wysyła do użytkownika A

Użytkownicy dalej przesyłają informacje zakodowane algorytmem symetrycznym korzystając z klucza prywatnego przesłanego wcześniej bezpiecznym algorytmem asymetrycznym.

Rys. 2. Poglądowy schemat kryptografii mieszanej (źródło: autor)
Fig. 2. Diagram of mixed cryptography (source: the author)

2.2.5. Protokół Diffiego-Hellmana (DH)

Nazwa protokołu pochodzi od nazwisk twórców algorytmu (rys.3), powstał w celu rozwiązania problemu uzgodnienia i bezpiecznej dystrybucji kluczy prywatnych przez jawny kanał.



Użytkownicy A i B ustalają dwie liczby pierwsze, stosunkowo dużą liczbę p i liczbę q

Użytkownik A wybiera losowo dowolną liczbę pierwszą a, dokonuje obliczeń i wynik przesyła użytkownikowi B

Użytkownik B wybiera losowo dowolną liczbę pierwszą b, dokonuje obliczeń i wynik przesyła użytkownikowi A

Otrzymana po przeliczeniu wartość K przez obu użytkowników niezależnie jest uzgodnionym kluczem.

Rys. 3. Poglądowy schemat algorytmu DH (źródło: autor)
Fig. 3. Diagram of cryptography DH (source: the author)

Analizując przedstawiony przykład znalezienie wartości losowo wybieranych liczb pierwszych a lub b stosunkowo długich (1024 bitów), które nie są ujawniane i znając pozostałe parametry przesyłane otwartym kanałem nie jesteśmy w stanie dysponując dzisiejszą mocą obliczeniową komputerów wyznaczyć tych liczb – jest to tzw. dyskretny problem logarytmiczny Diffiego-Hellmana.

2.2.4. Certyfikat autentyczności (CA)

Certyfikat autentyczności (*ang. certificate authority*) rozwiązuje problem autentyfikacji kluczy publicznych, może istnieć jednak obawa ze dystrybuowany jawnym kanałem klucz publiczny może być narażony na ataki, dlatego aby zapewnić integralność został wprowadzony mechanizm przechowujący klucze publiczne – „zaufana instytucja" – baza przechowująca certyfikaty kluczy publicznych; jest to zbiór określonych danych jednoznacznie identyfikujących daną jednostkę (komputer, osobę, siec korporacyjną) i zwiera pola typu:
- nazwę certyfikowanego obiektu,
- identyfikator obiektu,
- klucz publiczny obiektu,
- czas ważności (wygaśnięcia),
- nazwę wystawcy certyfikatu,
- identyfikator wystawcy,
- podpis wystawcy - jednoznaczny skrót całego certyfikatu zaszyfrowany przy pomocy klucza prywatnego wystawcy.

Podsumowując protokół SSL jest implementacją przedstawionych metod szyfrowania, ustandaryzowanym zestawem algorytmów i schematów zapewniających bezpieczny kanał przesyłania danych otwartym kanałem – Internetem, a biblioteka OpenSSL zawiera wszystkie metody, mechanizmy do generowania klucza publicznego, wystawiania certyfikatu poświadczającego ważność klucza publicznego, generowania kluczy prywatnych dla użytkowników biorących udział w sesji, czy w końcu dystrybucję kluczy, autoryzację, nawiązanie połączenia i wymianę danych.

3. Zastosowanie praktyczne OpenVPN

W przedstawionym przykładzie rolę serwera pełni router z systemem linuksowym, natomiast klienci są uruchomieni na komputerach pod systemem Windows.

3.1. Instalacja pakietu OpenVPN pod systemem WinXP

Pakiet openvpn-2.0.9-gui-1.0.3-install.exe można pobrać ze strony projektu http://openvpn.se/download.html. Dystrybucja zawiera bibliotekę OpenSSL, nakładkę graficzną oraz moduł do generowania i zarządzania certyfikatami i kluczami. Wybieramy instalowany opcjonalnie (domyślnie wyłączony) moduł My Certificate Wizard. W trakcie instalacji potwierdzamy instalację wirtualnej karty sieciowej *TAP-Win32 Adapter V8*, w połączeniach sieciowych dodane zostaje nowe połączenie lokalne, którego nazwę zmieniamy na OpenVPN.

3.2. Konfiguracja wstępna OpenSSL

- zmieniamy rozszerzenie plików:
  C:\Program Files\OpenVPN\easy-rsa\openssl.cnf.sample
  na *openssl.cnf*
  .../vars.bat.sample na *vars.bat*
- edytujemy zmienne pliku *vars.bat*:
  set KEY_DIR=keys – katalog główny
  set KEY_SIZE=1024 – długość klucza
  oraz ustawienia regionalne
  set KEY_COUNTRY=PL
  set KEY_PROVINCE=lubuskie
  set KEY_CITY=ZielonaGora, itd.
- tworzymy w katalogu *easy-rsa* katalog *keys*
- tworzymy w tym katalogu dwa pliki *index.txt* i *serial*.
- plik *serial* edytujemy wpisując *00* .
- uruchamiamy z interpretatora poleceń CMD plik *vars.bat*.

3.3. Generowanie kluczy i certyfikatów:

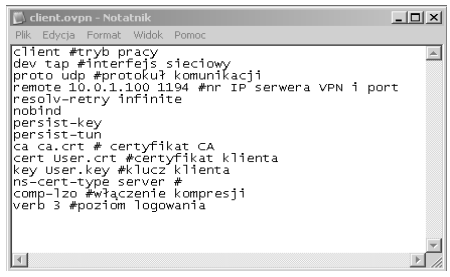
dla serwera
- ca.cert, ca.key,
- uruchamiamy *build-ca.bat* otrzymujemy w katalogu *keys* certyfikat wystawcy oraz klucz publiczny wystawcy,
- dh1024.pem,
- uruchamiamy *build-dh.bat* otrzymujemy klucz DH, protokół do uzgadniania kluczy prywatnych,
- server.cert, server.key,

– uruchamiamy z parametrem *build-key-server.bat server*, w trakcie generowania podajemy hasło do klucza oraz potwierdzamy podpisanie klucza prywatnego serwera (bramy) przez CA, otrzymujemy certyfikat i klucz prywatny serwera,

dla klienta(ów)
– user.req, user.key,
– do wystawienia certyfikatów i kluczy dla klientów sieci VPN uruchamiamy aplikację do wystawiania wniosków o wydanie certyfikatu (My Certificate Wizard) w polu Common Name podajemy nazwę użytkownika, pozostałe pola muszą być wypełnione analogicznie jak przy generowaniu CA oraz wskazujemy katalog keys - otrzymujemy klucz prywatny klienta oraz wniosek o wystawienie certyfikatu user.req,
– user.cert,
– uruchamiamy z parametrem *build-key.bat user* i podpisujemy przez CA, otrzymujemy certyfikat klienta,

3.4. Konfigurowanie klienta VPN pod Windows XP

– kopiujemy z katalogu …\sample-config\ plik *client.ovpn* do katalogu …\config\ oraz edytujemy (rys. 4)
– w katalogu powinny się również znaleźć: certyfikat wystawcy CA, certyfikat i klucz klienta.



Rys. 4. Plik konfiguracyjny klienta VPN (źródło: zrzut ekranowy)
Fig. 4. The configurationally file of the customer VPN (source: the print screen)

3.5. Uruchomienie klienta VPN

Klienta VPN można uruchomić ręcznie lub może się uruchamiać automatycznie zmieniając typ uruchomienia OpenVPN na automatyczny po starcie systemu. W trakcie zestawienia połączenia tunelowego pytani jesteśmy o hasło do certyfikatu *user.cert*, dlatego w przypadku połączenia automatycznego musimy hasło wpisać do klucza na stałe, wykonując polecenie:

openssl rsa -in user.key -out user1.key

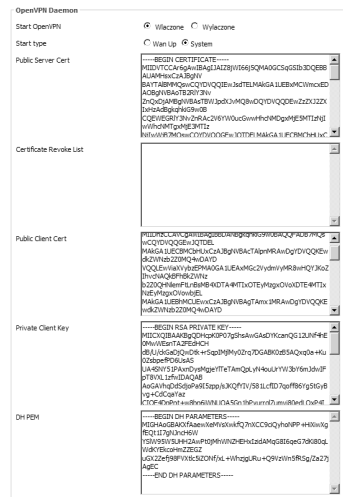
nie zapominając o podmienieniu *user.key* nowym *user1.key*, oczywiście zmieniając nazwę pliku na poprzednią.

3.6. Instalacja OpenVPN na routerze pracującym pod systemem linuksowym

Instalacja pakietu OpenVPN na routerze z systemem linuksowym, który ma możliwość zastąpienia oryginalnego firmware dostarczonego przez producenta urządzenia na firmware dd-wrt, który można pobrać ze strony http://www.dd-wrt.com. Należy wybrać wersję, która ma zaimplementowany pakiet VPN (dd-wrt.v24_vpn).

3.7. Konfiguracja serwera VPN

W routerze w zakładce Services (rys.5) w sekcji OpenVPN Demon ustawiamy Start OpenVPN na włączony i kopiujemy odpowiednio zawartości plików od -----*BEGIN CERTIFICATE*----- do -----*END CERTIFICATE*----- następująco:



– Public Server Cert – ca.cert
– Public Client Cert – server.cert
– Private Klient Key – server.key
– DH PEM – dh1025.pem

Rys. 5. Konfiguracja serwera VPN (źródło: zrzut ekranowy)
Fig. 5. Server VPN configuration (source: the print screen)

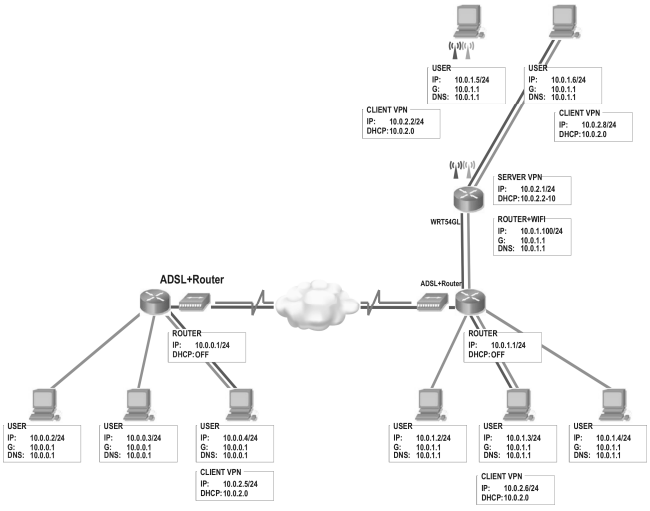
Natomiast w sekcji OpenVPN Config zawarta jest konfiguracja serwera:

mode server – tryb pracy
proto udp – protokół
port 1194 – port
server-bridge 10.0.2.1 255.255.255.0 10.0.2.2 10.0.2.10 – IP serwera VPN, maska oraz zakres DHCP przydzielanych adresów klientom,
dev tap0 – interfejs
keepalive 15 60 – co 15 s wysyłany ping testowy po 60 s w wypadku braku odpowiedzi restart demona VPN,
verb 3 – poziom logowania,
comp-lzo – kompresja włączona
client-to-client – routing pomiędzy klientami sesji VPN
duplicate-cn – wielokrotne połączenie klientów z tym samym certyfikatem
tls-server – wymiana kluczy przed zestawieniem tunelu jest szyfrowana
ca ca.crt – certyfikat CA
dh dh.pem – protokół uzgadniania kluczy prywatnych
cert cert.pem – certyfikat serwera (bramy)
key key.pem – klucz prywatny serwera

Pozostaje jeszcze ustawienie automatycznego uruchamiania Daemona VPN (rys.6.) (Administracja, Polecenia):

Rys. 6. Automatyczne uruchomienie serwera VPN (źródło: zrzut ekranowy)
Fig. 6. Automatic starting the server VPN (source: the print screen)

3.8. Topologia sieci VPN, przykład zastosowania:



Rys. 7. Topologia sieci VPN – przykładowe rozwiązanie (źródło: autor)
Fig. 7. Net VPN topology - the example solution (source: the author)

Dostęp do serwera VPN z poza segmentu sieci LAN (rys. 7), w której znajduje się serwer OpenVPN jest możliwe przez uruchomienie na routerze ADSL tej sieci usługi zarządzania zmieniającym się adresem IP - Dynamic Domain Name System (DDNS) używając nazwy domeny zamiast adresu IP. Jak również musi być otwarty i przekierowany na routerze ADSL port, na którym klient VPN zlokalizowany w innej sieci zestawia połączenie tunelowe z serwerem VPN (port udp 1194 -> 10.0.1.100).

4. Wnioski

Intencją artykułu nie była szczegółowa analiza kryptograficzna, a jedynie w bardzo skrótowy sposób pokazanie mechanizmów bezpiecznej komunikacji otwartym kanałem, oraz przedstawienie technologii VPN – mechanizmu który praktycznie małym nakładem finansowym pozwala tworzyć sieci VPN na dostatecznie dużym poziomie bezpieczeństwa. Przedstawiona w przykładzie topologia sieci VPN oczywiście nie wyczerpuje możliwości zastosowania i konfiguracji. Doświadczenie zawodowe autora z przeprowadzonych kontroli operatorów telekomunikacyjnych a szczególnie sieci WiFi pokazuje, że sieci takie są słabo zabezpieczane. Użytkownicy korzystający w szczególności z sieci radiowych są narażeni na ataki osób trzecich, a ich własne zasoby dyskowe są nieświadomie udostępniane.

5. Literatura

[1] Marek Serafin: Sieci VPN. Helion, Warszawa 2008.
[2] http://openvpn.se
[3] Praca zbiorowa. Vademecum Teleinformatyka I-III. IDG, Warszawa 2004.

Artykuł recenzowany