VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra Informatiky

SSL virtuální privátní sítě

SSL Virtual Private Networks

2013

Bc. Peter Jašica

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Peter Jašica**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: SSL virtuální privátní sítě
SSL Virtual Private Networks

Zásady pro vypracování:

Cílem diplomové práce je návrh, realizace a otestování různých řešení technologie SSL VPN.

Osnova práce:
1. Popište technologii SSL VPN.
2. Navrhněte a v laboratorních podmínkách realizujte různé druhy SSL virtuálních privátních sítí. Použijte k tomu různé platformy - komerční (Cisco, MikroTik) a open source.
3. Do sítě implementujte PKI (Public Key Infrastructure).
4. Srovnejte jednotlivá řešení. Zhodnoťte výhody a nevýhody jejich použití.

Seznam doporučené odborné literatury:

FRAHIM Jazib, HUANG Qiang. *SSL Remote Access VPNs*. Indianapolis: Cisco Press, 2008. ISBN 1-58705-242-3.
DEAL Richard. *The Complete Cisco VPN Configuration Guide*. Indianapolis: Cisco Press, 2006. ISBN 1-58705-204-0.
Dokumentace k zařízením MikroTik.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Petr Machník, Ph.D.**

Datum zadání: 18.11.2011
Datum odevzdání: 07.05.2013

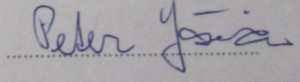
doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlásenie

„Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal."

V Ostrave dňa 31.7.2013



Poďakovanie

Rád by som poďakoval vedúcemu mojej diplomové práce Ing. Petrovi Machníkovi Ph.D. za pomoc a rady, ktoré mi pri tvorbe tejto práce poskytol.

Abstrakt

Táto diplomová práca sa zaoberá virtuálnymi privátnymi sieťami (VPN) typu SSL (Secure Socket Layer). Vysvetľuje ich princíp, popisuje technológiu využívanú pri tvorbe tohto typu VPN. Predstavuje vybrané návrhy a realizácie SSL VPN s implementáciou asymetrického šifrovania a zrovnáva výhody a nevýhody ich použitia. Konfigurácie sú realizované na platformách Open source (Linux), MikroTik typu Site-to-Site a Remote Client postavené na protokole SSTP, ktorý je zapuzdrený s SSL/TLS vrstve. V ďalšej použitej platforme Cisco je konfigurácia prevedená pomocou grafického rozhrania SDM 2.5 (Security Device Manager), ktorý konfiguráciu zjednodušuje a značne urýchľuje. Na platforme Cisco je realizácia prevedená v troch spôsoboch. A to typu Clientless, Thin-Client a Tunnel mode.

Kľúčové slova

SSL, Secure Socket Layer, VPN, Open source, Linux, MikroTik, Site-to-Site, Remote Client, SSTP, SSL/TLS, SDM, Security Device Manager, Cisco, Clientless, Thin-Client, Tunnel mode

Abstrakt

This thesis talks about virtual private networks (VPN) type SSL (Secure Socket Layer). The paper explains its principle, describes the technology used for creation of such VPN type. Also it presents chosen drafts and realization of SSL VPN featuring the implementation of asymmetric cryptography together with comparison of its pros and cons. The configurations are done on platforms like Open source (Linux), MikroTik type Site-to-site and Remote Client which are all based on SSTP protocol that is encapsulated in SSL/TLS layer. In the next used Cisco platform, the configuration is done by the graphic interface SDM 2.5 (Security Device Manager), that makes the configuration more simple and faster. On Cisco platform the realization is done in three ways. Namely the type Clientless, Thin-Client and Tunnel mode.

Keywords

SSL, Secure Socket Layer, VPN, Open source, Linux, MikroTik, Site-to-Site, Remote Client, SSTP, SSL/TLS, SDM, Security Device Manager, Cisco, Clientless, Thin-Client, Tunnel mode

Zoznam použitých skratok

A	
AAA	Authentication, Authorization and Accouting
ATM	Asynchronous Transfer Mode
C	
CA	Certificate autority
CN	Common Name
CRL	Certificate Revocation List
F	
FTP	File Transfer Protocol
H	
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I	
IANA	Internet Assigned Numbers Authority
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPsec	Internet Protocol Security
ITU-T	International Telecommunication Union
L	
L2TP	Layer 2 Tunneling Protocol
M	
MSCHAP	Microsoft Challenge-Handshake Authentication Protocol
MAC	Message Authentication Code
N	
NAT	Network Address Translation
P	
PGP	Pretty Good Privacy
PKI	Public Key Infrastruckture
POP3	Post Office Protocol

PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
R	
RA	Registration Autority
RFC	Request For Comments
RSA	Rivest, Shamir, Adleman
S	
SDM	Security Device Manager
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSL/TLS	Secure Sockets Layer/Transport Layer Security
SSTP	Secure Socket Tunneling Protocol
T	
TAP	interface TAP
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TUN	network TUNnel
U	
UDP	User Datagram Protocol
V	
VoIP	Voice Over Internet Protocol
VPN	Virtual Private Network

OBSAH

1. ÚVOD

V dnešnej dobe používania počítačových sieti a hlavne siete Internet, je pojem bezpečnosť veľmi dôležitým pojmom. Osoby, alebo spoločnosti využívajúce verejné siete, či už na platobné transakcie, alebo prácu s veľmi citlivými údajmi, ktoré môžu byť v sieti treťou stranou zneužité, často podceňujú bezpečnosť. Preto je dôležité tento problému riešiť, a to pomocou využitia softwarových alebo hardwarových možností. Jednou s možnosti ako zabezpečiť a ochrániť komunikáciu medzi týmito subjektmi je technológia VPN (Virtual Private Networks).

Existuje veľa spôsobov realizácie VPN a vybrať konkrétne riešenie nie je jednoduché, ja sa preto snažím v tejto práci popísať a na praktických riešeniach ukázať realizáciu VPN využívanú technológiu protokolov rodiny SSL/TLS (Secure Socket Layer/Transport Layer Security) na platformách Cisco, MikroTik a Linux s implementáciou PKI (Public Key Infrastructure).

V nasledujúcej kapitole popisujem teoretickú problematiku VPN sieti, technológie SSL a TLS, princíp PKI, digitálne certifikáty a certifikačné a registračné autority, ktoré sú bezpochybne stavebným prvkom hierarchickej štruktúry PKI.

Tretí bod je zameraný na vytváranie certifikačnej autority, certifikátov a kľúčov pomocou aplikácie OpenSSL. Ďalej táto časť zahŕňa podrobné konfigurácie jednosmerného a obojsmerného SSL overovania so serverom Apache.

V ďalších kapitolách rozoberám konkrétne konfigurácie na jednotlivých topológiach pod platformami Linux, MikroTik a Cisco. V siedmej kapitole zrovnávam jednotlivé riešenia týchto platforiem, výhody a nevýhody ich použitia.

2. VIRTUÁLNE PRIVÁTNE SIETE

Hlavným dôvodom vzniku VPN (Virtual Private Network) – Virtuálnych privátnych sietí je decentralizácia podnikov a nutnosť elektronickej komunikácie a obchodovania nie len v rámci podniku, ale aj s obchodnými partnermi. Úlohou je nájsť potrebnú rovnováhu medzi nákladmi na prenos informácií a prístup k podnikovým údajom a bezpečnosťou, možnosťami manažmentu a funkčnosťou. Sieť musí zahŕňať hlasové a obrazové služby, čo kladie nároky na kvalitu a náklady siete [14].

VPN sa na trhu objavili počiatkom 90-tych rokov ako konkurencia k službe prenajatých okruhov, kedy prevádzkovatelia sietí začali ponúkať službu Frame Relay na pripojenie lokálnych sietí pomocou verejnej rozľahlej siete. Neskôr sa objavili siete na báze ATM (Asynchronous Transfer Mode), ktoré lepšie spĺňali požiadavky multimediálnych aplikácií. V poslednej dobe s rozmachom sietí na báze IP (Internet Protocol) sa pozornosť obracia k VPN, ktoré využívajú verejné siete IP a internet. V súčasnosti výrobcovia sieťových zariadení (napr. CISCO) ponúkajú multiservisné platformy integrujúce podporu pre Frame Relay, ATM aj IP.

MAN a WAN siete sú tvorené fyzickým prepojením uzlov. Nad fyzickou štruktúrou siete sa vytvárajú logické smerovacie štruktúry nezávisle na fyzickej štruktúre Tieto logické smerovacie štruktúry sa nazývajú virtuálne privátne siete VPN (Virtual Private Network).

Princípom VPN (Virtual Private Network) je využitie verejnej siete (napr. Internet) na bezpečné (autentizované, šifrované) a pre používateľa úplne transparentné prepojenie vzdialených pobočiek, či mobilných používateľov, do podnikovej siete. Pričom za najväčšie výhody sa pokladá:

- VPN poskytujú vzdialený prístup a prepájanie vzdialených lokálnych sietí s podstatne nižšími nákladmi (jednoduchšia integrácia do už existujúcich riešení), ktoré sú nutné na zriaďovanie prenajatých dátových okruhov a telefonický vzdialený prístup.

- Redukcia nákladov spojených s realizáciou sieťových pohybov a zmien (budovanie jednoúčelových sietí). Aktuálnym trendom je tvorba dynamických sietí (možnosťou presúvania sa užívateľov, tvorby nových skupín, celosvetová dostupnosť...), na čo je VPN plne prispôsobená[14].

2.1. Typy VPN

Sieť VPN môžeme rozdeliť do troch základných typov: klient – klient, klient – brána (gateway), brána – brána (gateway – gateway). Komunikácia môže byť zabezpečená na rôznych vrstvách RM OSI modelu, a to napríklad na aplikačnej, transportnej, sieťovej alebo datalinkovej vrstve.

Na aplikačnej vrstve je šifrovanie zaistené programovo, napríklad šifrovacou metódou Pretty Good Privacy (PGP) alebo pomocou zabezpečených kanálov ako je Secure Shell (SSH). Hlavne SSH, ktoré je možné použiť pre vytvorenie tunelu v režimu port – forwarding.

Na transportnej vrstve sa používajú protokoly, ako je napríklad Secure Socket Layer (SSL), pre ochranu užitočného obsahu komunikácie medzi dvomi stranami. Typický sa tento spôsob zabezpečenia využíva pri komunikácii s webovým prehliadačom. Na tejto vrstve je chránený len obsah komunikácie, ale IP pakety, ktoré tieto informácie obsahujú, môže ktokoľvek získať.

Na sieťovej vrstve sa už nešifruje len užitočný obsah, ale aj TCP/IP informácie. Na tejto vrstve pracuje protokol IPSec.

2.2. Certifikačná autorita (CA)

Certifikačná autorita je základným stavebným prvkom hierarchickej štruktúry PKI. Hlavnou úlohou certifikačnej autority je vydávať, spravovať a rušiť certifikáty svojich klientov. Klientmi certifikačnej autority môžu byť buď koncový používatelia, alebo certifikačné autority nižšej úrovne. CA má 4 najdôležitejšie aktíva, ktoré si musí dôkladne strážiť:

1) Súkromný kľúč CA je tým najväčším aktívom. Jeho kompromitácia je pre CA katastrofa, po ktorej už nemá zmysel opätovne činnosť CA obnovovať. Okrem súkromného kľúča musí CA chrániť aj sekvenciu vydaných čísel certifikátov.

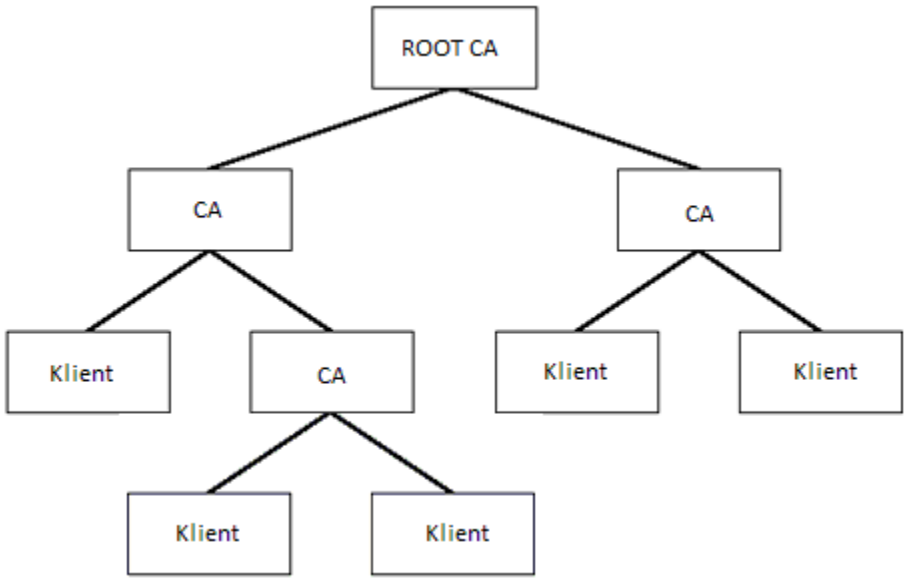
2) Databázu používateľov musí CA chrániť z hneď 2 dôvodov

CA nesmie vydať 2 osobám certifikát s takým istým predmetom. Databáza obsahuje osobné údaje ako rodné čísla, čísla občianskych preukazov

3) Archív súkromných šifrovacích kľúčov používateľov, ak takúto službu CA poskytuje nesmie dopustiť zneužitie týchto súkromných kľúčov

4) Archív listín uložených na CA, archív vydaných certifikátov a CRL. V prípade certifikátov a CRL sa síce jedná o verejné informácie, ale údaje musia byť poskytované minimálne po celú dobu činnosti CA, aby bolo možné overiť dokumenty, ktoré boli podpísané certifikátmi vydanými touto CA. Certifikáty je možné vydať takmer pre čokoľvek. Certifikát môže potvrdzovať totožnosť fyzickej osoby, funkcie v spoločnosti, Organizácie alebo časti organizácie, alebo môže potvrdzovať identitu server na Internete, alebo dokonca autenticitu a neporušenosť softvérového produktu.

Okrem mena subjektu obsahuje digitálne ID aj informácie o čase jeho platnosti, certifikačnej autorite ktorá ho vydala, použitých kryptografických algoritmoch a ďalšie rozšírenia upresňujúce alebo obmedzujúce jeho použitie. Najdôležitejšie z týchto rozšírení špecifikuje, či vydaný certifikát môže byť použitý ako certifikát podradenej certifikačnej autority alebo je to certifikát konečného používateľa. [5,6]



Obr. 1 Organizácia certifikačných autorít

2.3. Registračná autorita (RA)

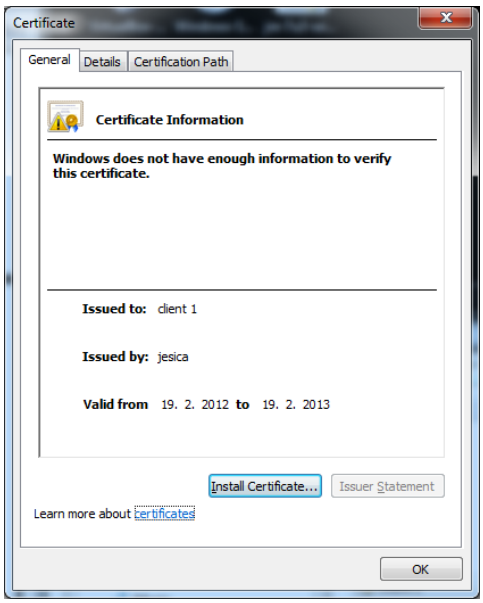
Registračná autorita prijíma žiadosti o vydanie certifikátu. Jej úlohou je fyzické overenie totožnosti žiadateľa o certifikát. Následne je overenú žiadosť odovzdaná certifikačnej autorite, ktorá overí údaje od užívateľa následne vydá, alebo nevydá príslušný certifikát. [5]

2.4. Digitálny certifikát

Digitálny certifikát je dátový súbor v štandardizovanom formáte, ktorý slúži na identifikáciu osoby, alebo servera. Súčasný štandard je X.509, ale existuje návrh na používanie certifikátu z OpenPGP. V základe rozdeľujeme certifikáty na dôveryhodné a nedôveryhodné a to podľa spôsobu vydania. [5,6]

Momentálne sa tento certifikát používa vo verzii 3 a má takúto štruktúru :

- **Verzia** - udáva číslo verzie certifikátu
- **Sériové číslo** - unikátne sériové číslo certifikátu v rámci jednej certifikačnej autority
- **ID algoritmu** - identifikátor algoritmu, použitého na podpis certifikačnou autoritou
- **Vydavateľ** - meno certifikačnej autority, ktorá certifikát vydala a podpísala
- **Platnosť** - čas, počas ktorého je certifikát platný; uvedený je začiatok a koniec platnosti
- **Subjekt** - meno subjektu, na ktoré bol certifikát vydaný
- **Informácia o verejnom kľúči** - obsahuje verejný kľuč subjektu, ktorý dodal certifikačnej autorite a algoritmus použitý na vygenerovanie kľúča
- **Jednoznačný identifikátor vydavateľa** - voliteľný atribút
- **Jednoznačný identifikátor subjektu** - voliteľný atribút
- **Rozšírenia** - voliteľný atribút, umožňuje vložiť do certifikátu ďalšie informácie, existujú preddefinované rozšírenia, ale môžu sa vytvárať aj nové. Každé rozšírenie obsahuje svoj jednoznačný číselný identifikátor a samotnú hodnotu ľubovoľného typu [15]



Obr. 2 Digitálny certifikát

2.5. PKI (Public Key Infrastructure)

Ak si predstavíme, že chceme používať asymetrické šifrovanie vo veľkej sieti, ktorá má napríklad stovky/tisícky užívateľov, museli by sme navzájom vymeniť verejné kľúče medzi každým užívateľom (každý s každým). V prípade, že by jeden z užívateľov stratil, alebo by mu bol nejakým spôsobom privátny kľúč ukradnutý, je nutné vygenerovať nový pár kľúčov a následne opakovane rozoslať medzi ostatnými kolegami. Tento problém je možné vyriešiť pomocou PKI.

PKI (Public Key Infrastructure) popisuje infraštruktúru pre distribúciu verejného kľúča. Slúži k zaisteniu potrebných prostriedkov na bezpečnú komunikáciu. overovanie totožnosti a dodržanie integrity počas prenosu. Hlavné časti infraštruktúry verejných kľúčov je certifikačná autorita (CA), registračná autorita (RA) a zoznam neplatných certifikátov (CRL). [5]

PKI (Public Key Infrastructure) je sústava technických, ale hlavne organizačných opatrení spojených s vydávaním, správou, používaním a odvolávaním platnosti kryptografických kľúčov, certifikátov. Jednou z možných noriem PKI definuje sada internetových štandardov RFC popisujúcich základní využitia asymetrickej kryptografie na Internete, nadväzujú na ne normy týkajúce sa bezpečnej pošty S/MIME a iné. Treba zdôrazniť, že normy PKI vychádzajú z noriem ITU-T rady X.500 (konkrétne X.509 pre popis certifikátu), ale špecifikujú konkrétnu množinu parametrov a praktík určených pre Internet. Teda nie všetky rozšírenia certifikátov popísaných v norme X.509. Preto by sme nemali používať spojenie „certifikát podľa X.509v3“, ale „certifikát podľa RFC-3280“. Pre priblíženie uvediem zopár informácii o X.509. X.509 je štandardom ITU-

T (International Telecommunication Union) pre infraštruktúru verejného kľúča. X.509 medzi iným špecifikuje štandardné formáty pre certifikáty verejných kľúčov a „certification path validation algorithm“ – to je algoritmus, ktorý verifikuje autenticitu CA. Začína s tým čo podpísal daný certifikát a ide až k ROOT CA. X.509 začal v nadväznosti na štandard X.500 a prevzal hierarchický systém certifikačných autorít na vydávanie certifikátov. To je v protiklade s „web of trust“ modelom, ako PGP, kde hocikto (nie iba CA) môže podpísať iné certifikáty (a takto potvrdiť ich validitu). Verzia 3 štandardu X.509 zahŕňa aj flexibilitu v podpore iných typológii napr. „bridge“ a „mesh“. Systém X.500 nikdy nebol naplno implementovaný, a IETF PKI „working group“ prijalo štandardy k flexibilnejšej organizácii internetu. V skutočnosti termín certifikátu z X.509 sa obyčajne spája s profilom štandardu z X.509 v3. V systéme s X.509 CA vydáva certifikát, v ktorom sa zviaže verejný kľúč s daným„ Distinguished name“ v tradícii X.500, alebo k alternatívnemu menu ako email, alebo DNS položka. Trusted root certifikát nejakej organizácie, môže byť distribuovaný všetkým svojim zamestnancom, ktorí tak môžu využívať podnikový PKI systém. Prehliadače ako Microsoft Internet Explorer, Netscape/Mozilla a Opera sa dodávajú s predinštalovanými koreňovými SSL certifikátmi veľkých spoločností, ktoré tak majú výhodu, že budú prístupné bez problémov. [6]

2.6. SSL VPN

SSL VPN je často označovaná rada vzájomne nekompatibilných technológii, ale všetky sú postavené na rovnakej myšlienke. Pre zašifrovanú komunikáciu prístupu k serveru, využíva technológie protokolov rodiny SSL/TLS (Secure Socket Layer/Transport Layer Security). Cieľom SSL VPN je vytvorenie šifrovaného tunelu založeného na protokole SSL. Oproti klasickému IPsec VPN, nemusí byť na koncovej stanici nainštalovaná žiadna špeciálna aplikácia, keďže je SSL prítomné v bežných webových prehliadačoch. Užívateľ sa prostredníctvom webového prehliadača s importovaným certifikátom, alebo užívateľským menom a heslom, môže pristupovať k dôverným informáciám. SSL pracuje medzi transportnou a aplikačnou vrstvou TCP/IP modelu. K rozšíreniu možnosti SSL VPN riešení sú používané malé aplikácie v podobe Java appletou, alebo ActiveX prvkov. [1]

Funkčnosť SSL VPN spočíva vytvorením tunela medzi vpn bránou a webovým prehliadačom na klientskom koncovom bode, čím zabezpečíme prístup k vnútorným informačným zdrojom. Požiadavok od klienta je prijatý bránou, následne predá na žiadaný server, ktorý bráne vráti odpoveď a tá ju odošle klientovi. Komunikácia medzi bránou a klientom je chránená pomocou šifrovania SSL knižnice. [2]

2.6.1. Výhody SSL VPN

Tento typ VPN poskytuje možnosť bezpečnej komunikácie medzi rôznymi vzdialenými užívateľmi a privátnymi sieťami. V súčasnosti podporujú SSL všetky webové prehliadače bez potreby inštalácie klienta ako prípade IPsec.

Je jednoducho použiteľný pre všetkých užívateľov aj bez veľkých počítačových skúsenosti, ktorý môžu používať svoj obľúbený prehliadač.

SSL VPN tiež poskytuje výhodu pripojenia aj pre užívateľov, ktorí pristupujú zo sieti, ktoré majú obmedzený a kontrolovaný prenos a to preto, že tento typ komunikuje na základe bezpečnostného HTTP prenosu (HTTPS).

Taktiež SSL VPN nemá žiadny problém s prekonávaním NAT, alebo proxy serverov.

Medzi ďalšie výhody patrí funkcia vymazania všetkých stiahnutých súborov a informácii po ukončení spojenia. Toto zabraňuje hackerom získavanie informácii z nedôveryhodných zariadení.

2.6.2. Nevýhody SSL VPN

Niektoré prehliadače umožňujú automatické uloženie prihlasovacích údajov do systému, čo ušetrí čas pri opätovnom prihlásení, ale veľmi zjednodušuje možnosť odcudzenia týchto údajov. Preto je potrebné po odhlásení všetky informácie vymazať. Avšak z niektorých nedôveryhodných staníc z ktorých môže užívateľ pristupovať, ako napríklad zariadenia v internetovej kaviarni, neumožnia tieto informácie odstrániť, a preto je lepšie sa im úplne vyhnúť.

Najväčšou nevýhodou SSL VPN je, že nie je možné poskytnúť oprávnenie pre užívateľa na základe IP adresy. Všetci užívatelia sú pripojený prostredníctvom jednej IP adresy, čím sa zamedzuje možnosť filtrácie a kontroly prenosu podľa zdrojovej adresy.

2.7. Transport Layer Security (TLS)

Protokol Transport Layer Security (TLS) je nasledovník SSL, ktorý taktiež umožňuje aplikáciám komunikovať po sieti spôsobom, ktorým zabraňuje odchytávaniu či falšovaniu správ. Pomocou kryptografie poskytuje TLS svojím koncovým bodom autentizáciu a súkromie pri komunikácii Internetu. V porovnaní s SSL protokolom je vylepšený o niekoľko bezpečnostných opatrení, a to napríklad ako používanie overovacieho kódu správy rozšíreného o kľúč, takže len vlastník kľúča dokáže MAC overiť, alebo tiež o pseudonáhodná funkciu, ktorá rozdeľuje vstupne dáta na

polovice a spracováva každú z nich iným hashovacím algoritmom (MD5 a SHA-1), potom ich spája dohromady, čo poskytuje ochranu, pokiaľ by bola nájdená slabina jedného algoritmu.

2.8. Ustanovenie TLS/SSL spojenia

Ustanovenie TLS/SSL spojenia funguje na princípe asymetrického šifrovania, keď každá z komunikujúcich strán ma dvojicu symetrických kľúčov – súkromný a verejný. Verejný kľúč je možné zverejniť a pokiaľ týmto kľúčom ktokoľvek zašifruje nejakú správu, je zaistené, že túto správu bude môcť rozšifrovať len majiteľ použitého verejného kľúča svojím súkromným kľúčom.

Protokol TLS/SSL je založený na výmene záznamov. Každý záznam môže byť voliteľne zakomprimovaný, môže byť k nemu pripojený autentizačný kód (message authentication code - MAC) a môže byť zašifrovaný. Každému záznamu je priradený typ obsahu, ktorý určuje protokol vyššej úrovne.

SSL/TLS zahrnuje tri základné fázy:

1. Dohodu účastníkov na podporovaných algoritmoch
2. Výmenu kľúčov založenú na šifrovaní s verejným kľúčom a autentizáciu vychádzajúcu z certifikátov
3. Šifrovanie komunikácie symetrickou šifrou

V priebehu prvej fázy ustanovenia bezpečného spojenia si klient a server dohodnú kryptografické algoritmy, ktoré budú použité. A to napríklad“

- pre výmenu kľúčov: RSA, Diffie-Hellman, DSA, alebo Fortezza
- pre symetrickú šifru: RC2, RC4, IDEA, DES, 3DES, alebo AES
- pre jednocestné hašovacie funkcie“ MD5, alebo SHA

Ustanovenie SSL spojenia (SSL handshake) prebieha v nasledujúcich krokoch:

- Klient pošle správu ClientHello oznamujúcu najvyššiu verziu TLS, ktorú podporuje, náhodné číslo, zoznam doporučených šifrovacích sád a kompresné metódy.

- Server odpovie správou ServerHello obsahujúcu zvolenú verziu protokolu, náhodné číslo, šifrovaciu a kompresnú metódu vybranú z klientskeho ponúknutého zoznamu.

- Server pošle svoj certifikát, pokiaľ to zvolená šifra umožňuje. Súčasne certifikáty sú založené na X.509, ale existuje návrh na používanie certifikátu z OpenPGP.

- Server môže pomocou správy CertificateRequest vyžadovať certifikát od klienta, aby bolo spojenie autentizované vzájomne.

- Server pošle správu ServerHelloDone, ktorá signalizuje, že ukončil inicializačnú dohodu na používaných mechanizmoch.

- Klient odpovie správou ClientKeyExchange, ktorá môže, ale nemusí obsahovať PreMasterSecret, verejný kľúč(podľa zvolenej šifry).

- Klient a server následne z náhodných čísel a PreMasterSecret pomocou navrhnutej pseudonáhodnej funkcie vypočítajú „master secret“. Všetky ostatné kľúče sú z nej odvodené(a z generovaných náhodných hodnôt).

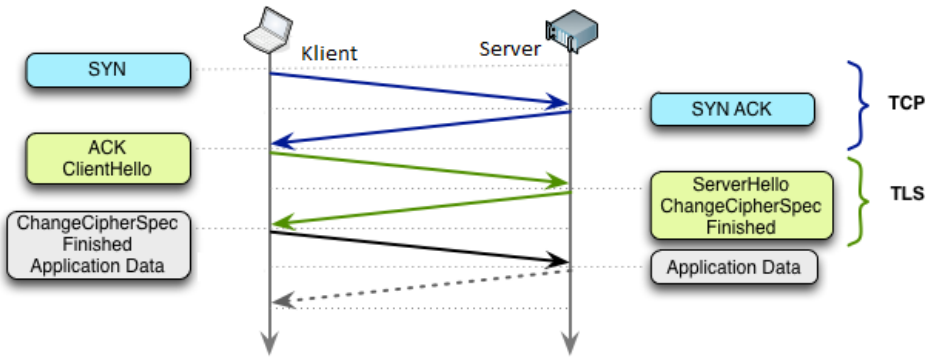
- Klient následne odošle správu ChangeCipherSpec, ktorá v podstate informuje, že všetky ostatné dáta od klienta budú šifrované.

- Na záver klient pošle šifrovanú správu Finished a overí jej hash a MAC (message authentication code) predchádzajúcich iniciačných správ.

- Server sa pokúsi dešifrovať klientovú správu Finished a overiť jej hash a MAC. AK toto dešifrovanie, alebo overenie nezlyhá, je inicializácia považovaná za úspešnú. V opačnom prípade by bolo spojenie ukončené.

- Nakoniec server pošle správu ChangeCipherSpec spolu so svojou zašifrovanou správou Finished a klient urobí analogické dešifrovanie a overenie.

- V tomto okamžiku je inicializácia dokončená, čím je povolený aplikačný protokol. [16]



Obr. 3 Ustanovenie TLS/SSL spojenia

2.8.1. TLS/SSL bezpečnosť

TLS/SSL zahrnuje veľké množstvo bezpečnostných opatrení, a to napríklad:

- Klient používa verejný kľúč certifikačnej autority (CA) k overeniu ich digitálneho podpisu v serverovom certifikáte. Ak je možné digitálny podpis CA overiť, klient príjme serverový certifikát ako platný certifikát vydaný dôveryhodnou CA.
- Klient overuje, či je vydávajúca certifikačná autorita na zozname dôveryhodných certifikačných autorít
- Klient kontroluje dobu životnosti serverového certifikátu. Autentizačný proces sa zastaví ak doba jeho platnosti vypršala.
- K ochrane pred útokmi typu Man-in-the-Middle porovnáva klient aktuálne DNS meno serveru s menom z certifikátu.
- Ochrana pred niekoľkými známymi útokmi (vrátane Man-in-the-Middle), ako snaha o použitie nižšej(menej bezpečnej) verzie protokolu, alebo slabšieho šifrovacieho algoritmu.
- Opatrenie všetkých aplikačných záznamov poradovými číslami a používanie týchto čísel v MAC.
- Používanie overovacieho kódu správy rozšíreného o kľúč, takže len vlastník kľúča dokáže MAC overiť. Definované v RFC2104, ale len u TLS.

- Správa (Finished) ukončujúca inicializáciu obsahuje hash všetkých správ vymenených v rámci inicializácie oboma stranami.

- Pseudonáhodná funkcia rozdeľuje vstupne dáta na polovice a spracováva každú z nich iným hashovacím algoritmom (MD5 a SHA-1), potom ich spája dohromady. To poskytuje ochranu, pokiaľ by bola nájdená slabina jedného algoritmu, taktiež len u TLS.

- SSL v3 je oproti SSL v2 vylepšené o pridanie šifier založených na SHA-1. [16]

3. VYTVÁRANIE CERTIFIKÁTOV POMOCOU APLIKÁCIE OPENSSL A SSL OVEROVANIE

OpenSSL tvorí knižnica pre šifrovanie a overovanie totožnosti a k nej príslušne programy. Jeho využitie je hlavne v HTTPS serveroch, ale taktiež sa používa aj pri iných situáciách, ako napríklad pri prenose šifrovaných dát v iných protokoloch. Vytváranie Certifikačných autorít, certifikátov pomocou aplikácie OpenSSL je veľmi rýchle. Vydávanie certifikátov prostredníctvom OpenSSL v systéme Linux popisujú nasledovné kroky:

Krok 1: Vytvorenie kľúča Certifikačnej autority a páru certifikátu Certifikačnej autority. Počas tohto procesu, budeme musieť vyplniť niekoľko položiek, ako napríklad Common Name (CN), organizáciu, štát, alebo provinciu. V parametri -days zadávame dobu platnosti Certifikačnej autority (CA).

```
openssl genrsa -des3 -out ca.key 4096
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
```

Krok 2: Vytvorenie privátneho kľúča (.key), žiadosti o certifikát a následne podpísanie certifikátu pre server už nami vytvorenou certifikačnou autoritou v predchádzajúcom odstavci. Následne je opäť potrebné vyplniť potrebné položky. Položka CN nemôže obsahovať zhodu s CN certifikačnej autority, z dôvodu možnej kolízie: CN by mal zodpovedať názvu, alebo ip adrese serveru.

```
openssl genrsa -des3 -out server.key 4096
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 3650 -in server.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt
```

Po vygenerovaní tohto kľúča a certifikátu môžeme žiadosť o podpísanie certifikátu (.csr) zmazať a začať vytvárať certifikáty pre klientov, ktoré budú podpísané taktiež rovnakou certifikačnou autoritou:

```
openssl genrsa -des3 -out client.key 4096
openssl req -new -key client.key -out client.csr
openssl x509 -req -days 3650 -in client.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out client.crt
```

Pre kontrolu vytvoreného certifikátu slúži nasledovný príkaz:

```
openssl x509 -noout -text -in server.crt -purpose
```

Na uloženie privátneho kľúča a certifikátu do jedného súboru vo formáte PKCS#12, ktorý je chránený heslom a tiež býva použitý na import do webových prehľadávačov slúži nasledovný príkaz:

```
openssl pkcs12 -export -inkey client.key -in client.cer -out client.p12
```

V takomto prípade súbory .key .cer a .csr vymažeme. Certifikát vo formáte .p12 následne bezpečnou cestou prenesieme na stanicu klienta.

3.1. SSL overovanie so serverom Apache

Protokol HTTP sám o sebe neumožňuje žiadnu možnosť šifrovania dát pri prenose medzi klientom a serverom. Zároveň ani neumožňuje jednoznačnú identifikáciu klienta a servera. To znamená, že je možné odchytiť voľne prenášané dáta a zároveň sa vydávať za iného klienta, alebo server, čím je možné neoprávnene užívať práva klienta, alebo serveru. Tento problém je možné vyriešiť použitím šifrovania pomocou SSL, kde dochádza k overeniu identity klienta a serveru certifikátom pomocou verejného a privátneho kľúča, čím sa uskutoční šifrovaný prenos dát. Obecne sa používa overovanie serveru, čím zamedzíme krádež, alebo vsunutie nechcených dát v rámci komunikácie medzi klientom a serverom. Zároveň je však možné vydávať aj klientske certifikáty pre identifikáciu klienta. Vo výsledku, môžeme využiť aj vďaka využitia princípu PKI zaistenie a dôveryhodnú autentizáciu komunikujúcich strán a to dvomi možnými metódami. Jednosmerne SSL overovanie a Obojsmerné SSL overovanie identity.

Jednosmerné SSL overovanie (One-way SSL authenication): Umožňuje SSL klientovi overiť identitu SSL serveru, ale SSL server neumožňuje overiť identitu SSL klienta. Tento spôsob SSL overovania využíva pri komunikácii protokol HTTPS ako väčšina webových serverov.

Obojsmerne SSL overovanie (Two-way SSL autentification| mutual SSL autentification): umožňuje SSL klientovi overiť identitu SSL serveru a zároveň umožňuje SSL serveru taktiež overiť identitu SSL klienta. Tento typ overovania sa taktiež nazýva aj klientskou autentizáciou, pretože SSL klient pomocou klientskeho certifikátu preukazuje svoju identitu SSL serveru.

3.2. Konfigurácia jednosmerného SSL overovania

Pri tejto konfigurácií je potrebne mať nainštalovanú službu apache2. Spustenie jednosmerného SSL overovania pozostáva z nasledujúcich krokov:

Krok1: Vygenerovaný certifikát, privátny kľúč a certifikát certifikačnej autority skopírujeme do adresára /etc/apache2/ssl.

Krok 2: Otvorenie portu 443 v súbore /etc/apache2/ports.conf nasledujúcou direktívou:

```
<IfModule mod_ssl.c>
Listen 443
</IfModule>
```

Krok 3: Aktivovanie ssl modulu príkazom v terminále:

```
a2enmod ssl
```

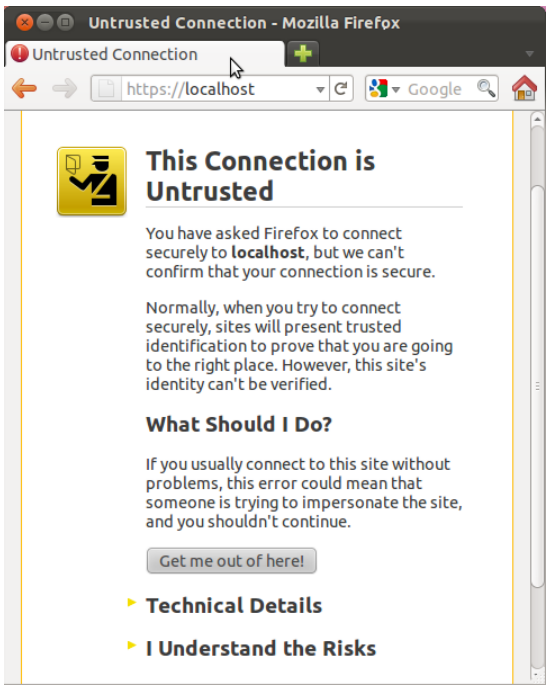
Krok 4: V adresári /etc/apache2/sites-available/ v súbore default-ssl je potrebné nastaviť direktívy SSLEngine na hodnotu ON a SSLCertificateFile, SSLCertificateKeyFile, kde sa definuje cesta k certifikátom a kľúčom servera. Po úprave majú nastať zmeny uvedené v prílohe pod označením I:

Krok 5: Reštart deamona v terminále:

```
service apache2 restart
```

3.3. Testovanie jednosmerného SSL overenia

Testovanie funkčnosti je možné vykonať pomocou webového prehliadača. Pri načítaní adresy je potrebné importovanie certifikátu do webového prehliadača.



Obr. 4 Testovanie jednosmerného SSL overenia

3.4. Konfigurácia obojsmerného SSL overovania

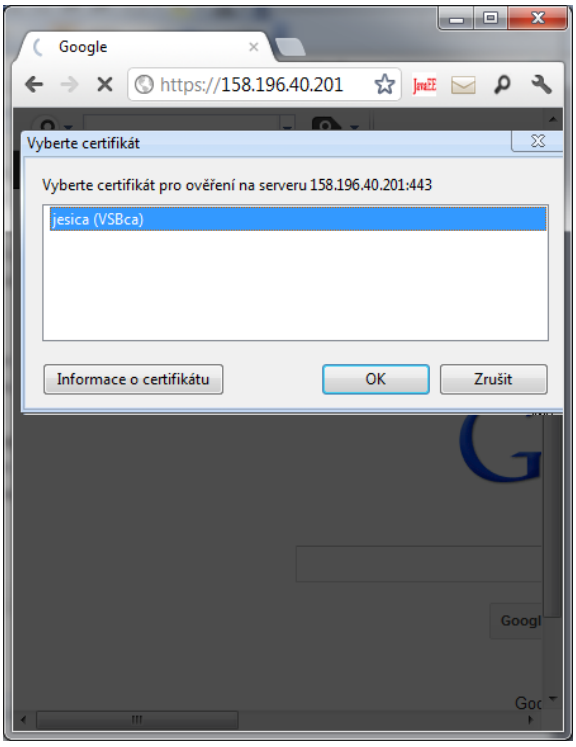
Rozdiel v konfigurácii jednosmerného a obojsmerného nie je veľmi rozdielny. Je potrebné pridať niekoľko direktív v súbore default-ssl, ktorý sa nachádza v adresári /etc/apache2/sites-available/. A to nasledovne:

```
SSLVerifyClient require
SSLVerifyDepth 10
SSLCACertificateFile /etc/apache2/ssl/ca.cer
```

Kde direktíva SSLVerifyClient s hodnotou require zabezpečuje, že so serverom nebudú môcť komunikovať klienti, ktorí sa nepreukážu platným certifikátom od jednej z dôveryhodných autorít. Direktíva "SSLVerifyDepth" určuje, či môže byť klient vydaný aj podriadenou CA. Posledná direktíva "SSLCACertificateFile" definuje cestu k súboru s certifikátmi autorít, od ktorých sú akceptované klientske certifikáty. Po týchto úpravách je potrebne reštartovať webový server.

3.5. Testovanie obojsmerného SSL overenia

Testovanie funkčnosti je možné vykonať pomocou webového prehliadača. Pri načítaní adresy je potrebné importovanie certifikátu do webového prehliadača. Na túto adresu má prístup len používateľ, ktorý vlastní klientsky certifikát od príslušnej autority.



Obr. 5 Obojsmerna SSL autentizacia

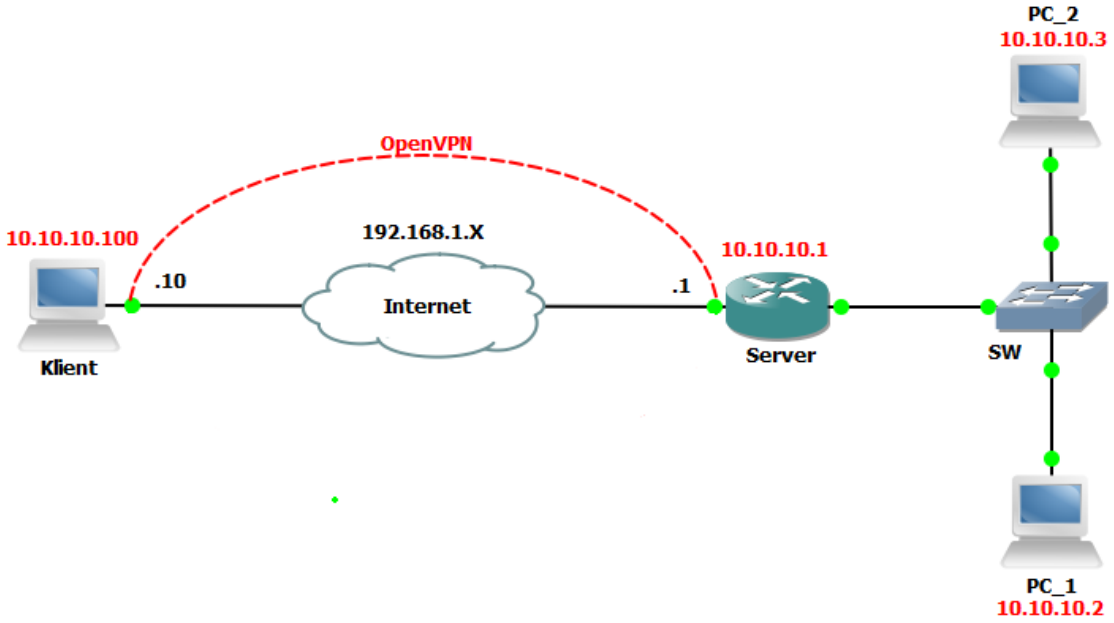
4. SSL VPN NA PLATFORME OPEN SOURCE

OpenVPN vytvára VPN pomocou SSL/TLS (Secure Socket Layer/Transport Layer Security). SSL/TLS spoľahlivý a hodne rozšírený protokol umožňujúci vytvoriť bezpečnostný tunel.

OpenVPN je open-source VPN riešenie, ktoré je výbornou alternatívou k IPSec. Na rozdiel od IPSec pracuje OpenVPN v užívateľskom prostredí operačného systému a nie je súčasťou jadra, k čomu napomáha prevádzanie nízko úrovňové šifrovanie dát na virtuálnom sieťovom rozhraní(TUN, alebo TAP). OpenVPN používa rozšírenú open-source implementáciu SSL/TLS. Taktiež umožňuje využívanie všetkých dostupných kryptografických metód v OpenSSL, čím je tento VPN protokol jedným z najbezpečnejších. OpenVPN nemá žiadny problém s prekonávaním NAT, alebo proxy serverov. Zdrojová adresa paketu sa pri prechode môže ľubovoľne meniť.

OpenVPN je možné prevádzkovať na transportnom protokole UDP alebo TCP. IANA (anglicky Internet Assigned Numbers Authority) stanovila štandardný port 1194 (UDP aj TCP), avšak toto číslo portu môže byť nastavené na ľubovoľnú hodnotu. [7,8]

4.1. Konfigurácia OpenVPN 2.3.1



Obr. 6 Topológia siete

V prvom kroku je potrebná inštalácia programu na oboch stranách budúcej novovytvorej siete:

```
apt-get install openvpn openssl
```

Certifikáty musia byť podpísané tou istou certifikačnou autoritou, takže si CA vytvoríme na strane serveru. Avšak si najskôr vytvoríme adresár pre CA a iné potrebné podadresáre a následne prázdny súbor index.txt, súbor serial s obsahom "01":

```
cd /etc/ssl/
mkdir diplomCA diplomCA/certs diplomCA/crl diplomCA/newcerts diplomCA/private
touch /etc/ssl/diplomCA/index.txt
echo 01 > /etc/ssl/diplomCA/serial
```

Vygenerujeme certifikát certifikačnej autority, ktorý si sami sebou podpíšeme:

```
cd /etc/ssl/diplomCA
openssl req -new -x509 -nodes -out cacert.pem -keyout cakey.pem -days 365
```

Pre dokončenie generovania certifikátu zadáme požadované doplňujúce informácii o CA:

```
Generating a 1024 bit RSA private key
.++++++
....++++++
writing new private key to 'cakey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CZ
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:vsb
Email Address []:
```

Vygenerovaný kľúč a certifikát umiestnime do správnych adresárov. Certifikát do adresára /etc/ssl/diplomCA a kľúč, ktorému zmeníme práva proti neoprávnenej manipulácii do adresára /etc/ssl/diplomCA/private/

```
mv cacert.pem certs/ && mv cakey.pem private/
chmod 400 private/cakey.pem
```

Veľmi dôležitá je kontrola ciest v konfiguračnom súbore programu openSSL, ktorý sa nachádza v /etc/ssl/openssl.cnf. Tieto cesty nájdeme v sekcii [CA_default]. Tento výpis je zadokumentovaný v prílohe pod označením II.

Po kontrole súboru openssl.conf si vytvoríme certifikáty pre server a klientov. Pri vyplnení požadovaných parametrov je dôležité zadať rovnaké parametre uvedené v sekcii [policy_match] konfiguračného súboru openssl.cnf. Na strane servera si vytvoríme adresár server, kde budú dočasne uložené súbory request.pem a key.pem.

```
mkdir server && cd server
openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
```

Potvrdenie certifikačnou autoritou:

```
openssl ca -in request.pem -out cert.pem
```

V tomto kroku môže nastať chyba, v prípade, že nefungujú relatívne cesty v súbore openssl.cnf definované pomocou "$dir",

```
Chyba
openssl ca -in request.pem -out cert.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Error opening CA certificate ./diplomCA/cacert.pem
17226:error:02001002:system library:fopen:No such file or directory:bss_file.c:352:fopen
('./diplomCA/cacert.pem','r')
17226:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:354:
unable to load certificate
```

Tento problém vyrieši zadefinovanie týchto ciest staticky. Upravená sekcia so statickými cestami je zobrazená v prílohe pod označením III. Pri správnom potvrdení certifikačnou autoritou potvrdíme žiadosť:

```
Using configuration from /usr/lib/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 3 (0x3)
            Not Before: Apr 24 21:18:00 2012 GMT
            Not After : Apr 24 21:18:00 2013 GMT
        Subject:
            countryName               = CZ
            stateOrProvinceName       = Some-State
            organizationName          = Internet Widgits Pty Ltd
            commonName                = vsb
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            X509v3 Authority Key Identifier:
                keyid:FF:7E:01:54:86:D2:1C:A8:B5:AF:41:34:46:BA:0C:D5:42:41:92:79
Certificate is to be certified until Apr 24 21:18:0 2013 GMT (365 days)
Sign the certificate? [y/n]:y
```

Po úspešnom potvrdení premiestnime certifikát a kľúč z adresára server do /etc/openvpn/:

```
mv cert.pem /etc/openvpn/cert.pem
mv key.pem /etc/openvpn/key.pem
```

Na strane servera je ešte potrebné vytvoriť súbor s Diffie-Hellmann algoritmom, ktorý bude uložený do adresára /etc/ssl/diplomCA. Tento súbor je potrebný na ustanovenie kľúčov pri komunikácii. Vytvoríme ho pomocou príkazu:

```
openssl dhparam -out /etc/ssl/diplomCA/dh1024.pem 1024
```

Obdobným postupom vytvoríme certifikát pre klientov, ktorý taktiež musím potvrdiť certifikačnou autoritou do adresára client:

```
mkdir client && cd client
openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
openssl ca -in request.pem -out cert.pem
```

Týmto máme vygenerované všetky potrebné certifikáty (certifikát a certifikačnú autoritu) a kľúč, ktoré prenesieme bezpečnou cestou do adresára /etc/openvpn/ na klientskej stanici.

Na strane servera vytvoríme v /etc/openvpn/ súbor vpn_server.conf, Ktorý bude obsahovať kód s potrebnými informáciami a správnymi cestami k požadovaným súborom:

```
mode server
tls-server
dev tap0
port 1194

ifconfig 10.10.10.1 255.255.255.0
ifconfig-pool 10.10.10.100 10.10.10.200 255.255.255.0
duplicate-cn
proto udp

ca /etc/ssl/diplomCA/certs/cacert/cacert.pem
cert /etc/openvpn/cert.pem
key /etc/openvpn/key.pem
dh /etc/ssl/diplomCA/dh1024.pem

log-append /var/log/openvpn
status /tmp/vpn.status 10

user root
group root
comp-lzo
verb 3
```

```
keepalive 1 220
```

Rovnako budeme postupovať aj na klientskej strane. V zložke /etc/openvpn/ vytvoríme súbor vpn_client.conf . Tento súbor musí taktiež obsahovať správne informácie a cesty:

```
remote 192.168.1.1 ### IP adresa serveru
tls-client
dev tap
pull

mute 10
ca /etc/openvpn/cacert.pem
cert /etc/openvpn/cert.pem
key /etc/openvpn/key.pem

comp-lzo
verb 3
```

Po týchto krokoch môžme OpenVPN spustiť. Na strane servera príkazom:

```
/etc/init.d/openvpn start
```

po ktorom sa nám zobrazí informácia o jeho stave:

```
root@jesica:/etc/openvpn# /etc/init.d/openvpn start
 * Starting virtual private network daemon (s)...                 *   Autostarting VPN 'vpn_server'
* Already running (PID file exists)
                                                        [ OK ]
root@jesica:/etc/openvpn# /etc/init.d/openvpn restart
 * Stopping virtual private network daemon (s)...                  *   Stopping VPN 'vpn_server'
[ OK ]
 * Starting virtual private network daemon (s)...                 *   Autostarting VPN 'vpn_server'
[ OK ]
```

Na strane klienta spustíme aplikáciu z adresára /etc/openvpn príkazom:

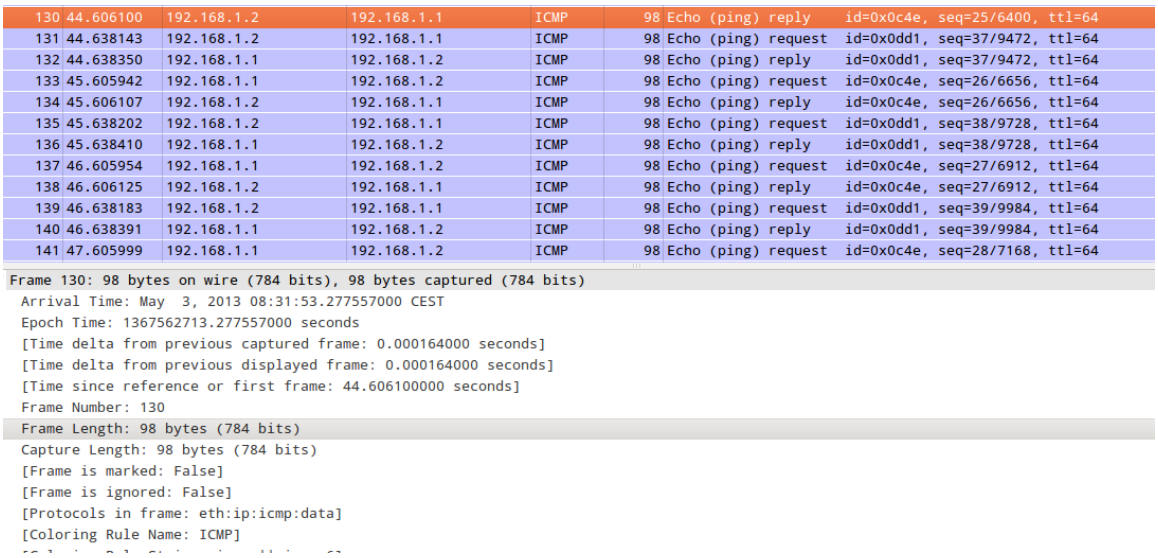
```
openvpn --config ./vpn_client.conf
```

s informaciou uvedenou v prílohe pod označením IV.

4.2. Testovanie OpenVPN

Tunel je testovaný pomocou odchytávania paketov v programe wireshar. Test bol prevedený v dvoch variantách:

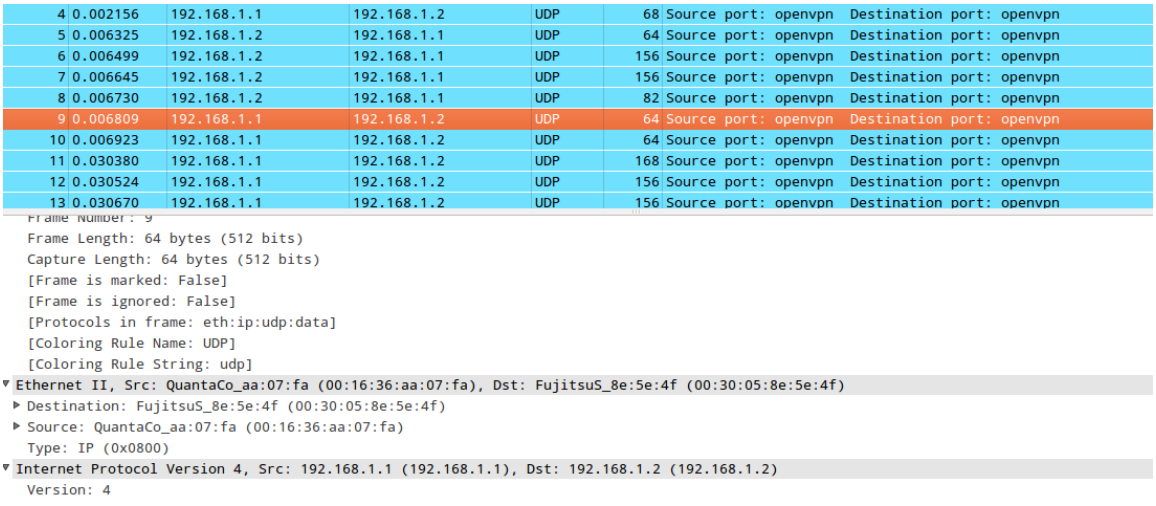
Odchytávanie paketov mimo tunnel



130	44.606100	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) reply id=0x0c4e, seq=25/6400, ttl=64
131	44.638143	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) request id=0x0dd1, seq=37/9472, ttl=64
132	44.638350	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) reply id=0x0dd1, seq=37/9472, ttl=64
133	45.605942	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x0c4e, seq=26/6656, ttl=64
134	45.606107	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) reply id=0x0c4e, seq=26/6656, ttl=64
135	45.638202	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) request id=0x0dd1, seq=38/9728, ttl=64
136	45.638410	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) reply id=0x0dd1, seq=38/9728, ttl=64
137	46.605954	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x0c4e, seq=27/6912, ttl=64
138	46.606125	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) reply id=0x0c4e, seq=27/6912, ttl=64
139	46.638183	192.168.1.2	192.168.1.1	ICMP	98	Echo (ping) request id=0x0dd1, seq=39/9984, ttl=64
140	46.638391	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) reply id=0x0dd1, seq=39/9984, ttl=64
141	47.605999	192.168.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x0c4e, seq=28/7168, ttl=64

Obr. 7 Odchytávanie packetov mimo tunnel

Odchytávanie paketov v tunely



4	0.002156	192.168.1.1	192.168.1.2	UDP	68	Source port: openvpn Destination port: openvpn
5	0.006325	192.168.1.2	192.168.1.1	UDP	64	Source port: openvpn Destination port: openvpn
6	0.006499	192.168.1.2	192.168.1.1	UDP	156	Source port: openvpn Destination port: openvpn
7	0.006645	192.168.1.2	192.168.1.1	UDP	156	Source port: openvpn Destination port: openvpn
8	0.006730	192.168.1.2	192.168.1.1	UDP	82	Source port: openvpn Destination port: openvpn
9	0.006809	192.168.1.1	192.168.1.2	UDP	64	Source port: openvpn Destination port: openvpn
10	0.006923	192.168.1.1	192.168.1.2	UDP	64	Source port: openvpn Destination port: openvpn
11	0.030380	192.168.1.1	192.168.1.2	UDP	168	Source port: openvpn Destination port: openvpn
12	0.030524	192.168.1.1	192.168.1.2	UDP	156	Source port: openvpn Destination port: openvpn
13	0.030670	192.168.1.1	192.168.1.2	UDP	156	Source port: openvpn Destination port: openvpn

Obr. 8 Odchytávanie packetov v tunnely

5. SSL VPN NA PLATFORME MIKROTIK

5.1. Secure Socket Tunneling Protocol (SSTP)

Protokol SSTP je jeden z najmladších tunelovacích protokolov. SSTP bol vyvinutý firmou Microsoft a súčasnej dobe je implementovaný v systémoch Windows od verzie Windows Vista sp1 a Windows Server 2008. [9]

Protokol SSTP umožňuje prenos PPP rámcov cez paketovú sieť v podobe PPTP, alebo L2TP s rozdielom, že celý SSTP protokol je zapuzdrený v SSL/TLS vrstve. SSTP vytvára tunel rovnakým spôsobom ako sa vytvára HTTPS spojenie na server. Ku svojej funkčnosti využíva port TCP 443. SSPT má vďaka SSL/TLS zaistenú autentičnosť, integritu a dôvernosť celého tunelu. SSL/TLS hanshake (ustanovenie kľúčov) zaisťuje spoľahlivá autentizácia servera a klienta. [9]

SSTP so svojou veľkou odozvou nie je vhodný na komunikáciu VoIP, alebo iné protokoly vyžadujúce nízke oneskorenie. Medzi jeho nevýhody patrí využívanie transportného protokolu TCP, čo pri potvrdzovaní prijatých segmentov spôsobuje vyššiu odozvu. Ďalšou nevýhodou, je slabá podpora v operačných systémoch a zariadení. Naopak výhodou SSTP je dokonalé krytie, pri odhaľovaní tunela. SSTP tunel je veľmi zložité rozoznať od klasického HTTPS spojenia, čím je veľký problém tento tunel zablokovať. SSTP je vhodné použiť v prípade, že potrebujeme tunel maskovať na HTTPS spojenie. V sieťach, kde sú tunely zakázané, je SSTP jedným z možných riešení. [9]

5.2. Nadviazanie SSTP tunela

Nadviazanie SSTP tunela medzi klientskou a serverovou stanicou prebieha v nasledujúcich krokoch:

Krok 1: SSTP klient nadviaže spojenie s SSTP serverom medzi dynamickým prideleným portom na SSTP klientovi a na strane serveru na porte 443.

Krok 2: SSTP klient pošle SSL Client-Hello správu, čo znamená, že SSTP chce vytvoriť SSL reláciu s SSTP serverom.

Krok 3: SSTP server odošle svoj certifikát SSTP klientovi.

Krok 4: SSTP klient overí certifikát, určuje metódu šifrovania pre SSL spojenie. Vygeneruje SSL kľúč a zašifruje ho verejným kľúčom SSTP certifikátu servera a následne odošle SSL kľúč SSTP serveru.

Krok 5:Server dešifruje šifrovaný SSL kľúč súkromným kľúčom svojho certifikátu. Každá ďalšia komunikácia medzi klientom a serverom SSTP je šifrovaná pomocou SSL kľúča.

Krok 6: SSTP klient odošle HTTP cez SSL správu žiadosť SSTP serveru

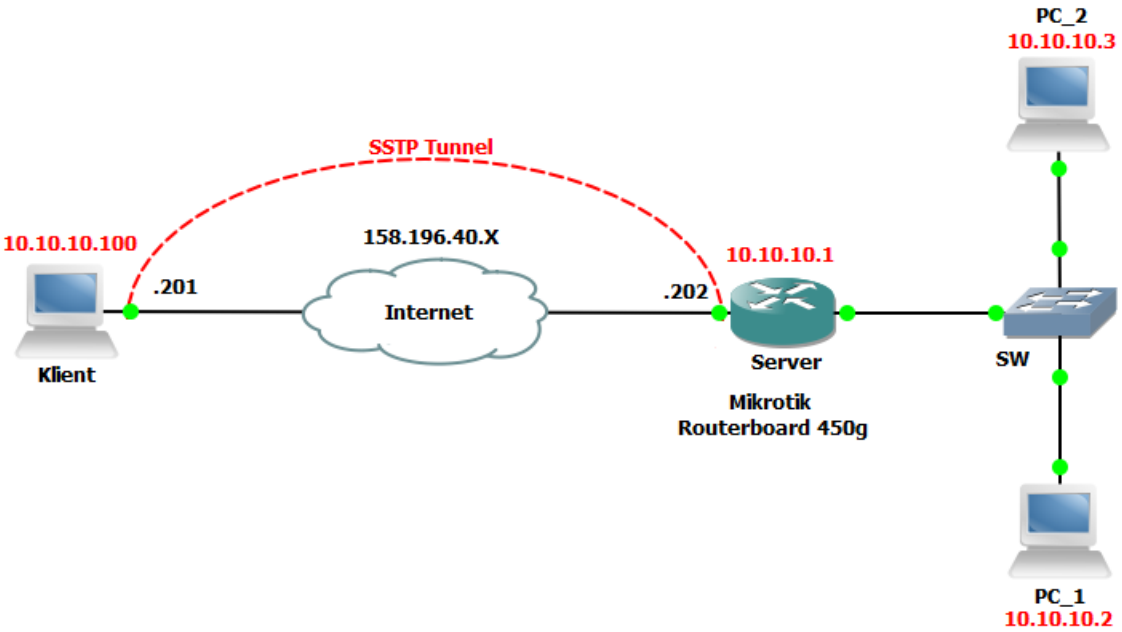
Krok 7: SSTP klient vyjednáva tunel s SSTP serverom

Krok 8: SSTP klient dohodne PPP spojenie s SSTP serverom. Táto dohoda obnáša aj overenie užívateľa cez PPP a konfiguráciu nastavenia protokolu IPv4, alebo IPv6.

Krok 9: SSTP klient začne posielať IPv4, alebo IPv6 cez PPP spojenie. [9]

5.3. Konfigurácia SSTP (Remote Client)

Vzorová konfigurácia, bola vytvorená na sieťových zariadeniach komerčnej firmy Mikrotik (Routerboard 450g) a klientska koncová stanica, z dôvodu jednoduchšieho spravovania certifikačných autorít a certifikátov na systémoch Linux (Ubuntu 11.10).



Obr. 9 Topológia siete (Remote Client)

5.4. Konfigurácia Serveru

V prvom kroku je potrebné nastaviť základnú konfiguráciu systému Mikrotik, a to nastavenie rozhraní. Server je prostredníctvom rozhrania ether1 pripojený do siete Internet a rozhranie ether3 patrí do siete s pracovnými stanicami. Pred začiatkom konfigurácie nesmieme zabudnúť na nastavenie aktuálneho dátumu a času, z dôvodu možného vyskytnutia problému vypršania platnosti pri overení certifikátu. Certifikáty a certifikačná autorita, boli vytvorené v systéme Linux pomocou aplikácie OpenSSL. Pri vytváraní certifikátov je veľmi dôležité zadať do popisu certifikátu správne hodnoty a to obzvlášť pri hodnote CN, ktorá musí byť pomenovaná IP adresou rozhrania, na ktoré sa bude SSTP klient pripájať prostredníctvom SSTP tunela. V prípade, že použijeme Windows klienta, môžeme využiť aj doménové meno.

Najjednoduchším spôsobom ako nahrať certifikáty na daný smerovač, je prenesenie súborov certifikačnej autority, certifikátu a kľúča do okna File využitím aplikácie winbox, odkiaľ sú jednoduchými príkazmi importované. [9]

Nastavenie dátumu a času na aktuálne hodnoty:

```
[admin@MikroTik] /system clock> set time=3:33:33
[admin@MikroTik] /system clock> set date=apr/30/2012
```

Nastavenie rozhraní na sieťovom prvku Mikrotik Routerboar 450g. Rozhranie ether3 je pripojené do verejnej siete Internet a rozhranie do súkromnej siete.

```
[admin@MikroTik] //ip address add address=158.196.40.202/24 interface=ether1
[admin@MikroTik] //ip address add address=10.10.10.1/24 interface=ether3
```

Importovanie certifikátov, CA, a kľúčov, ktoré boli jednoduchým prenesením z adresára do okna Files nakopírovane prostredníctvom aplikácie winbox:

```
[admin@MikroTik] /certificate> import file-name=server.crt
[admin@MikroTik] /certificate> import file-name=server.key
[admin@MikroTik] /certificate> import file-name=ca.crt
```

Po základnej konfigurácii si vytvoríme užívateľa s heslom, ktorým sa budeme prihlasovať do VPN siete. Ďalej je nastavenie lokálnej a vzdialenej adresy na ktorú sa bude prihlasovať klientsky užívateľ. Lokálna adresa je rovnaká ako adresa rozhrania súkromnej siete. [9]

```
[admin@MikroTik] /ppp secret  >add name=Laptop service=sstp password=123
local-address=10.10.10.1 remote-address=10.10.10.100
```

Overenie správnej konfigurácie vytvorenia užívateľa a nastanie lokálnej a vzdialenej adresy:

```
[admin@MikroTik] /ppp secret> print detail
Flags: X - disabled
 0   name="Laptop" service=sstp caller-id="" password="123" profile=default
     local-address=10.10.10.1 remote-address=10.10.10.100 routes=""
     limit-bytes-in=0 limit-bytes-out=0
```

Spustenie SSTP serveru a nastavenie potrebných atribútov:

```
[admin@MikroTik] /interface sstp-server server> set certificate=cert1
[admin@MikroTik] /interface sstp-server server> set enabled=yes
[admin@MikroTik] /interface sstp-server server> set authentication=mschap2
```

Kontrola spustenia a nastavenia SSTP servera:

```
[admin@MikroTik] /interface sstp-server server> print
                 enabled: yes
                    port: 443
                 max-mtu: 1500
                 max-mru: 1500
                    mrru: disabled
       keepalive-timeout: 60
         default-profile: default
          authentication: mschap2
             certificate: cert1
 verify-client-certificate: no
```

Výpis z terminálu servera, ktorý zobrazuje záznamy o pripojených klientoch

```
[admin@MikroTik] /interface sstp-server> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME            USER          MTU CLIENT-ADDRESS     UPTIME   ENCO
 0    DR<sstp-...     Laptopv       1500 158.196.40.201        12s
```

5.5. Konfigurácia klienta

Vzhľadom k častým problémom, ktoré obnáša spravovanie novo vytvorenej certifikačnej autority a certifikátov do systému Microsoft Windows, z dôvodu veľkej bezpečnostnej politiky tohto systému voči nedôverným autoritám, je vytvorený SSTP klient pod platformou Linux (Ubuntu 11.10). Keďže je v tomto systéme protokol SSTP štandardne zakázaný, je potrebné nastaviť jeho povolenie. Pred inštaláciou je potrebne stiahnuť inštalačne balíky, ktoré nám umožnia tento protokol pod systémom Linux využívať:

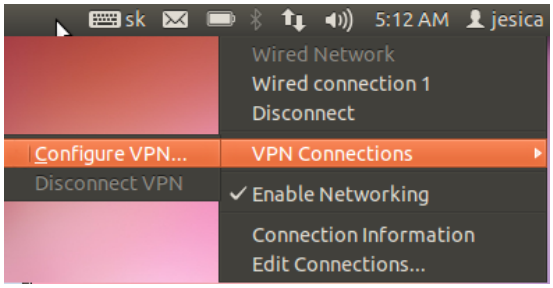
- sstp-client_1.0.4_i386.deb
- network-manager-sstp-gnome_0.9.1-0ubuntu2_i386.deb
- network-manager-sstp_0.9.1-0ubuntu2_i386.deb

Po úspešnom stiahnutí súborov pokračujeme inštaláciou:

```
root@jesica:/home/jesica/Downloads# dpkg --install sstp-client_1.0.4_i386.deb
root@jesica:/home/jesica/Download#dpkg --installnetwork-manager-sstp-gnome_0.9.10 ubuntu3_i386.deb
root@jesica:/home/jesica/Downloads# dpkg --install network-manager-sstp_0.9.1-0ubuntu3_i386.deb
```

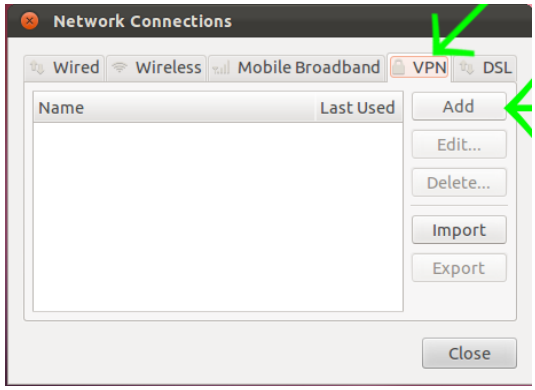
Nastavenie VPN pripojenia nastavíme podľa krokov zobrazených v grafike:

Krok 1:



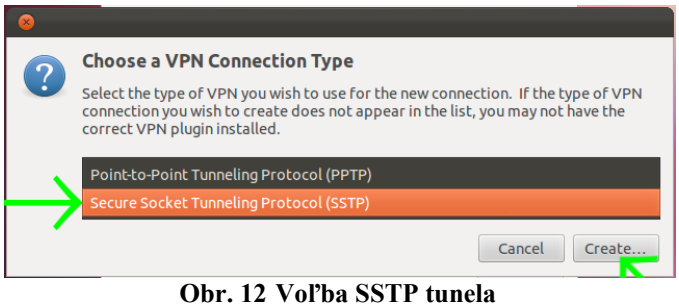
Obr. 10 Konfigurácia VPN pripojenia u SSTP klienta

Krok 2:



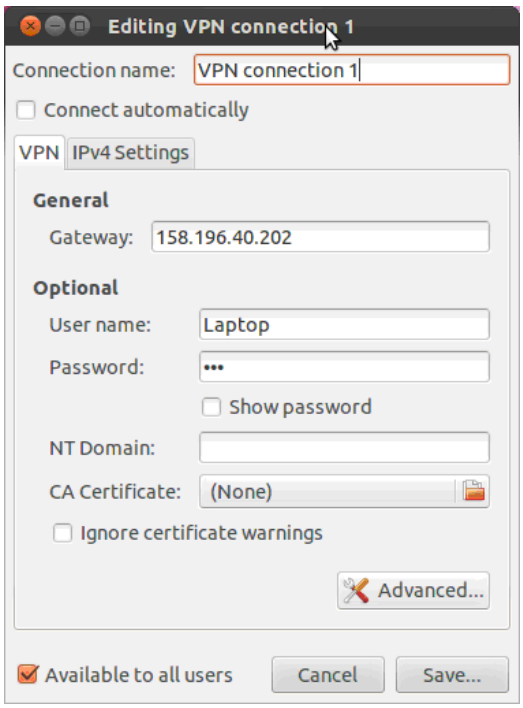
Obr. 11 Pridanie novej VPN

Krok 3:



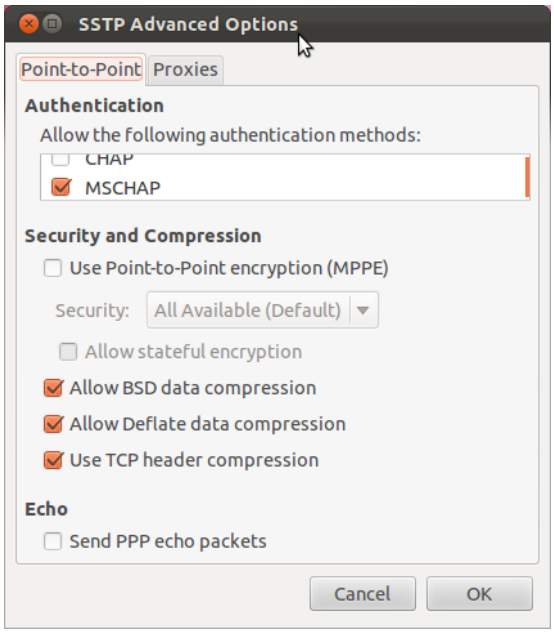
Obr. 12 Voľba SSTP tunela

Krok 4: Pomenovanie VPN pripojenia, nastavenie adresy servera a prihlasovacích údajov

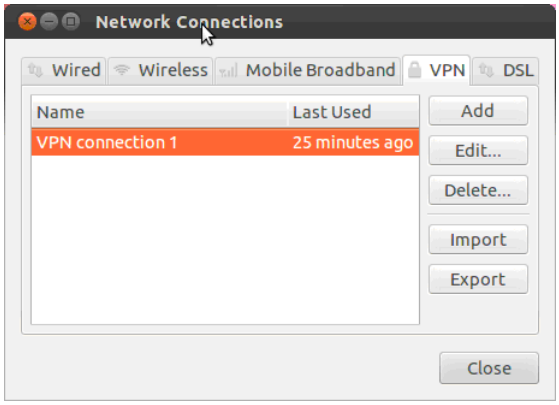


Obr. 13 Definovanie VPN spojenia

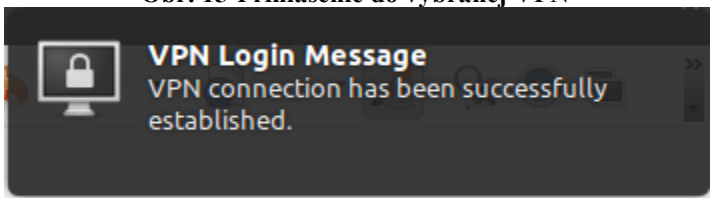
Krok5: Povolenie MSCHAP overovanie:



Obr. 14 Nastavenie autentizácie



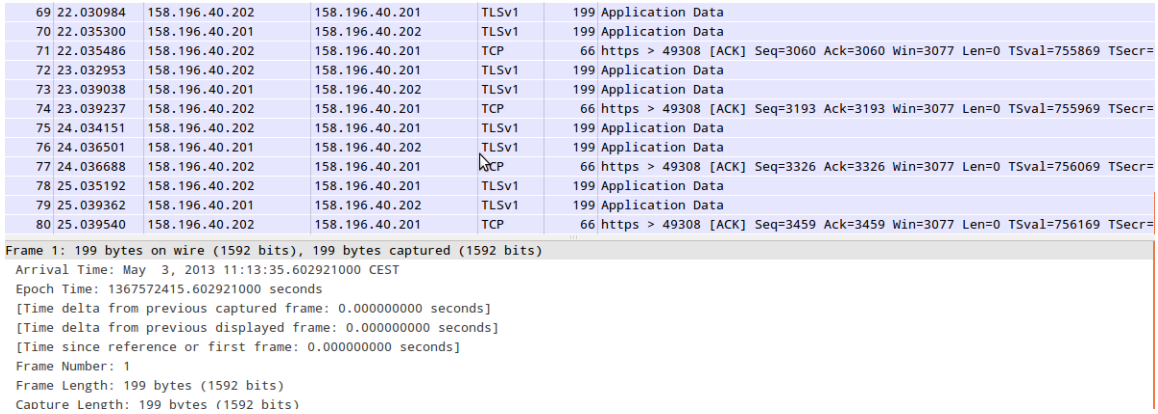
Obr. 15 Prihlásenie do vybranej VPN



Obr. 16 Ohlásenie a stave spojenia

5.6. Testovanie Remote client

Tunel je testovaný pomocou odchytávanania packetov v programe wireshar. Na výpise vidíme šifrovanú komunikáciu, ktorá je rovnaká ako u hhtps protokolu.



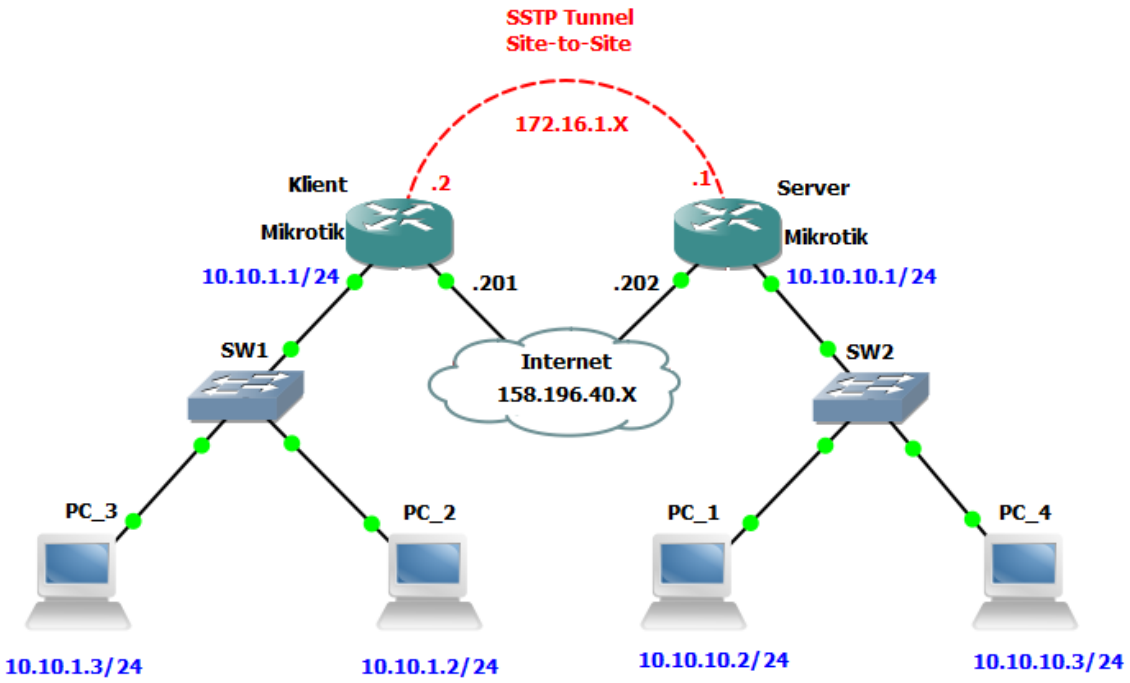
69	22.030984	158.196.40.202	158.196.40.201	TLSv1	199	Application Data
70	22.035300	158.196.40.201	158.196.40.202	TLSv1	199	Application Data
71	22.035486	158.196.40.202	158.196.40.201	TCP	66	https > 49308 [ACK] Seq=3060 Ack=3060 Win=3077 Len=0 TSval=755869 TSecr=
72	23.032953	158.196.40.202	158.196.40.201	TLSv1	199	Application Data
73	23.039038	158.196.40.201	158.196.40.202	TLSv1	199	Application Data
74	23.039237	158.196.40.202	158.196.40.201	TCP	66	https > 49308 [ACK] Seq=3193 Ack=3193 Win=3077 Len=0 TSval=755969 TSecr=
75	24.034151	158.196.40.202	158.196.40.201	TLSv1	199	Application Data
76	24.036501	158.196.40.201	158.196.40.202	TLSv1	199	Application Data
77	24.036688	158.196.40.202	158.196.40.201	TCP	66	https > 49308 [ACK] Seq=3326 Ack=3326 Win=3077 Len=0 TSval=756069 TSecr=
78	25.035192	158.196.40.202	158.196.40.201	TLSv1	199	Application Data
79	25.039362	158.196.40.201	158.196.40.202	TLSv1	199	Application Data
80	25.039540	158.196.40.202	158.196.40.201	TCP	66	https > 49308 [ACK] Seq=3459 Ack=3459 Win=3077 Len=0 TSval=756169 TSecr=

```
Frame 1: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits)
 Arrival Time: May  3, 2013 11:13:35.602921000 CEST
 Epoch Time: 1367572415.602921000 seconds
 [Time delta from previous captured frame: 0.000000000 seconds]
 [Time delta from previous displayed frame: 0.000000000 seconds]
 [Time since reference or first frame: 0.000000000 seconds]
 Frame Number: 1
 Frame Length: 199 bytes (1592 bits)
 Capture Length: 199 bytes (1592 bits)
```

Obr. 17 Testovanie Remote client

5.7. Konfigurácia SSTP (Site-to-Site) Mikrotik

Vzorová konfigurácia, bola vytvorená rovnako na sieťových zariadeniach komerčnej firmy Mikrotik (Routerboard 450g) a koncové stanice nachádzajúce sa v súkromných sieťach boli konfigurované na platforme Linux. Server aj Klient SSTP sú pomocou rozhrania ether3 pripojení do verejnej siete. Súkromné siete na oboch stranách VPN sú pripojené na rozhraniach ether4. Obe súkromne siete sú medzi sebou smerované prostredníctvom SSTP tunela s adresou 172.16.1.1/32. Na oboch stranách tunela, musia byť naimportované certifikáty, kľúče a certifikačná autorita, podobne ako pri konfigurácii Remote Client. Tieto potrebné súbory si taktiež vytvoríme v systéme Linux (Ubuntu 11.10) pomocou knižnice OpenSSL. Taktiež nesmieme zabudnúť na smerovanie medzi potrebnými sieťami. [9]



Obr. 18 Topológia siete Site-to-Site

5.8. Konfigurácia serveru a klienta

Nastavenie dátumu a času na aktuálne hodnoty:

```
[admin@MikroTik] /system clock> set time=6:33:35
[admin@MikroTik] /system clock> set date=apr/30/2012
```

Nastavenie rozhraní na sieťovom prvku Mikrotik Routerboar 450g. Rozhranie ether3 je pripojené do verejnej siete Internet a rozhranie ether4 do súkromnej siete.

```
[admin@MikroTik] //ip address add address=158.196.40.202/24 interface=ether3
[admin@MikroTik] //ip address add address=10.10.10.1/24 interface=ether4
```

Certifikát, kľúč a certifikačnú autoritu nakopirujeme do adresára Files na Mikrotik Routerboar 450g pomocou aplikácie winbox a následujúcimi príkazmi ich naimportujeme. Súbory je potrebné importovať aj na strane klienta, avšak s klientskymi certifikátmi.

```
 [admin@MikroTik] /certificate> import file-name=server.crt
[admin@MikroTik] /certificate> import file-name=server.key
[admin@MikroTik] /certificate> import file-name=ca.crt
```

Po základnej konfigurácii na strane serveru, si vytvoríme užívateľa s heslom, pomocou ktorého sa budeme prihlasovať do VPN siete a nastavíme lokálnu a vzdialenú adresu na ktorú sa bude klient prihlasovať. Do príkazu je zakomponovaná trasa cez ktorú sa môže klient pripojiť. Inak by bolo potrebné nastaviť statické smerovanie na servery , aby bola možná komunikácia medzi klientskou a serverovou časťou.

```
[admin@MikroTik] /ppp secret> add name=Home service=sstp password=123
local-address=172.16.1.1 remote-address=172.16.1.2 routes="10.10.1.0/24 172.16.1.2 1"
[admin@MikroTik] ppp secret> print detail
Flags: X - disabled
 0   name="Home" service=sstp caller-id="" password="123" profile=default
     local-address=172.16.1.1 remote-address=172.16.1.2 routes=="10.10.1.0/24 172.16.1.2 1"
[admin@MikroTik]  /ppp secret>
```

Nastavenie správnych mien a ich kontrola príkazom print:

```
admin@ MikroTik] /certificate>set 0 name=CA
admin@ MikroTik]  /certificate>set 1 name=server
admin@ MikroTik]  /certificate> print
```

Spustenie SSTP servera:

```
[admin@ MikroTik] /interface sstp-server server> set certificate=server
[admin@ MikroTik] /interface sstp-server server> set enabled=yes
[admin@ MikroTik] /interface sstp-server server> set verify-client-certificate=yes
```

Kontrola SSTP servera:

```
[admin@ MikroTik] /interface sstp-server server> print
                  enabled: yes
                     port: 443
                  max-mtu: 1500
                  max-mru: 1500
                     mrru: disabled
          keepalive-timeout: 60
            default-profile: default
                certificate: server
  verify-client-certificate: yes
             authentication: pap,chap,mschap1,mschap2
```

Spustenie SSTP Klienta na klientskej časti VPN spojenia:

```
[admin@ MikroTik] /interface sstp-client> add user=Home password=123 connect-to=158.196.40.202
disabled=no certificate=client verify-server-certificate=yes
```

Kontrola SSTP Klienta

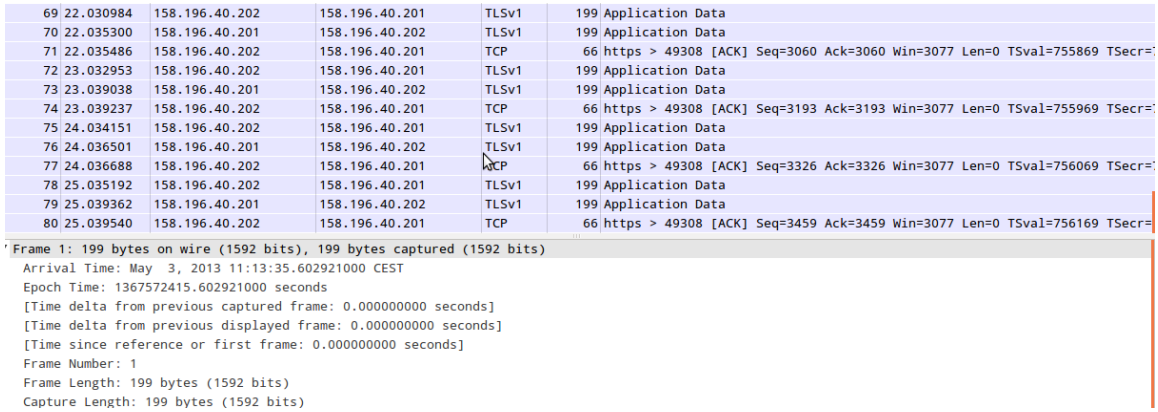
```
[admin@ MikroTik] /interface sstp-client> print
Flags: X - disabled, R - running
 0  R name="sstp-out1" max-mtu=1500 max-mru=1500 mrru=disabled connect-to=158.196.40.202:443
      user="Home" password="123" proxy=0.0.0.0:443 profile=default certificate=client
      keepalive-timeout=60 add-default-route=no dial-on-demand=no
      authentication=pap,chap,mschap1,mschap2 verify-server-certificate=yes
```

Pridanie statického smerovania na klientskej časti VPN:

```
[admin@ MikroTik] /ip route> add dst-address=10.10.10.0/24 gateway=172.16.1.1
```

5.9. Testovanie client server

Tunel je testovaný pomocou odchytávanania packetov v programe wireshark.



Obr. 19 Testovanie client server

6. SSL VPN NA PLATFORME CISCO

Platforma CISCO ponúka pripojenie pomocou SSL VPN v troch základných módoch:

Clientless

V tomto režime vzdialený užívateľ pristupuje do privátnej siete, prostredníctvom internetového prehliadača. Pri tomto druhu komunikácie je zabezpečená, len webová komunikácia.

Thin-Client

Tento spôsob sa nazýva presmerovanie portov. Klientska aplikácia používa TCP spojenie na server a port. Vzdialený užívateľ si stiahne Java, alebo ActiveX applet, ktoré funguje ako TCP proxy pre služby ktoré sú nakonfigurované na stránke. Tento typ zabezpečuje vzdialený prístup aj na newebovú komunikáciu, ako napríklad SMTP, POP3, TELNET, ping, atď.

Tunnel mode

V tomto režime má vzdialený užívateľ ma najširšie možnosti prostredníctvom aplikácii cez dynamicky stiahnuteľného Cisco AnyConnect VPN klienta pre SSL VPN. Spojenie je naviazané na SSL bránu.

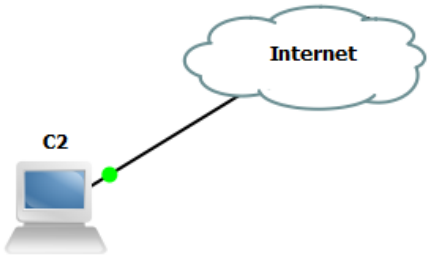
Zhodnotenie jednotlivých módov

Clientless	Thin-Client	Tunnel
Založené na prehliadači Webové aplikácie Zdieľanie súborov Outlook Web	TCP presmerovanie portov Používa Java applet Rozšírená podpora aplikácii Telnet, E-mail, SSH	"clientless" IPsec VPN

6.1. Clientless (webVPN)

Clientless SSL VPN taktiež známa ako WebVPN, technológia, ktorá ponúka obmedzený bezpečnostný prístup na niektoré zdroje v privátne sieti. Keďže sú možnosti obmedzené, môže byť použitá, len k určitému typu nasadenia. Medzi hlavné výhody Clientless SSL VPN je najjednoduchším možným typom. K jeho využívaniu na stačí štandardný webový prehľadávač. s podporov SSL. Klient nepoužíva žiadny iný software. Klient pristupuje pomocou webového prehliadača na webovú stránku (portál), prostredníctvom šifrovacieho protokolu HTTPS

štandardne na porte 443, kde sa prevedie autentizácia následne je sprístupnený portál s nastavenými možnosťami. [3, 11]

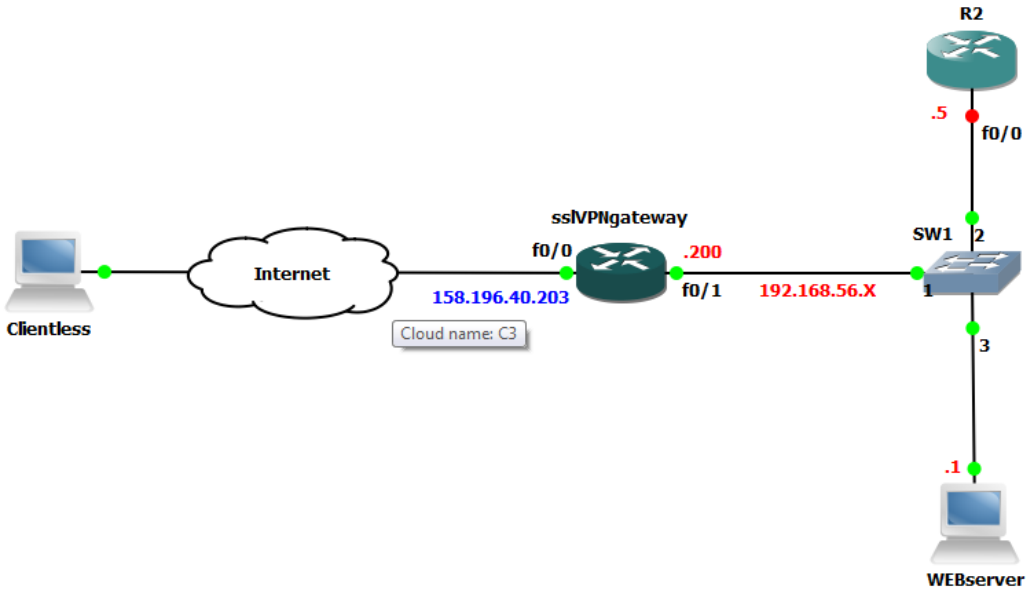


Obr. 20 Clientless

6.1.1. Konfigurácia Clientless SSL VPN (WebVPN)

Clientless SSL VPN umožňuje užívateľovi bezpečne pristupovať k zdrojom na súkromnej sieti, z ľubovoľného miesta. K svojej komunikácii si žiada, len webový prehliadač, ktorý povoľuje SSL. Užívateľ sa najskôr autentizuje na SSL VPN bránu, ktorá následne umožňuje užívateľovi prístup k sieťovým zdrojom. [3, 11]

Príklad konfigurácie Clientless SSL VPN bol aplikovaný podľa topológie na obrázku obr.18. Pri konfigurácii boli použité sieťové zariadenia Cisco 3725, ktoré boli konfigurované prostredníctvom konzoly a Security Device Manager (SDM 2.5) [3, 11]



Obr. 21 Topológia Clientless

Konfigurácia sslVPNgateway

Nastavenie interface

R1>enable
R1#configure terminal
R1(config)#hostname sslVPNgateway
sslVPNgateway (config)#interface fastEthernet 0/0
sslVPNgateway (config-if)#ip address 158.196.40.203 255.255.255.0
sslVPNgateway (config-if)#no shutdown
sslVPNgateway (config)#interface fastEthernet 0/1
sslVPNgateway (config-if)#ip address 192.168.56.200 255.255.255.0
sslVPNgateway (config-if)#no shutdown

Konfigurácia HTTP serveru

sslVPNgateway (config)#ip http server
sslVPNgateway (config)#username peter privilege 15 password cisco
sslVPNgateway (config)#ip http authentication local

Konfigurácia SSH

sslVPNgateway (config)#ip domain-name vsb.cz
sslVPNgateway (config)#crypto key generate rsa
sslVPNgateway (config)#username peto secret cisco
sslVPNgateway (config)#line vty 0 4
sslVPNgateway (config-line)#login local
sslVPNgateway (config-line)#transport input ssh

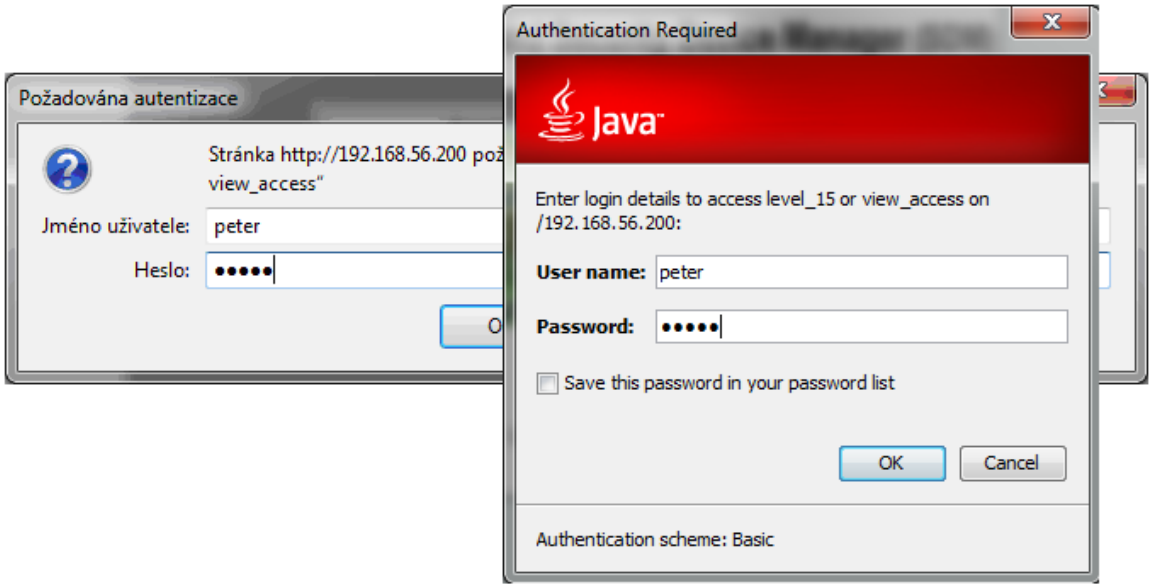
Nastavenie aktuálneho dátumu a času

sslVPNgateway#clock set 16:14:50 april 20 2012

Po nakonfigurovaní základných príkazov ako je nastavenie rozhraní, času a povolenie HTTP serveru prostredníctvom ktorého budeme pristupovať na VPN bránu je potrebné spustiť aplikáciu SDM.

Obr. 22 Pripojenie na rozhranie fa0/1

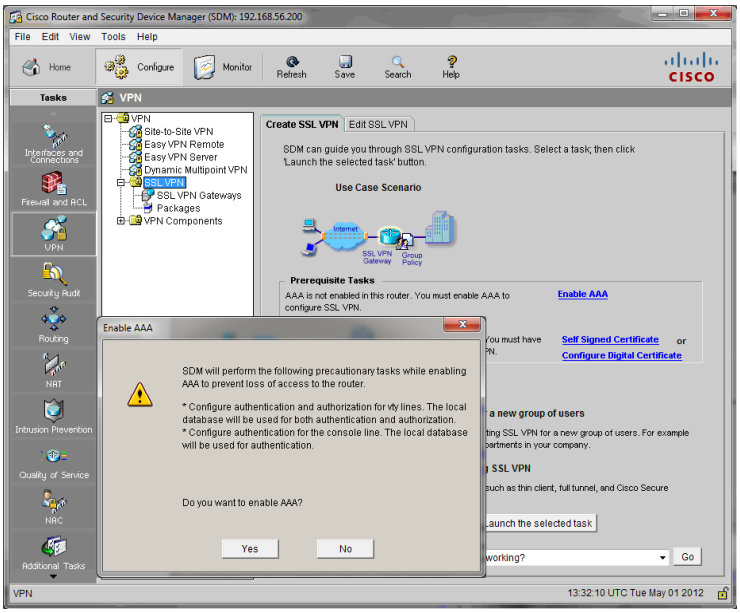
Zadáme adresu prostredníctvom ktorej bude konfigurácia prebiehať. Neodporúča sa použitie rozhrania, na ktorom bude bežať VPN brána. Následne vyplníme povinné prihlasovacie údaje užívateľa s privilegovaným účtom 15.



Obr. 23 Prihlásenie privilegovaným užívateľom

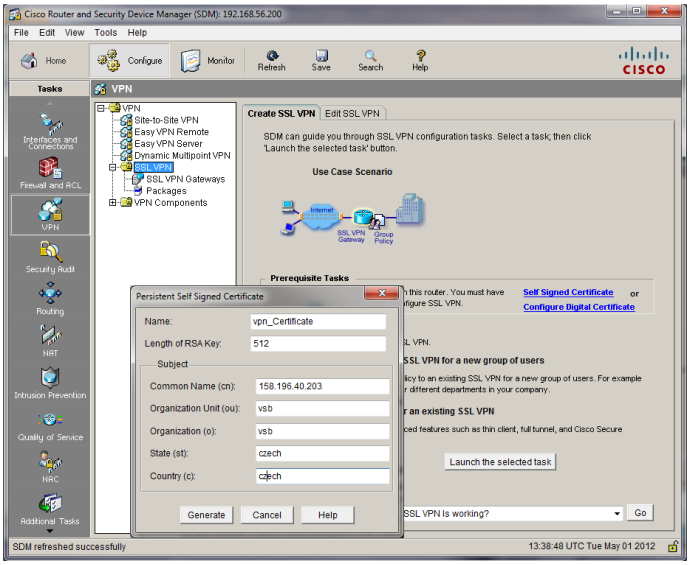
V konfigurácii pomocou SDM pokračujeme nasledujúcimi krokmi:

Krok 1: Povolenie AAA a vytvorenie Certifikátu kliknutím; v ľavom panely označíme možnosť SSL VPN a v hlavnom okne klikneme na Enable AAA a Self Signed Certificate



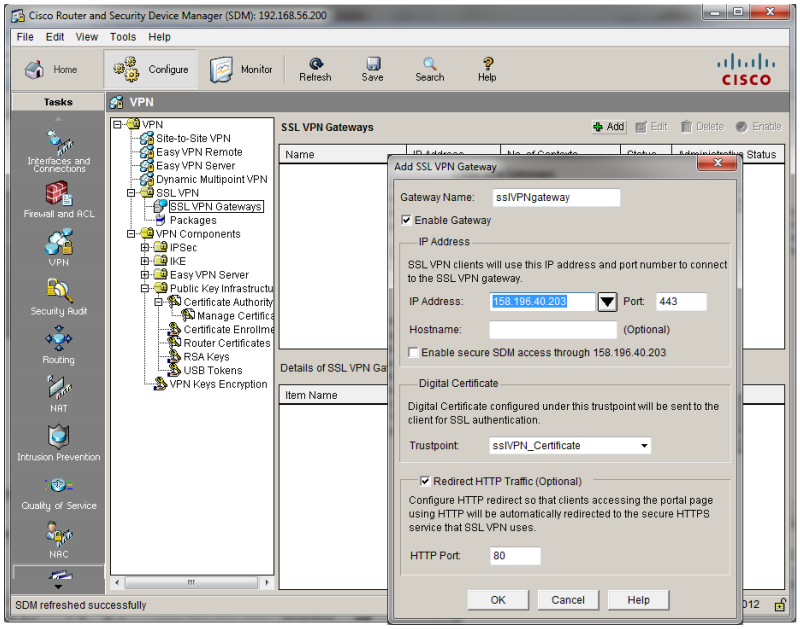
Obr. 24 Povolenie AAA

Pri vytváraní bezpečnostného certifikátu, ktorý bude pri komunikácii poskytnutý užívateľovi vyplníme všetky atribúty. Pri zázname CN je potrebné zadať adresu brány, alebo doménové meno brány.



Obr. 25 Vytvorenie bezpečnostného certifikátu

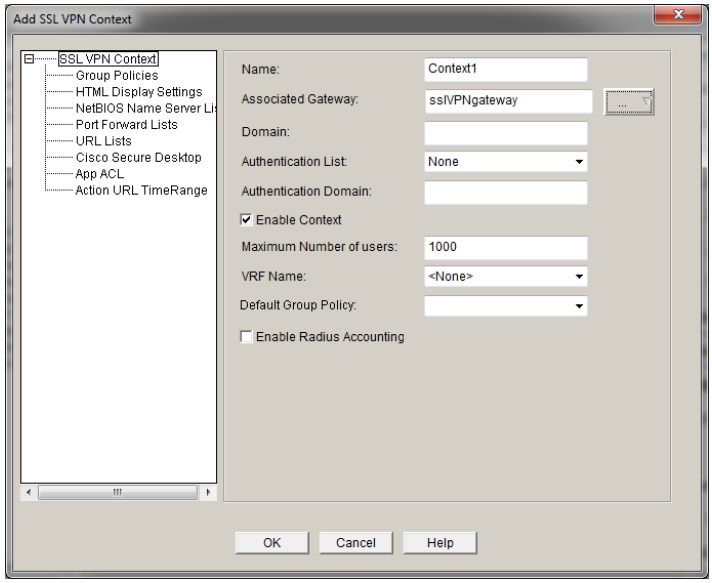
Krok 2: Z ponuky vyberieme Configure -> VPN ->sslVPN->SSL VPN Gateways -> Add ->Launch the selected task. Po objavení formulára je potrebné povoliť možnosť vytvorenia Brány. Zadáme jednoznačne meno brány, IP adresu, ktorá bude vstupnou bránou a zároveň musí táto adresa patriť do verejnej siete. Následne vyberieme dôveryhodný certifikát, ktorý bude poslaný SSL VPN klientovi. [3]



Obr. 26 Vytvorenie brány Clientless

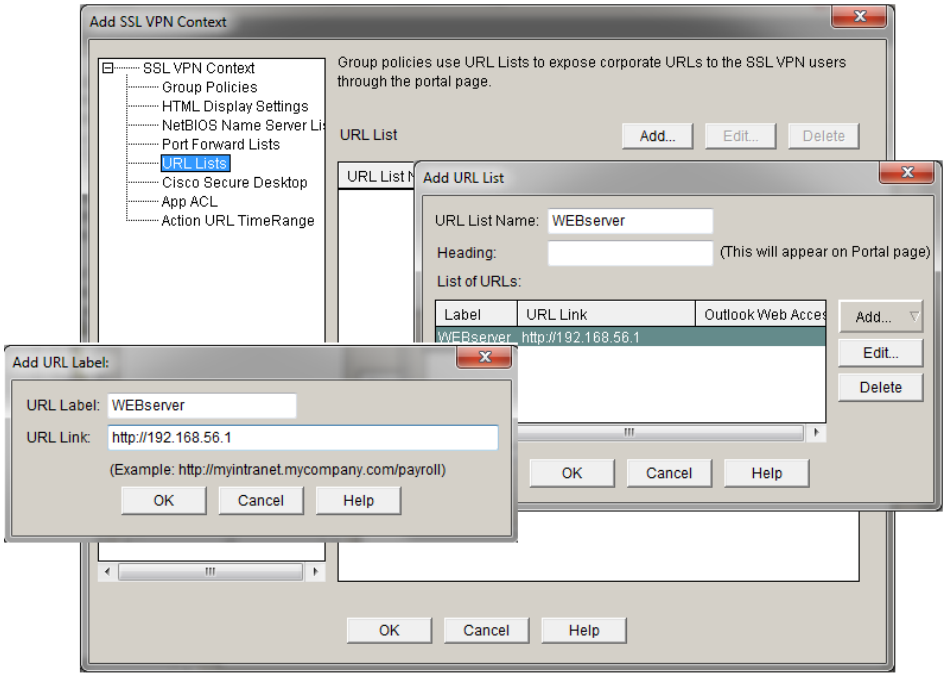
Krok 3: Po vytvorení SSL VPN brány, je potrebné nadefinovať politiku skupín. Klikneme na adresár SSL VPN a v hlavnom okne sa prepnutím do záložky Edit VPN a zvolením Add dostaneme k možnosti definovania politík. SSL VPN Kontext je miesto, kde je ukončená SSL VPN a miesto kde dochádza k vytvoreniu relácie s VPN užívateľom. Kontext tiež obsahuje všetky politiky použité pre klienta, vrátane autentizácie, autorizácie, alebo skupinovej politiky. Web VPN Kontext používa bránu ako koncový bod pre relácie. Jedna brána môže používať aj viac kontextov ako jeden, ale musí využívať značkovanie.

Vyplnenie formulára Kontextu:



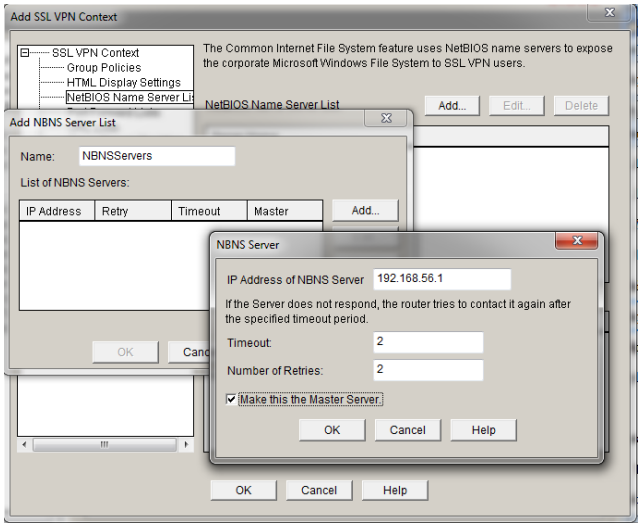
Obr. 27 Povolenie Kontextu

Krok 4: Pre nastavenie URL zoznamu je potrebné vybrať v ľavej lište VPN kontextu položku URL lists. Pokračujeme tlačidlom Add a vyplníme všetky potrebné informácie, ktoré si URL list žiada. Pomocou tlačidiel Add, Edit, Delete je možné spravovať dané zoznamy.



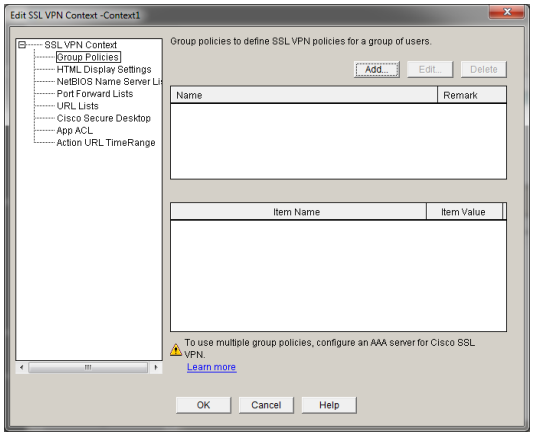
Obr. 28 Naplnenie URL zoznamu

Krok 5: Rozbalení SSL VPN context, klikneme na položku NetBios Name Server List ->Add. Na vytvorenie alebo zachovanie NBNS listu serverov musíme zadať názov každého listu, časový limit a počet opakovaní pre každý server, ktorý vytvoríme. Jeden server v každom tomto zozname musí byť nastavený ako Master.



Obr. 29 Naplnenie NetBios Server zoznamu

Krok 6: Definovanie politiky pod názvom policy1. Rozbalíme položku Context -> Group Policies -> Add. SSL VPN politiky skupín definujú portály a odkazy pre užívateľov, ktoré boli pridané do týchto politík. Ak vzdialený užívateľ zadá url SSL VPN, musí smerovač určiť, ktorej politiky je užívateľ členom, aby bolo možné zobraziť portál, ktorý bol nakonfigurovaný pre danú politiku. Ak je nakonfigurovaná len jedna politika, môže užívateľa overovať lokálne, alebo pomocou AAA servera a následne zobrazí portal.



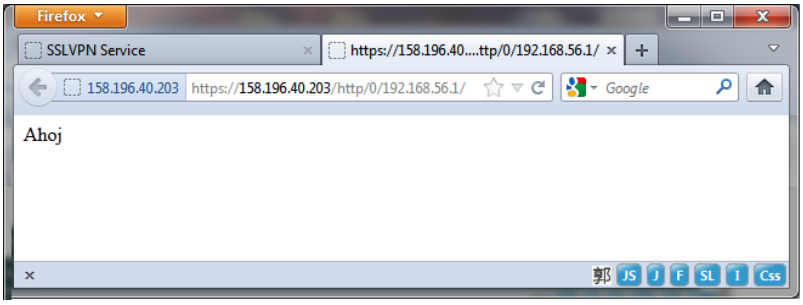
Obr. 30 Vytvorenie skupinovej politiky

6.1.2. Testovanie módu Clientless pomocou webového prehliadača

Obr. 31 Testovanie módu Clientless

Po úspešnom prihlásení na webový portál brány a schválení bezpečnostného certifikátu sa objaví úvodný portál na bezpečnostnej bráne. Zadaním adresy do portálu na bezpečnostnej bráne, budeme pristupovať k žiadaným dátam.

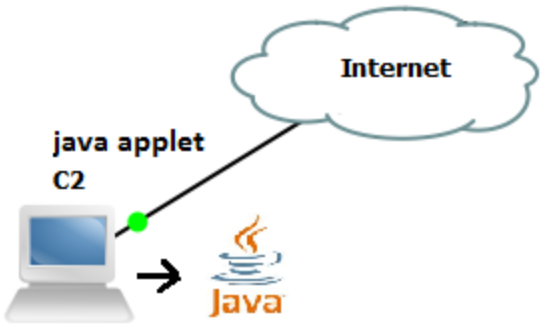
Obr. 32 Úspešné prihlásenie a zadanie cieľovej adresy



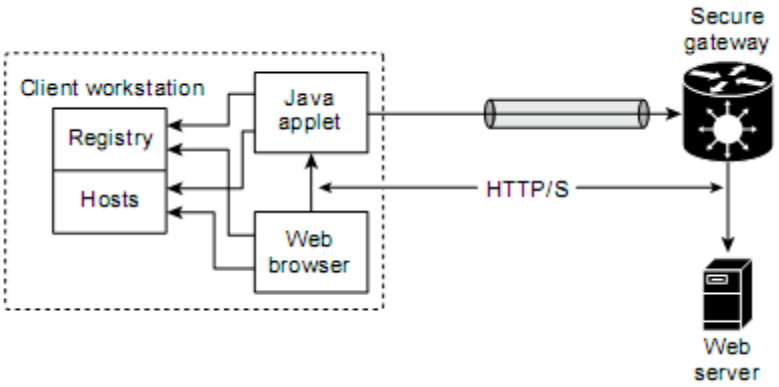
Obr. 33 Úspešné premostenie na WEBserver

6.2. Thin-Client (Port Forwarding)

Tento spôsob sa nazýva aj ako presmerovanie portov. Klientska aplikácia používa TCP spojenie na server a port. Vzdialený užívateľ si stiahne Java, alebo ActiveX applet, ktoré funguje ako TCP proxy pre služby ktoré sú nakonfigurované na stránke. Tento typ zabezpečuje vzdialený prístup aj na newebovú komunikáciu, ako napríklad SMTP, POP3, TELNET, ping, atď. Nevýhodou tejto techniky je nutnosť správy systému mapovania portov na cieľové adresy a porty serverov a ich aplikácii. Ďalším problémom, je nutnosť použitia fixných portov, s ktorými môže nastať pri niektorých službách veľký problém. [3, 12]



Obr. 34 Thin-Client

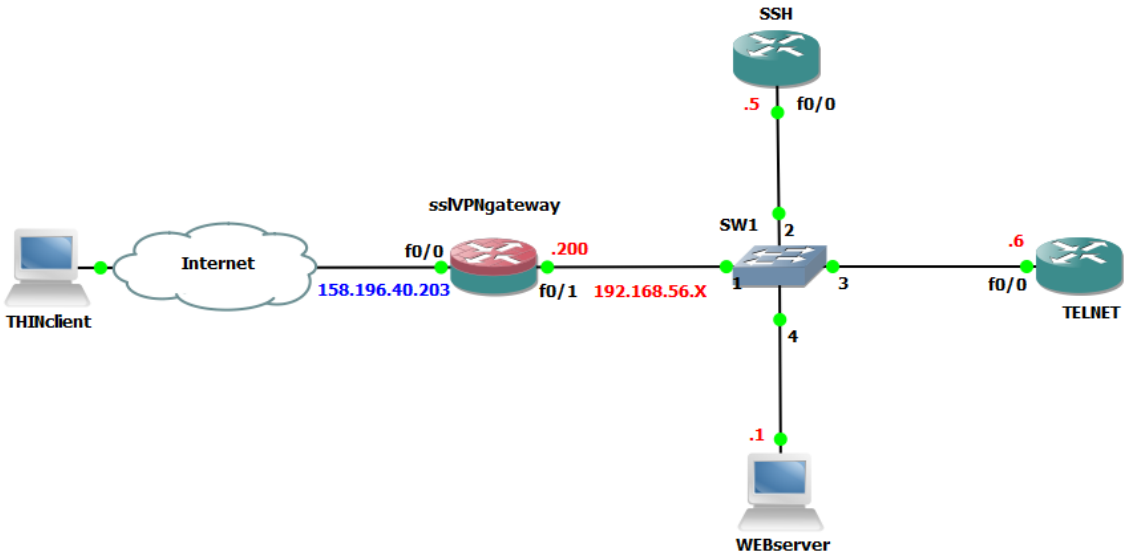


Obr. 35 Komunikácia Thin-Client spolu s bezpečnostnou bránou

6.2.1. Konfiguracia Thin-Client SSL VPN

Pri tomto móde SSL VPN si musí vzdialený klient stiahnuť malý Java applet, ktorý umožňuje bezpečnostnú komunikáciu prostredníctvom TCP aplikácie, použitím statických čísiel portov. Táto metóda SSL VPN nefunguje s aplikáciami, ktoré používajú dynamické prideľovanie portov, napríklad rôzne FTP aplikácie. [10]

Príklad konfigurácie Thin-Client SSL VPN bol aplikovaný podľa topológie na obrázku obr.33. Pri konfigurácii boli použité sieťové zariadenia Cisco 3725, ktoré boli konfigurované prostredníctvom konzoly a Security Device Manager (SDM 2.5) [3, 12]

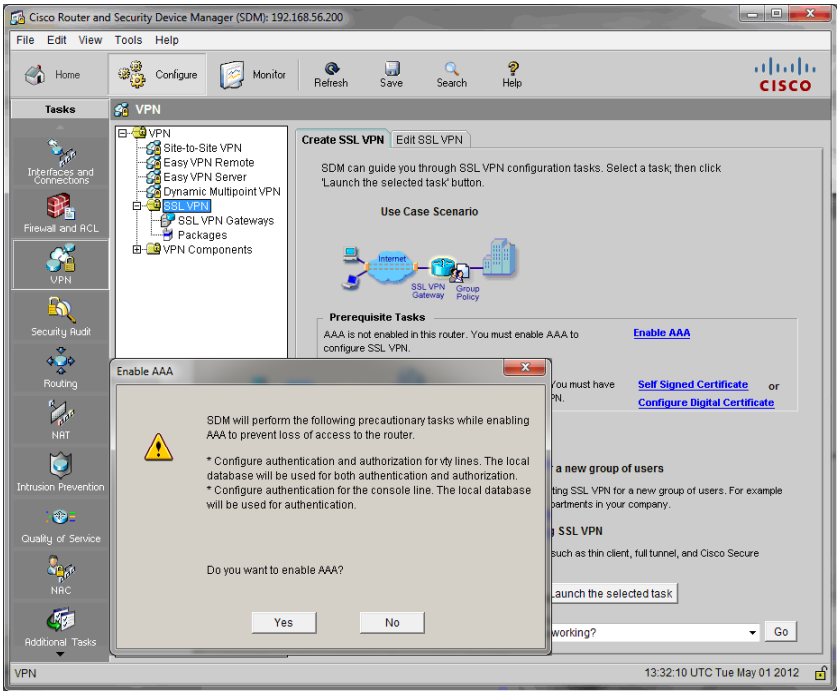


Obr. 36 Topológia Thin-Client (Port Forwarding)

Tak ako v predchádzajúcom Client-less móde je potrebné nastaviť rozhrania, čas a HTTP server. Tieto príkazy je možné nájsť v kapitole 6.1.1

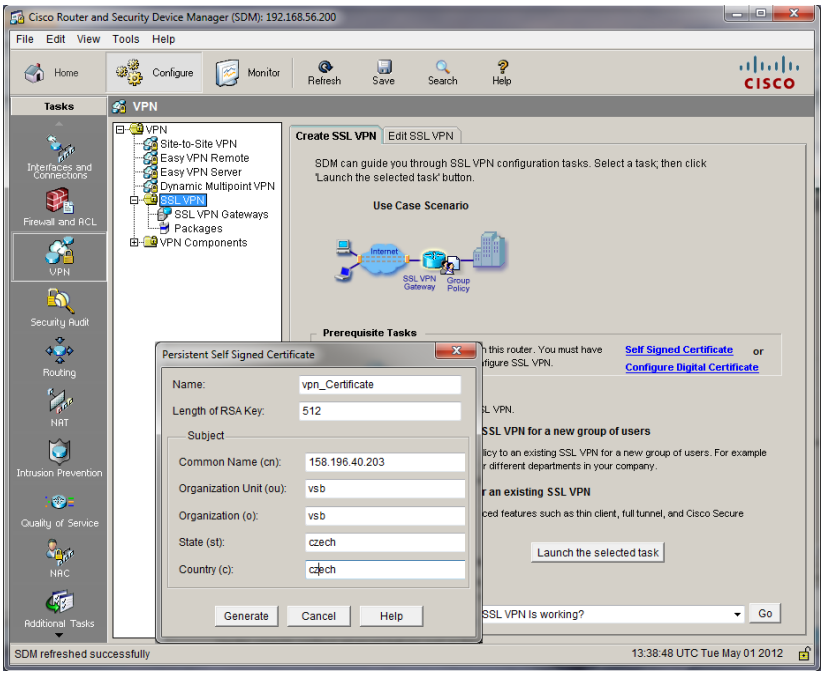
Po nakonfigurovaní týchto základných príkazov a spustení aplikácie SDM pokračujeme nasledujúcimi krokmi“

Krok 1: Povolenie AAA, vytvorenie certifikátu kliknutím v ľavom panely na možnosť SSL VPN->a v hlavnom okne Enable AAA a Self Signed Certificate



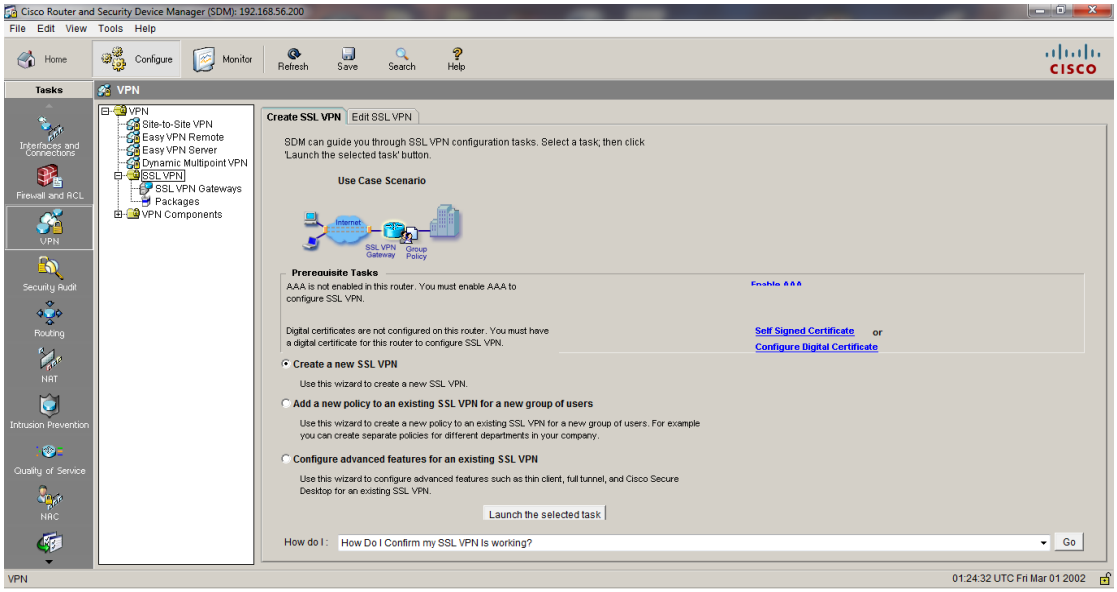
Obr. 37 Povolenie AAA

Pri vytváraní bezpečnostného certifikátu vyplníme, všetky atribúty. Pri zázname CN je potrebné zadať adresu brány, alebo doménové meno brány. Tento certifikát bude pri komunikácii poskytnutý Thin-Clientovi.



Obr. 38 Vytvorenie bezpečnostného certifikátu

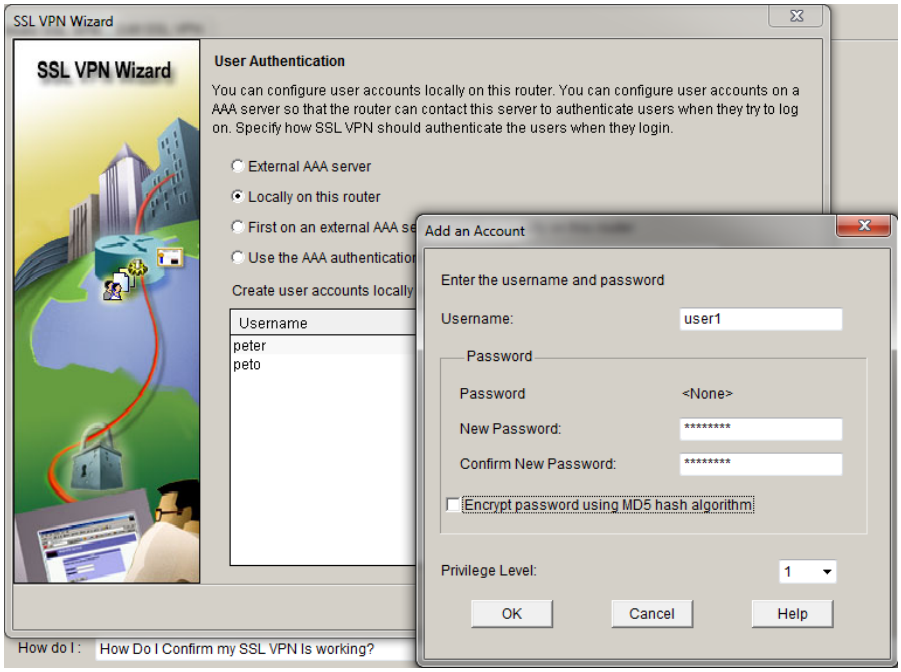
Krok 2: Z ponuky vyberieme Configure -> VPN ->sslVPN->Create -> Add ->Launch the selected task. Po objavení formulára je potrebné povoliť možnosť vytvorenia Brány. Zadáme jednoznačne meno brány, IP adresu, ktorá bude vstupnou bránou a zároveň musí táto adresa patriť do verejnej siete. Následne vyberieme dôveryhodný certifikát, ktorý bude poslaný SSL VPN klientovi.



Obr. 39 Vytvorenie novej SSL VPN

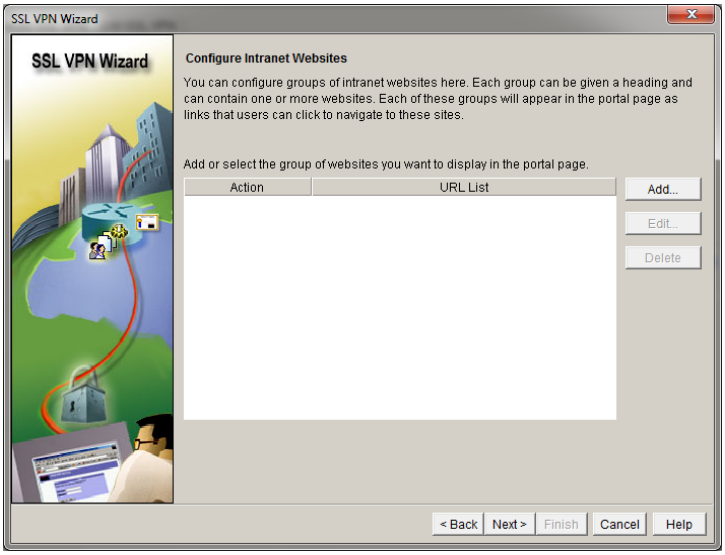
Krok 3: Vyplnenie polí k vytvoreniu SSL VPN brány. Do poľa IP adries vložíme adresu SSL VPN brány, prostredníctvom ktorej budeme pristupovať do súkromnej siete. Rozhranie na ktoré sa budeme prihlasovať, musí patriť verejnej sieti, aby bola dosiahnuteľná pre všetkých klientov. Zakážeme bezpečný prístup k IP adrese, z dôvodu potrebného prístupu prostredníctvom SDM. Ak vytvárame novú bránu, je potrebné vybrať digitálny certifikát, ktorý bude serverom predložený klientovi pri prihlásení do SSL VPN brány. Ak zvolíme IP adresu existujúcej brány, bude smerovač používať digitálny certifikát nakonfigurovaný pre danú bránu, čím je toto pole zakázané. V informačnej zóne sú zobrazené informácie o jej nastavení.

Krok 4: Konfigurácia užívateľských účtov na lokálnom smerovači. Overovanie Užívateľov bude na lokálnom smerovači, takže zvolíme túto možnosť autentizácie. Tlačidlom Add a Edit pridávame alebo upravujeme užívateľské účty na lokálnom smerovači



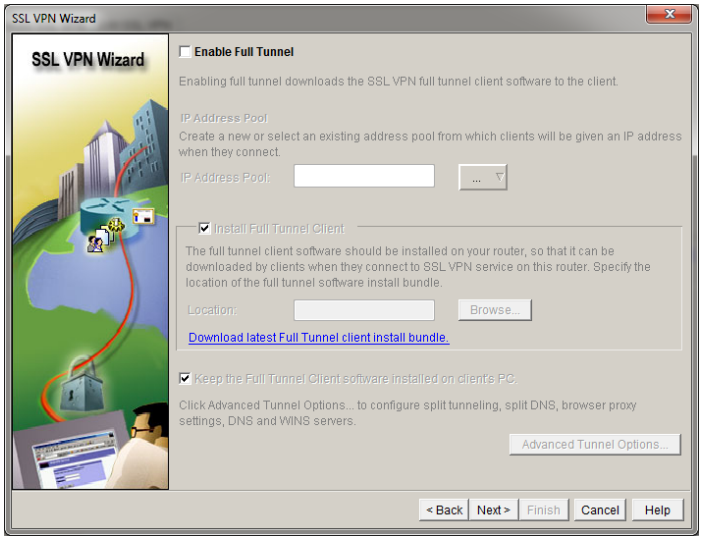
Obr. 40 uživateľske účty Thin-Client

Nasledujúci sprievodca nám ponúka konfiguráciu intranetových web stránok. Tento krok vynecháme, pretože Port-Forwarding používa aplikačný prístup.



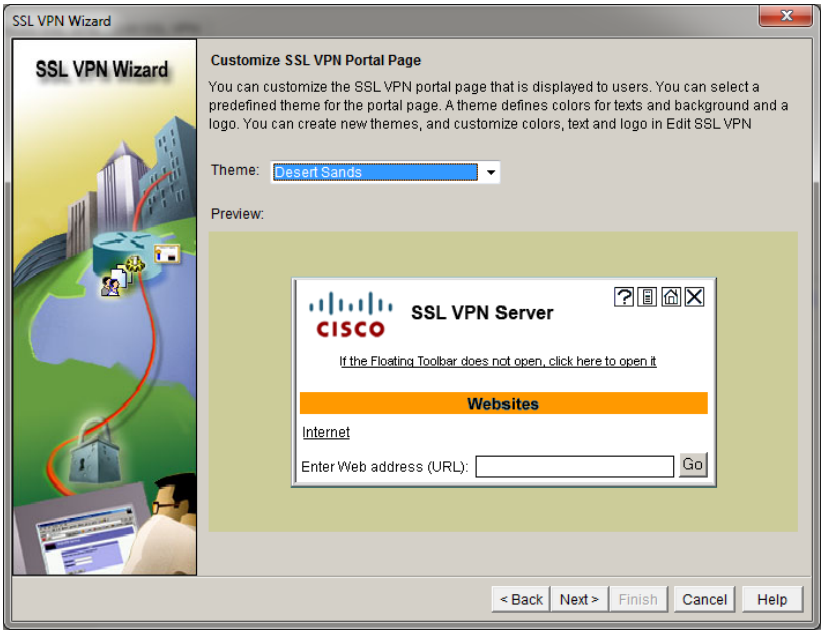
Obr. 41 Nastavenie intranetu

Krok 5: Zakázanie Full Tunnel módu a nastavenie úvodného portálu SSL VPN brány. Keďže nastavujeme mód Port Forwarding, tak v tomto kroku zakážeme možnosť Full Tunnel a pokračujeme tlačidlom next. Nastavenie vzhľadu portálu SSL VPN brány potvrdíme tlačidlom Next.



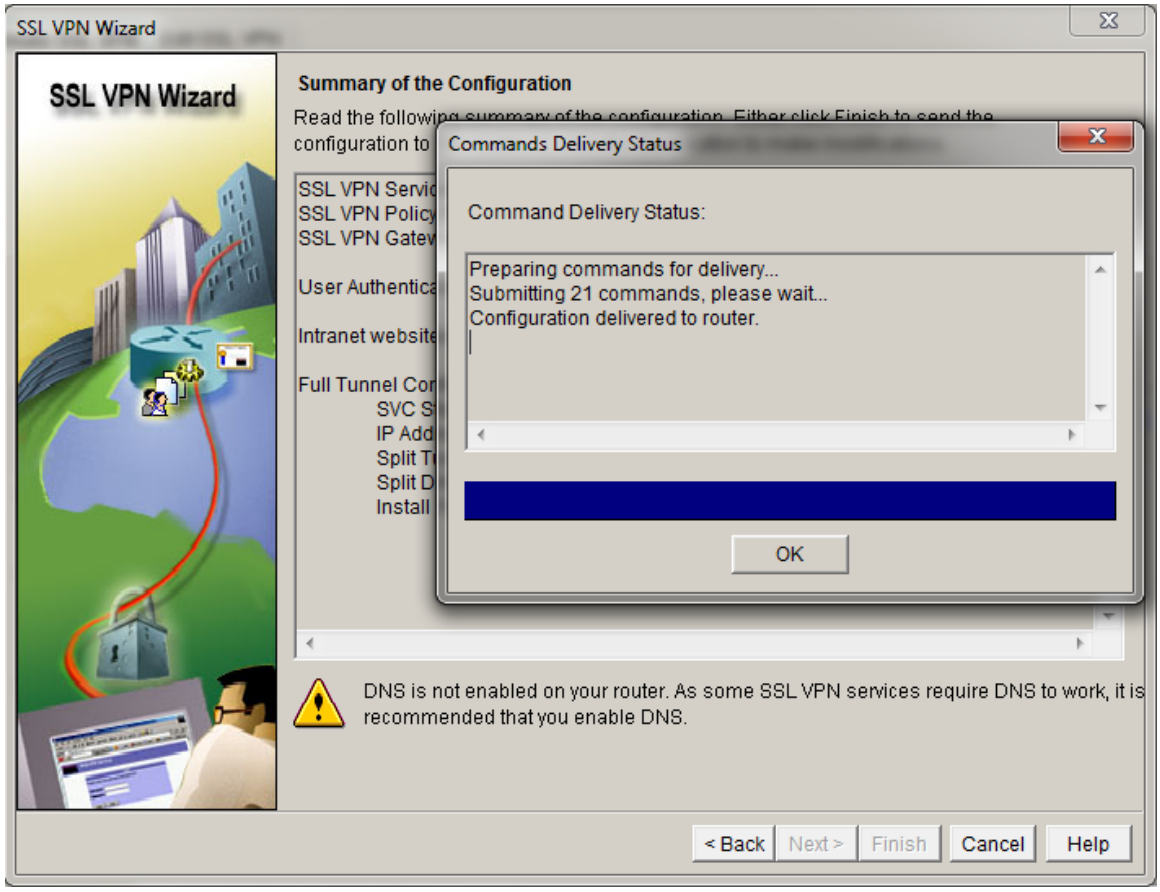
Obr. 42 Zakázanie módu full tunnel

Nastavenie úvodnej webovej stránky po prihlásení na ssl VPN bránu



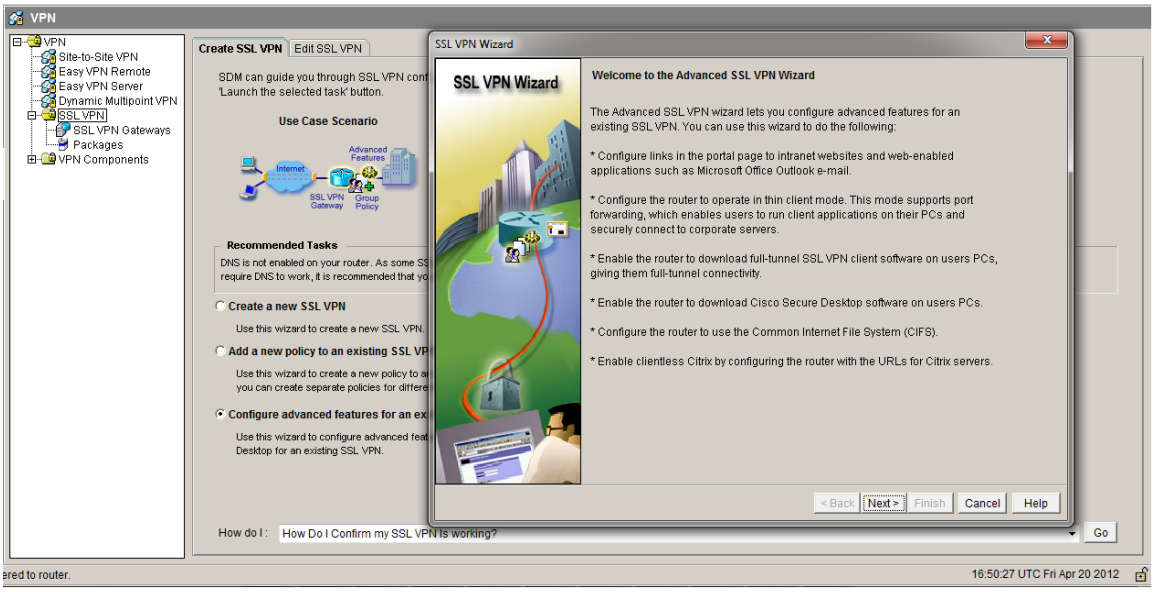
Obr. 43 Nastavenie úvodnej stránky portálu SSL VPN brány

Krok 6: Dokončenie konfigurácie, sa uskutoční potvrdením tlačidla Finish a oznamovacím oknom o jeho potvrdení príkazov.

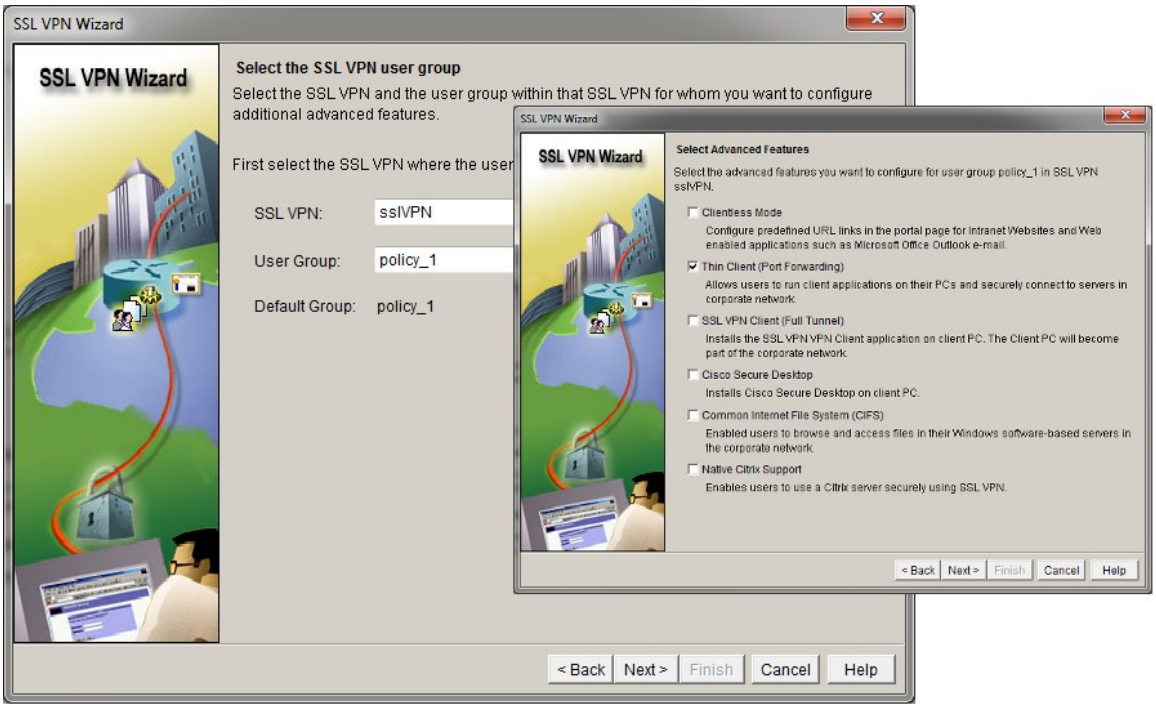


Obr. 44 **Dokončenie konfigurácie**

Krok 7: Po vytvorením SSL VPN brány SSL VPN kontextu a skupinovej politiky, musíme nakonfigurovať Thin-Client porty, pomocou ktorých sa klienti pripájajú na SSL VPN. Z ponuky vyberieme Configure -> VPN ->sslVPN->Create -> Configure ->Launch the selected task. Po zobrazení informačného panelu, potvrdíme tlačidlom Next. Vybraním SSL VPN a užívateľskej skupiny, a následne zaškrtneme výber Thin-Client, pomocou ktorého zadáme IP adresy a prístupové porty cez ktoré budeme pristupovať do súkromnej siete. Porty pri móde Thin-Client je potrebné nastaviť fixne, čo je považované za nevýhodu, z dôvodu vzniknutia možného problému pri niektorých službách, ako je napríklad FTP. [3, 12]



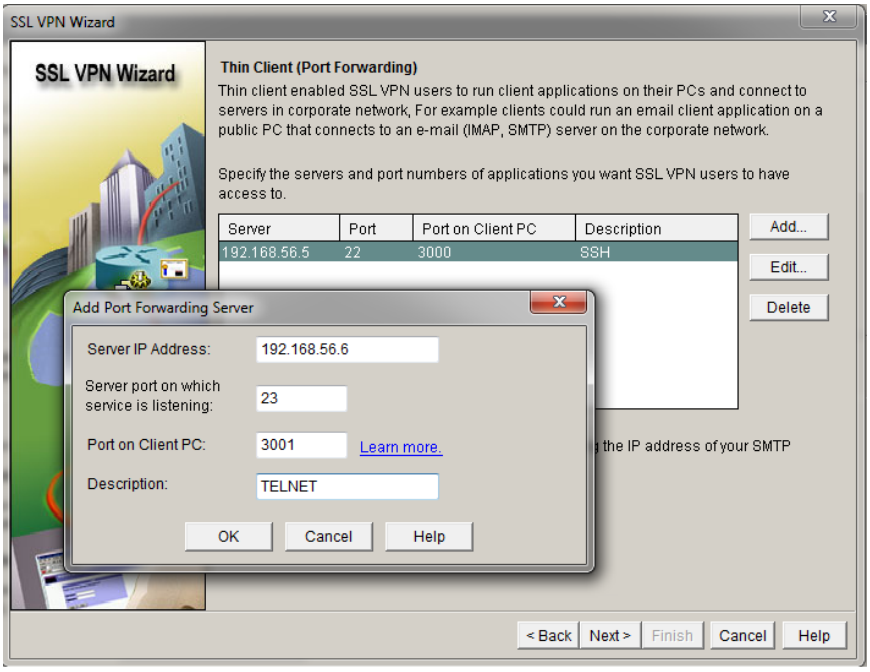
Obr. 45 Konfigurácia Port Forwarding 1



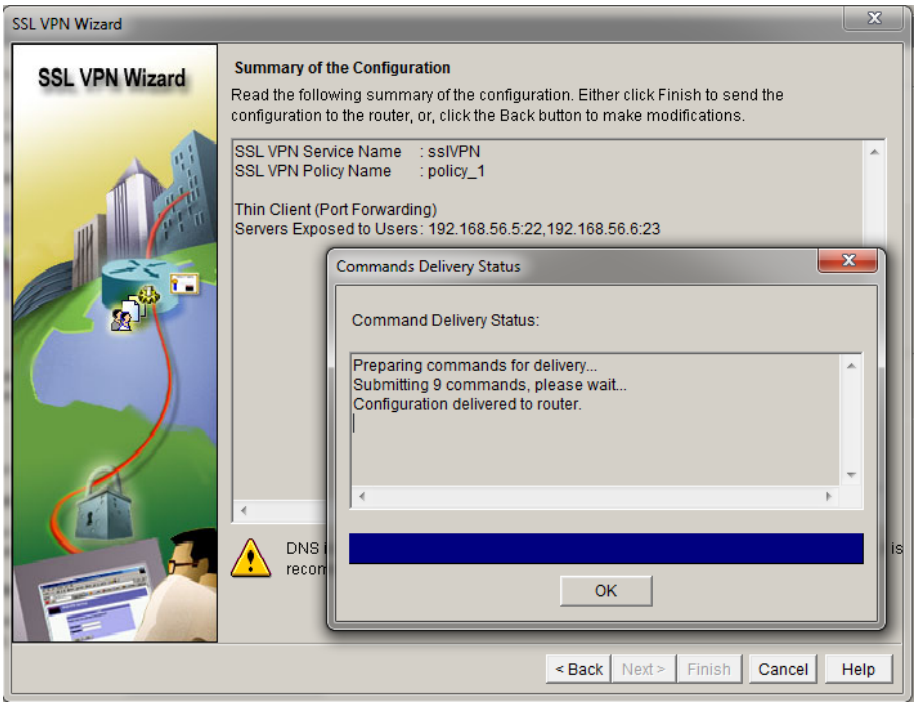
Obr. 46 Konfigurácia Port Forwarding 2

Port serveru, ktorý počúva na službu, môže byť štandardný ako ho definuje IANA, ale je možne zadať aj vlastný port. Port Forwarding sa umožňuje pripojiť vzdialenému užívateľovi na statické porty servera pomocou privátnej IP adresy. Napríklad, ak užívateľ používa aplikáciu ktorá je

pripojená na port 23 adresy 192.168.56.6 je požiadavka odoslaná na adresu 127.0.0.1 a port 3001. Po pridaní adries a portov potvrdíme akciu.

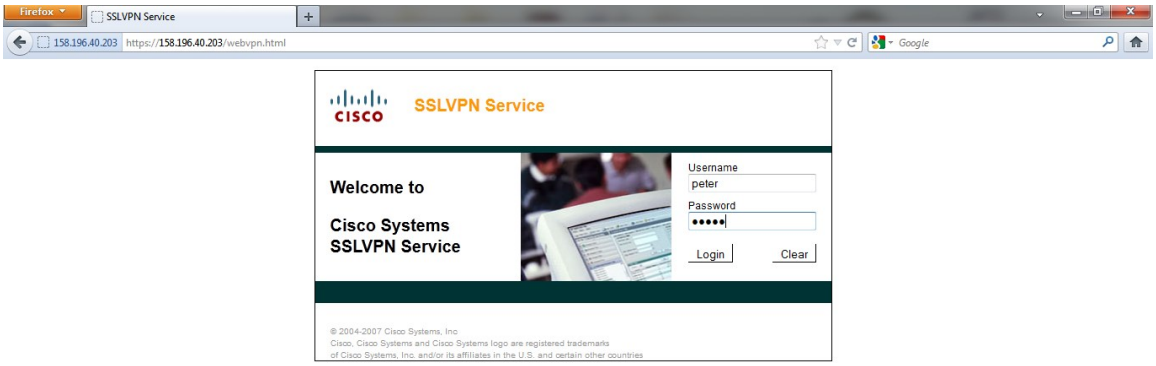


Obr. 47 Pridanie adresy a statického portu



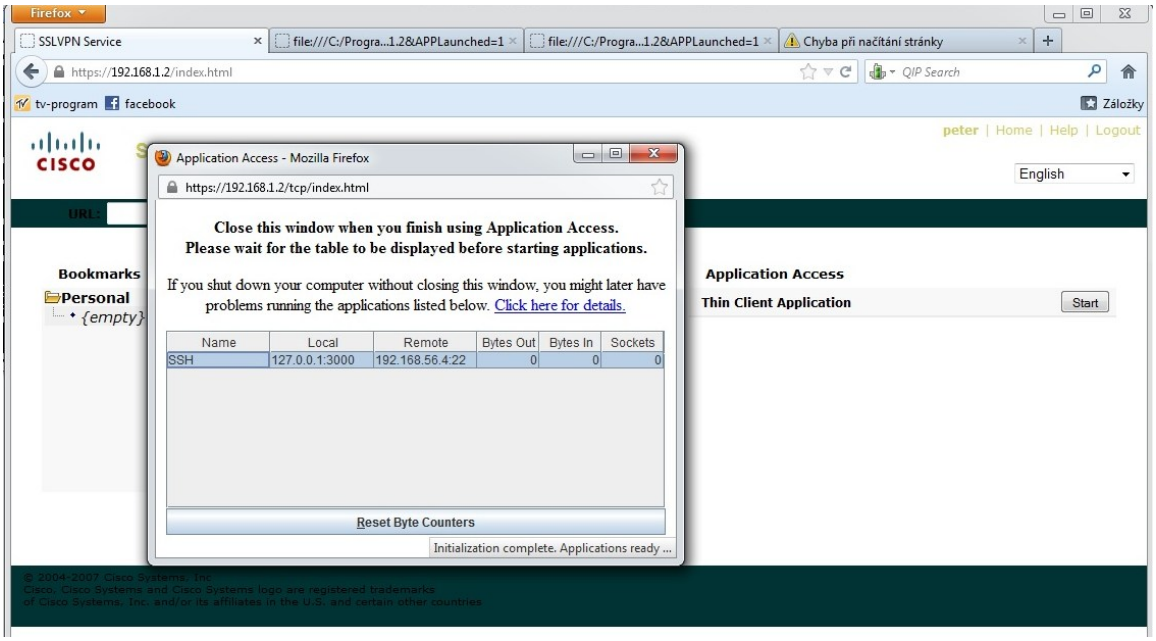
Obr. 48 Dokončenie konfigurácie port forwarding

6.2.2. Testovanie prostredníctvom webového prehliadača



Obr. 49 Úvodná prihlasovacia stránka

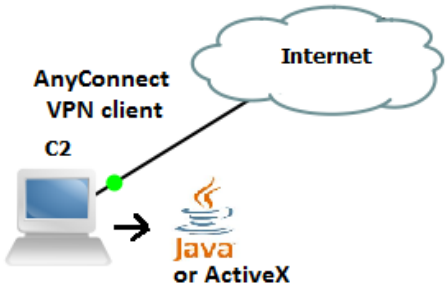
Po úspešnom prihlásení na SSL vpn bránu, nám je zobrazená tabuľka zo zoznamom Port Forwarding.



Obr. 50 Zobrazenie Port Forwarding

6.3. SSL VPN Tunnel Mode

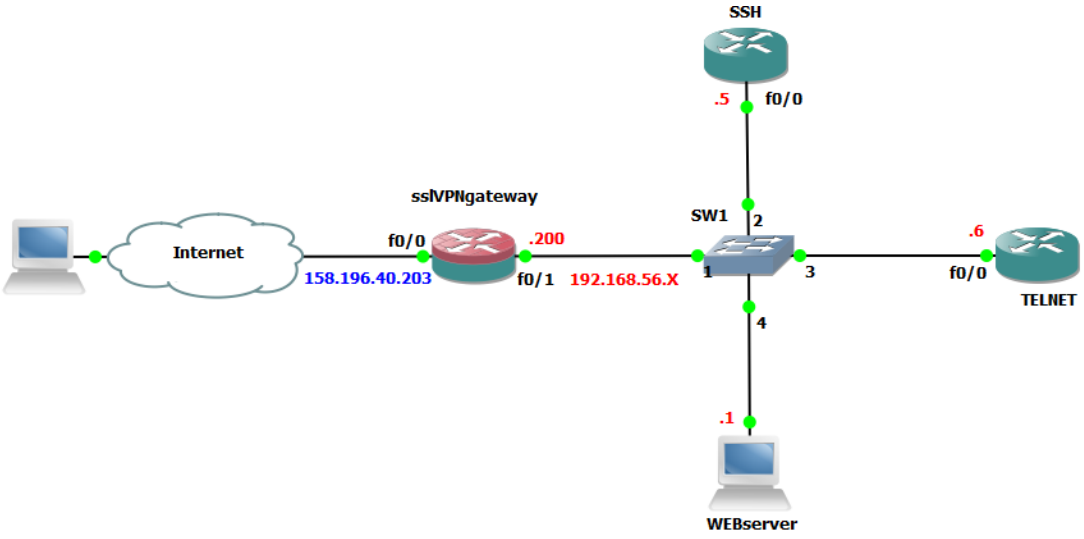
V tomto režime má vzdialený užívateľ ma najširšie možnosti prostredníctvom aplikácii cez dynamicky stiahnuteľného Cisco AnyConnect VPN klienta pre SSL VPN. Klient poskytuje virtuálny prístup k sieťovej vrstve rôznym aplikáciám. Tento typ poskytuje prístup ku stále zväčšujúcej sa množine bežne dostupných aplikácii ako je prehľadávanie stránok, služieb cez web (prístup k súborom), e-mail a aplikácie založené na TCP. Prináša dostupnosť množstva internetových aplikácii bez nutnosti ich inštalácie. Spojenie je naviazané na SSL bránu[13]



Obr. 51 Tunnel Mode

6.3.1. Konfigurácia Tunnel mode

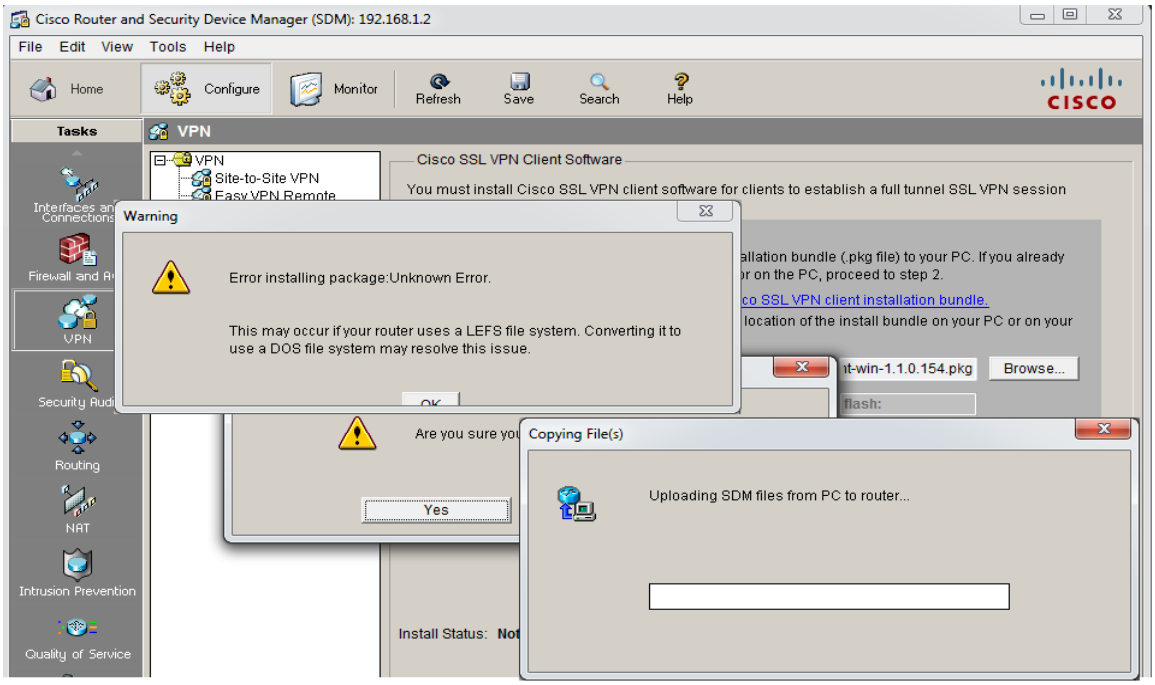
Príklad konfigurácie Tunnel SSL VPN bol aplikovaný podľa topológie na obrázku obr.49. Pri konfigurácii boli použité sieťové zariadenia Cisco 3725, ktoré boli konfigurované prostredníctvom konzoly a Security Device Manager (SDM 2.5) [3, 12]



Obr. 52 Topológia Tunnel-mode

Tak ako v predchádzajúcom Client-less móde je potrebné nastaviť rozhrania, čas a HTTP server. Tieto príkazy je možné nájsť v kapitole 6.1.1

Po nakonfigurovaní týchto základných príkazov, spustení aplikácie SDM, povolení AAA a vytvorenia certifikátu je potrebné na smerovač nainštalovať SVC (SSL VPN Client), ktorý je nutný pri pripojení klienta k bráne. Avšak problém nastal v importovaní a inštalácii balíčka na smerovač, čím nebolo možné v konfigurácii pokračovať.



Obr. 53 Error sprava

7. ZROVNANIE JEDNOTLIVÝCH RIEŠENÍ. ZHODNOTENIE VÝHOD A NEVÝHOD ICH POUŽITIA

V úvode tejto práce som spomínal výhody a nevýhody VPN SSL a VPN a iných vrstvách RM OSI. V tejto kapitole by som ale chcel zhodnotiť jednotlivé typy SSL VPN a to hlavne voči platformám Open source, MikroTik a Cisco.

SSTP

SSTP (Secure Socket Tunneling Protocol) pracuje spoľahlivo aj miestach kde sú VPN pripojenia zakázané. Medzi takéto krajiny, ktoré zakazujú používanie VPN patrí napríklad Belize. SSTP protokol využíva rovnaký port 443 ako SSL prenosy, čím umožňuje špeciálnym spôsobom formovať pakety a prechádzať skrz proxy a firewaly. Tento protokol je považovaný za najbezpečnejší protokol so skupiny protokolov VPN, pretože používa SSL certifikačnú autentifikáciu a 2048 bitové šifrovanie.

Medzi hlavnú výhodu, by som rád poukázal na nízku cenovú dostupnosť zariadení MikroTik a ako hlavnú nevýhodu SSTP považujem, že bol vytvorený výhradne spoločnosťou Microsoft a oficiálne je podporovaný len vyšších verziách ako Windows Vista sp1 a zatiaľ spoločnosť nemá záujem o jeho sprístupnenie pre užívateľ Mac OS, alebo Linux. Aj napriek tomu, po preštudovaní niekoľkých článkov a fór na internete sa mi protokol SSTP podarilo nainštalovať pomocou niekoľkých balíkov ako to môžete vidieť v kapitole 5.3.2.

Open source

Tak ako už napovedá anglický názov Open source, je to jedná z možnosti využívať VPN bez softwaru chránených autorským právom ako napríklad u spoločnosti Microsoft. OpenVPN používaný technológiu SSL pracuje nielen na systémoch ako Mac OS, Linux, alebo Microsoft, ale aj na niektorých IP telefónoch. Pre tento protokol síce stanovila IANA štandardný port 1194 na TCP aj UDP, ale môže taktiež nastavený na porte 443 s HTTPS prenosom. Naviac dokáže prenášať viac kanálov cez jeden TCP, alebo UDP port. OpenVPN nemá žiadny problém

s prekonávaním NAT, alebo proxy serverov. Medzi najväčšie nevýhody patrí veľká latencia zapríčinená systémom.

Cisco

Konfigurácia Cisco smerovačov pomocou aplikácie SDM značne zjednodušuje a zlepšuje produktivitu správy siete. SDM zrýchľuje a uľahčuje nasadenie Cisco smerovačov pre integrované služby a to nielen na SSL VPN , ale aj dynamické smerovanie, WAN pripojenia, firewally, ale aj Qos. Nevýhodou tejto platformy je fixne nastavenie portov u módu Thin-Client a samozrejme vysoká cena zariadení.

8. ZÁVER

Hlavným cieľom tejto práce bolo nahrnúť, zrealizovať a otestovať rôzne druhy SSL virtuálnych privátnych sieti na platformách Open source, Mikrotik, Cisco s implementáciou PKI a porovnávať tieto riešenia podľa rôznych parametrov. Tento typ VPN oproti klasickému IPsec VPN nemusí mať na koncovej stanici nainštalovanú žiadnu špeciálnu aplikáciu, keďže je SSL prítomne v bežných internetových prehliadačoch, takže sa môže užívateľ na táto bezpečnú bránu jednoducho pripojiť a pristupovať k súborom a webovým serverom vo svojej domácej, firemnej, alebo inej sieti.

Táto technológia je čim viac žiadanejšia a študenti, alebo iný záujemci o túto problematiku majú v tejto práci rozsiahly popis ako SSL VPN funguje. Majú taktiež k dispozícii široký prehľad o vybraných riešeniach, ich názorne topológie, podrobne konfigurácie a samozrejme porovnanie týchto riešení. Táto práca môže pomôcť nielen ako sa oboznámiť s týmto typom VPN, ale má slúžiť ako detailný návod, alebo manuál konfigurácie SSL VPN na spomínaných platformách. Tieto jednotlivé topológie a konfigurácie môžu záujemcovi slúžiť ako skelet pri realizácii svojej predstavy o budovaní jeho vlastných VPN tunelov, alebo sieti.

Ako jediným nedostatkom tejto diplomovej práce považujem neúspešné nakonfigurovanie SSL VPN full tunnel módu na platforme CISCO. Pre úplnú funkčnosť full tunelu módu, je potrebné na smerovač nainštalovať SVC (SSL VPN Client). Problém nastal pri importovaní tohto AnyConnect klienta na smerovač, ktorý by bol po pripojení užívateľa na bránu automaticky stiahnutý na koncovú stanicu. Spočiatku som sa domnieval, že je táto chyba spôsobená konfiguráciou vo virtuálnom prostredí GNS3. Avšak aj po viacnásobnej snahe importovať balíček rôznych verzii priamo na fyzický smerovač kopírovaním či už pomocou SDM, alebo z flash pamäte skončili neúspechom, a preto nebolo možne v konfigurácii pokračovať.

Otázka počítačových sieti a zvlášť bezpečnosti je pre mňa veľmi zaujímavá téma, preto som si vybral túto tému diplomovej práce, aby som sa o tejto problematike dozvedel čím najviac z teoretického a hlavne z praktického hľadiska. Hľadanie a hlavne študovanie potrebných zdrojov pre realizáciu tejto diplomovej práce ma obohatilo o technické veci, kedy som detailne pochopil funkciu VPN sieti. Hodne veľkým prínosom bolo pre mňa riešenie problémov, hľadanie chýb a nedostatkov pri konfigurácii týchto tunelov a sieti. Dúfam, že aj ostatným čitateľom bude táto práca nápomocná a v praxi prospešná.

ZOZNAM POUŽITEJ LITERATÚRY

[1] DataCom, komplexní řešení datových komunikací [online]. [cit.2012-03-21] Dostupné z WWW:<http://www.datacom.cz/ITSystems-IPsec-versus-SSLVPN>

[2] FRAHIM Jazib, HUANG Qiang. SSL Remote Access VPNs. Indianapolis: Cisco Press, 2008. ISBM 1-58705-242-3.

[3] DEAL Richard. The Complete Cisco VPN Configuration Guide. Indianapolis: Cisco Press, 2006. ISBN 1-58705-204-0

[4] Thomas M. Thomas. Zabezpečení počítačových síti. CP Books, 2006. ISBN 80-251-0417-6

[5] ArticSoft, ArticSoft PGP compatible encryption software for comporate data security [online]. [cit.2012-01-12] Dostupné z WWW:<http://www.articsoft.com/public_key_infrastructure.htm>

[6] STU BA, FEI, Podpora distančného vzdelávania v predmete SPA, tímový projekt [online]. [cit.2012-03-02] Dostupné z WWW:<http://labss2.fiit.stuba.sk/TeamProject/2004/team14/aktual/timak-v0.3.pdf

[7] OpenVPN Technologies, Inc.OpenVPN - Open Source VPN[online]. 2010 [cit.2010-11-25]. Dostupné z WWW: <http://openvpn.net/>.

[8] SCHNEIER, Bruce.Bruce Schneier[online]. 15. 2. 2005 [cit. 2010-11-25].SHA-1 Broken. Dostupné z WWW:<http://www.schneier.com/blog/archives/2005/02/sha1_broken.html>.

[9] Oficiálne stránky MikroTik [online]. [cit.2012-04-25] Dostupné z WWW:<http://wiki.mikrotik.com/wiki/Manual:Interface/SSTP>

[10] SSL vpn, podpora vyuky, ročniková práca [online]. [cit.2012-04-20] Dostupné z WWW:<http://www.cs.vsb.cz/grygarek/TPS/projekty/0708Z/SSLVPN-Gebauer-Dergel.pdf>

[11] Oficiálne stránky Cisco [online]. [cit.2012-04-16] Dostupné z WWW:<http://www.cisco.com/en/US/docs/security/asa/asa72/configuration/guide/webvpn.html>

[12] Oficiálne stránky Cisco [online]. [cit.2012-04-20] Dostupné z WWW:<http://www.cisco.com/en/US/products/ps6496/products_configuration_example09186a008072aa61.shtml>

[13] Oficiálne stránky Cisco [online]. [cit.2012-04-20] Dostupné z WWW:<http://www.cisco.com/en/US/products/ps6496/products_configuration_example09186a0080720346.shtml>

[14] SSL Vzorový projekt bezpečnej počítačovej siete [online]. [cit.2012-04-19] Dostupné z WWW:<http://diplom.utc.sk/wan/3580.pdf>

[15] Správa digitálnych certifikátov v prostredí MS Windows [online]. [cit.2013-04-19] Dostupné z WWW:< http://is.muni.cz/th/172593/fi_b/bc-xkubina.pdf>

[16] IPsec podpora vyuky [online]. [cit.2013-04-19] Dostupné z WWW:<http://ucitel.spsbv.cz/kotlarik/indexsoubory/POS/IPSec.pdf>

ZOZNAM OBRÁZKOV

Zoznam príloh

PRÍLOHY

Príloha č. I Obsah súboru default-ssl

```
<VirtualHost _default_:443>

        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        LogLevel warn
        ErrorLog /var/log/apache2/error.log
        CustomLog /var/log/apache2/ssl_access.log combined

        SSLEngine on
        SSLCertificateFile    /etc/apache2/ssl/server.cer
        SSLCertificateKeyFile /etc/apache2/ssl/server.key

        BrowserMatch ".*MSIE.*"
                nokeepalive ssl-unclean-shutdown
                downgrade-1.0 force-response-1.0

</VirtualHost>
```

Príloha č. II Obsah súboru openssl.cnf

```
[ CA_default ]

dir           = ./diplomCA           # Kořenový adresář CA
certs         = $dir/certs           # Adresář obsahující vydané certifikáty
crl_dir       = $dir/crl             # Adresář obsahující CRL
database      = $dir/index.txt       # Index databáze

new_certs_dir   = $dir/newcerts        # Adresář pro nové certifikáty

certificate     = $dir/certs/cacert.pem       # Certifikát CA
serial          = $dir/serial          # Soubor se sérií certifikátů (počítá)

crl           = $dir/crl/crl.pem         # Aktuální CRL
private_key     = $dir/private/cakey.pem# Soukromý klíč CA
RANDFILE      = $dir/private/.rand    # Soubor pro generování náhodných čísel

policy        = policy_anything       # Politiku certifikátů zvolíme volnou
```

Príloha č. III Nastavanie relativnzch ciest v súbore openssl.cnf

```
[ CA_default ]

dir           = /etc/ssl/diplomCA              # Where everything is kept
certs         = /etc/ssl/diplomCA/certs        # Where the issued certs are kept
crl_dir       = $dir/crl             # Where the issued crl are kept
database      = /etc/ssl/diplomCA/index.txt    # database index file.
#unique_subject = no                  # Set to 'no' to allow creation of
                                     # several ctificates with same subject.
new_certs_dir   = /etc/ssl/diplomCA/newcerts             # default place for new certs.

certificate     = /etc/ssl/diplomCA/certs/cacert.pem      # The CA certificate
#certificate    = $dir/cacert.pem       # The CA certificate
serial        = /etc/ssl/diplomCA/serial               # The current serial number
crlnumber      = $dir/crlnumber        # the current crl number
                                     # must be commented out to leave a V1 CRL
crl           = $dir/crl.pem          # The current CRL
private_key     = /etc/ssl/diplomCA/private/cakey.pem# The private key
RANDFILE       = $dir/private/.rand    # private random number file

x509_extensions = usr_cert             # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt       = ca_default           # Subject Name options
cert_opt       = ca_default           # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions       = crl_ext

default_days    = 365                  # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = sha1                 # which md to use.
preserve       = no                   # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy         = policy_anything
```

Príloha č. IV Úspešné naviazanie OpenVPN tunelu

```
Inspiron-N4010:/etc/openvpn# openvpn --config ./vpn_client.conf
Tue Apr 24 22:14:21 2012 OpenVPN 2.2.0 x86_64-linux-gnu [SSL] [LZO2] [EPOLL] [PKCS11]
[eurephia] [MH] [PF_INET6] [IPv6 payload 20110424-2 (2.2RC2)] built on Jul  4 2011
Tue Apr 24 22:14:21 2012 IMPORTANT: OpenVPN's default port number is now 1194, based on an
official port number assignment by IANA.  OpenVPN 2.0-beta16 and earlier used 5000 as the default port.
Tue Apr 24 22:14:21 2012 WARNING: No server certificate verification method has been enabled.  See
http://openvpn.net/howto.html#mitm for more info.
Tue Apr 24 22:14:21 2012 NOTE: OpenVPN 2.1 requires '--script-security 2' or higher to call user-defined
scripts or executables
Tue Apr 24 22:14:21 2012 LZO compression initialized
Tue Apr 24 22:14:21 2012 Control Channel MTU parms [ L:1574 D:138 EF:38 EB:0 ET:0 EL:0 ]
Tue Apr 24 22:14:21 2012 Socket Buffers: R=[126976->131072] S=[126976->131072]
Tue Apr 24 22:14:21 2012 Data Channel MTU parms [ L:1574 D:1450 EF:42 EB:135 ET:32 EL:0 AF:3/1 ]
Tue Apr 24 22:14:21 2012 Local Options hash (VER=V4): 'd79ca330'
Tue Apr 24 22:14:21 2012 Expected Remote Options hash (VER=V4): 'f7df56b8'
Tue Apr 24 22:14:21 2012 UDPv4 link local (bound): [undef]
Tue Apr 24 22:14:21 2012 UDPv4 link remote: [AF_INET]192.168.10.1:1194
Tue Apr 24 22:14:21 2012 TLS: Initial packet from [AF_INET]192.168.10.1:1194, sid=0ae8601d
ba3b1b52
Tue Apr 24 22:14:21 2012 VERIFY OK: depth=1, /C=CZ/ST=Some-
State/O=Internet_Widgits_Pty_Ltd/CN=vsb
Tue Apr 24 22:14:21 2012 VERIFY OK: depth=0, /C=CZ/ST=Some-
State/O=Internet_Widgits_Pty_Ltd/CN=vsb
Tue Apr 24 22:14:21 2012 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Tue Apr 24 22:14:21 2012 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC
authentication
Tue Apr 24 22:14:21 2012 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Tue Apr 24 22:14:21 2012 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC
authentication
Tue Apr 24 22:14:21 2012 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024
bit RSA
Tue Apr 24 22:14:21 2012 [vsb] Peer Connection Initiated with [AF_INET]192.168.10.1:1194
Tue Apr 24 22:14:23 2012 SENT CONTROL [vsb]: 'PUSH_REQUEST' (status=1)
Tue Apr 24 22:14:23 2012 PUSH: Received control message: 'PUSH_REPLY,ping 1,ping-restart
220,ifconfig 10.10.10.100 255.255.255.0'
Tue Apr 24 22:14:23 2012 OPTIONS IMPORT: timers and/or timeouts modified
Tue Apr 24 22:14:23 2012 OPTIONS IMPORT: --ifconfig/up options modified
Tue Apr 24 22:14:23 2012 TUN/TAP device tap0 opened
Tue Apr 24 22:14:23 2012 TUN/TAP TX queue length set to 100
Tue Apr 24 22:14:23 2012 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Tue Apr 24 22:14:23 2012 /sbin/ifconfig tap0 10.10.10.100 netmask 255.255.255.0 mtu 1500 broadcast
10.10.10.255
Tue Apr 24 22:14:23 2012 Initialization Sequence Completed
```