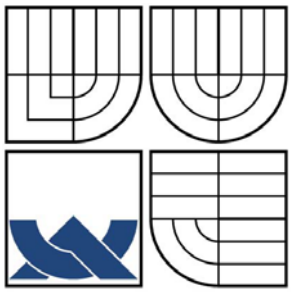
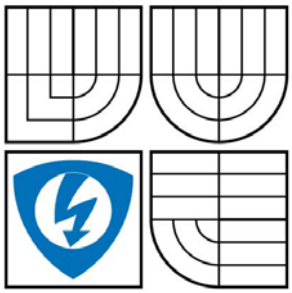


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

HARDWAROVÉ KRYPTOGRAFICKÉ MODULY PRO ZABEZPEČENÍ LAN

HARDWARE CRYPTOGRAPHIC MODULES FOR LAN SECURITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE **Bc. TOMÁŠ LOUTOCKÝ**
AUTHOR

VEDOUCÍ PRÁCE **Doc. Ing. VÁCLAV ZEMAN, Ph.D**
SUPERVISOR

BRNO 2008

Oficiální zadání

Licenční smlouva 1

Licenční smlouva 2

ANOTACE

Práce se zabývá problematikou virtuálních privátních sítí (VPN). První část práce je zaměřena na vysvětlení základních pojmů počítačové bezpečnosti, které jsou nutné pro lepší pochopení dalších částí. Ve druhé části je rozebrána technologie VPN a její rozdělení dle určitých aspektů. Celá následující část práce je věnována realizaci VPN pomocí protokolu IPSec a jeho podrobnému popisu. V praktické části práce je uvedeno vlastní řešení zabezpečení laboratorní sítě produkty firmy Safenet. Následně jsou uvedeny modulární postupy popisující práci s jednotlivými produkty používanými v síti. V postupech jsou také popsány některé bezpečnostní slabiny, které je možné v laboratorní síti využít, a také způsoby ochrany proti jejich zneužití.

Klíčová slova: VPN, IPSec, SafeNet, bezpečnost, certifikát

ABSTRACT

The thesis deal with the problems of virtual private network (VPN). The first part of the thesis is focused on the description of the basic terms of computer security which are useful for better understanding the other parts. There is a description of VPN technology and its separation of VPN by various aspects in the second part of the thesis. The next chapter is dedicated to the description of realization of VPN by using IPSec. There is shown how to secure laboratory network by using of the products of the Safenet Company in the practical part of the thesis. There are also stated the modular techniques how to use products in the network in practical part. Some of the modular techniques describe security weaknesses of the products which are possible to exploit in the laboratory network and they also describe the ways how to protect them against misusage.

Keywords: VPN, IPSec, SafeNet, security, certificate

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Hardwarové kryptografické moduly pro zabezpečení LAN jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.
Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne …… ……..
(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Doc. Ing. Václavu Zemanovi, Ph.D za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne …… ……..
(podpis autora)

Seznam použitých zkratek a symbolů

AES	Advanced Encryption Standard
AH	Authentication Header
AMC	Administrative Management Center
CA	Certification Authority
CLI	Command line interface
CRL	Certificate revocation list
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DOS	Denial of Service
DSA	Digital Signature Algorithm
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module
IPSEC	Internet Protocol security
ISAKMP	Internet Security Association and Key Management Protocol
MD5	Message-Digest algorithm 5
MITM	Man In The Middle
PED	Pin Entry Device
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PSK	Pre-Shared Key
SA	Security Association
SHA	Secure Hash Algorithm
SMC	Security Management Center
SSL	Secure Sockets Layer
SSO	Single Sign-On
USB	Universal Serial Bus
VPN	Virtual Private Network

Obsah

Úvod

Cílem této práce je popsat problematiku zabezpečení počítačových sítí se zaměřením na virtuální privátní sítě s využitím hardwarových kryptografických modulů. V práci jsou popsány základní termíny počítačové bezpečnosti, se kterými se dnes běžně setkáváme. Další část této práce se zabývá technologií virtuálních privátních sítí (VPN – Virtual Private Network). Zde jsou popsány jednotlivé topologie VPN, typy této technologie dle umístění v referenčním modelu OSI a rozdíly softwarového a hardwarového řešení. Podrobně je rozebrán protokol IPSec. Nejprve jsou charakterizovány režimy tohoto bezpečnostního protokolu, následně je vysvětlena funkce jeho hlavních dílčích protokolů. Na závěr této kapitoly je protokol IPSec zhodnocen. Jsou zde popsány podporované algoritmy, výhody a nevýhody této technologie. V praktické části práce je popsáno vlastní řešení zabezpečení laboratorní sítě pomocí certifikovaných produktů firmy Safenet. V poslední části jsou uvedeny modulární postupy, ve kterých si studenti vyzkouší základní konfigurací a použití prvků sítě. Uvedena je také řada postupů zaměřená na bezpečnost, kdy se studenti seznámí s bezpečnostními slabinami a možnými obranami proti nim.

1. Základní pojmy počítačové bezpečnosti

V oblasti počítačové bezpečnosti se setkáváme s řadou pojmů, zejména z oblasti kryptografie. V podkapitolách jsou vysvětleny základní pojmy, které se používají a jsou zmiňovány v následujících kapitolách této práce.

1.1 Kontrolní součet

Kontrolní součet nebo také hash či otisk (viz lit. [4], [18]) je jednocestná funkce, která z libovolně dlouhého textu vytvoří krátký řetězec konstantní délky. Jednocestná znamená, že vytvoření otisku by mělo být výpočetně jednoduché, ale získání původní zprávy z otisku velice náročné. Funkce vytvoření kontrolního součtu by také měla být bezkolizní, což znamená, že není výpočetně možné najít dva různé texty se stejným otiskem. Mezi nejznámější algoritmy patří SHA-1 a MD5. Algoritmus MD5 byl vytvořen v roce 1991, ale byly v něm nalezeny bezpečnostní chyby, proto se od použití MD5 v bezpečnostních aplikacích upouští. V roce 2005 byl objeven algoritmus, který umožňuje nalézt kolizi v SHA-1, proto se raději dnes doporučuje používat hashovacích funkce SHA-256, SHA-384, SHA-512 a SHA-224. U těchto algoritmů totiž zatím nebyly nalezeny žádné bezpečnostní slabiny.

1.2 Symetrické šifry

Symetrické šifrování (viz lit. [4], [18]) využívá při šifrování stejný klič jako při dešifrování. Nevýhodou tohoto typu šifrování je nezbytnost bezpečné výměny onoho klíče mezi oběma stranami před začátkem komunikace, protože kdyby útočník odchytil tento klíč, mohl by si celou komunikaci dešifrovat. Naopak velkou výhodou je rychlost šifrovacího algoritmu. Rozšířeným algoritmem je DES (Data Encryption Standard) vyvinutý v 70. letech v USA americkou vládou. Používá šifrovací klíč délky 56 bitů, ale tato délka symetrického klíče se dnes považuje za nedostatečnou. Proto byly vyvinuty

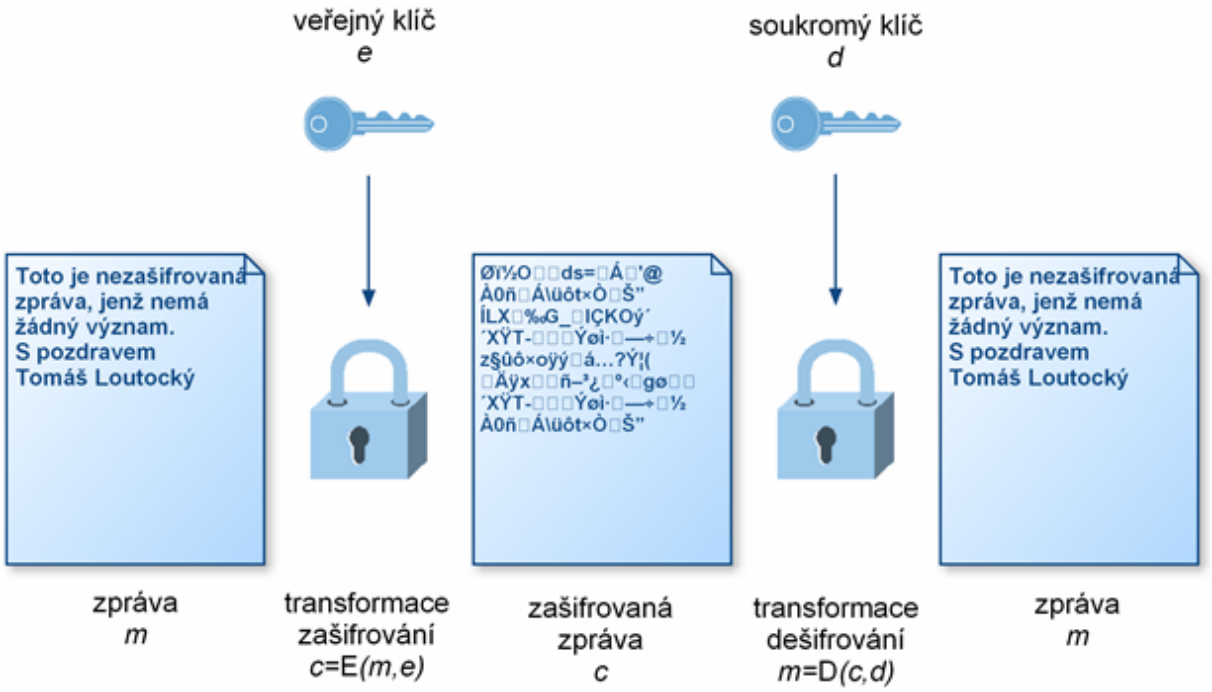
algoritmy s větší délkou šifrovacího klíče, např. od výše zmíněného algoritmu DES byl odvozen algoritmus 3 DES s délkou klíče 112 bitů, dále např. AES či IDEA. K popularitě algoritmu IDEA přispělo zejména jeho použití ve volně dostupném šifrovacím balíku PGP.

Obr. 1: Princip symetrických šifer

1.3 Asymetrické šifry

Tento typ šifer (viz lit. [4], [18]) na rozdíl od symetrických nepoužívá jeden utajený šifrovací klíč, ale používá klíče dva - jeden soukromý a druhý veřejný. Vlastností asymetrických šifer je, že je jednoduché zašifrovat zprávu veřejným klíčem, ale velice složité získat onu původní zprávu jen na základě znalosti veřejného klíče. Soukromý klíč si jeho majitel drží v bezpečí, kdežto veřejný klíč bývá distribuován veřejnosti. Nevýhodou asymetrických šifer je fakt, že aplikace asymetrických algoritmů je výrazně pomalejší než užití algoritmů symetrických. To je dáno matematickou podstatou asymetrických algoritmů a délkou používaných klíčů. Proto se například při šifrování komunikace používá symetrická šifra a asymetrická jen pro počáteční výměnu klíče pro

onu symetrickou šifru. Nejznámějším asymetrickým šifrovacím algoritmem je RSA. Délka klíče asymetrických šifer algoritmu RSA se v dnešní době považuje za bezpečnou, pokud je dlouhá minimálně 1024 bitů. Často se můžeme setkat i s delším klíčem. Například certifikační autority, velké korporace, používají délku klíče 2 nebo i 4 kbity. Mezi další asymetrické šifry patří ECC, DSA, Diffie-Hellmanův, Elgamal atd.

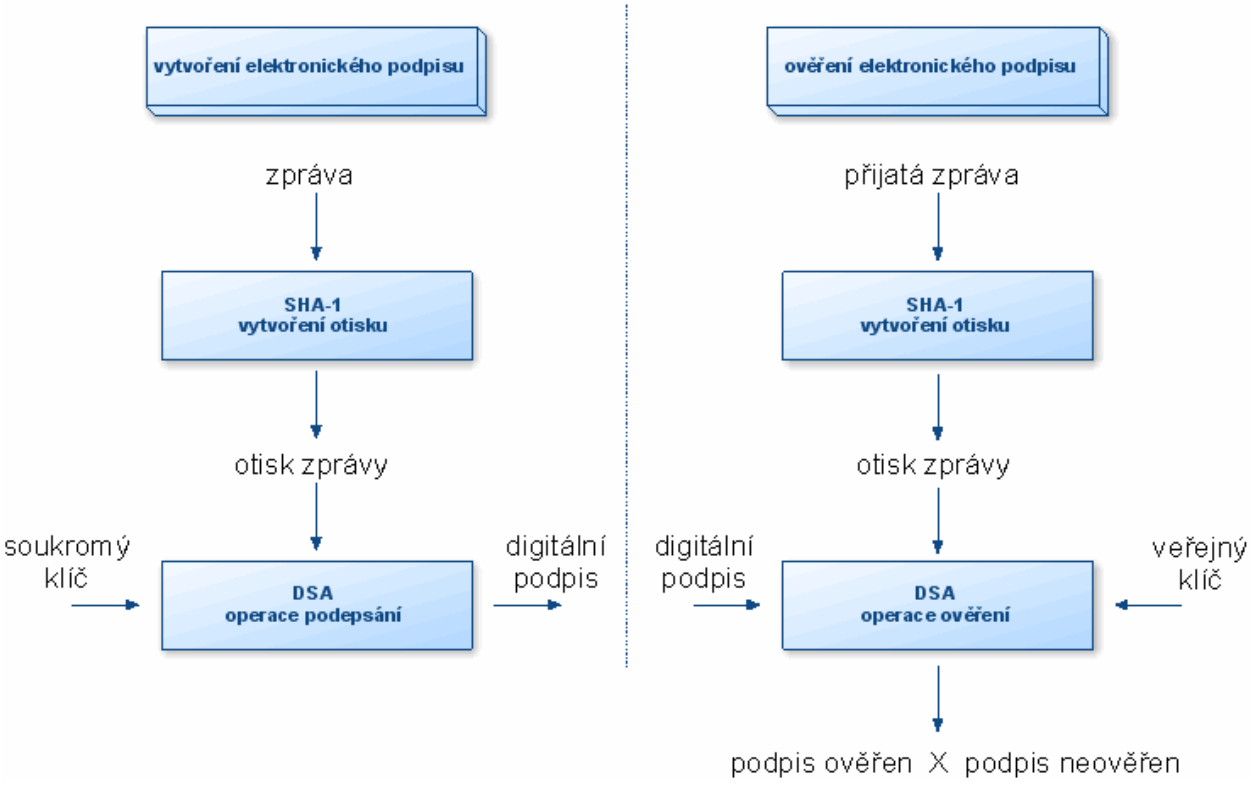


Obr. 2: Princip asymetrických šifer

1.4 Elektronický podpis

Elektronický podpis (viz lit. [18], [21]) stejně jako podpis rukou zajišťuje důkaz o pravosti dat. Elektronické podepsání dokumentu se skládá ze dvou kroků. Prvním krokem je vypočítání kontrolního součtu z dokumentu. Dalším krokem je podepsání tohoto kontrolního součtu soukromým klíčem uživatele. Elektronický podpis je tedy otisk zprávy podepsaný soukromým klíčem. Soukromým klíčem se podepisuje pouze otisk, protože podepsání celé zprávy, často spolu s obsáhlou přílohou, by mohlo trvat velmi dlouho.

Proto je výhodnější podepisovat pouze příslušný otisk zprávy. Navíc zpráva s elektronickým podpisem bude čitelná i v případě, kdy příjemce nemá příslušné nástroje pro ověření její pravosti. Ověření pravosti elektronického podpisu spočívá ve třech krocích. V prvním kroku si příjemce spočte kontrolní součet přijaté zprávy. Ve druhém kroku pomocí veřejného klíče odesílatele ověří elektronický podpis odesílatele a následně v posledním kroku porovná vypočítaný otisk z prvního kroku a ověřený ze druhého kroku. Pokud jsou stejné, může příjemce věřit, že danou zprávu napsal odesílatel. Jen on mohl podepsat kontrolní součet zprávy, neboť vlastní soukromý klíč. Na obr. 3 je zobrazena konkrétní varianta digitálního podpisu dle normy FIPS 186-2 využívající hashovací algoritmus SHA-1 a podepisovací algoritmus DSA.



Obr. 3: Princip elektronického podpisu dle FIPS 186-2

1.5 Certifikát

Certifikát (viz lit. [4], [18]) je datová struktura, jejímž nejdůležitějším úkolem je ověření identity vlastníků veřejného klíče. Tímto zabraňuje podvržení veřejného klíče. Tato vlastnost certifikátů se nejčastěji používá pro elektronický podpis, ale také například pro bezpečnou komunikaci pomocí protokolu SSL. Nejznámějším formátem certifikátů a také zde popisovaným je certifikát podle normy X.509 (dle RFC 3280). Dnes se nejběžněji používají certifikáty verze 3 normy X.509. Data obsažená v certifikátu jsou popsána pomocí jazyku ASN.1. Tento jazyk se používá pro popis obecných datových struktur. Dále se kóduje pomocí kódováni DER (Distinguished Encoding Rules). Případně pro snadnější práci jsou následně převedena do kódovaní BASE64 (formát PEM).

1.5.1 Ukázka certifikátu

```
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 1 (0x1)
      Signature Algorithm: md5WithRSAEncryption
      Issuer: C=CZ, ST=Czech rep, L=Olomouc, O=Testovaci CA, CN=Testovaci
CA/emailAddress=autorita@abiko.cz
      Validity
         Not Before: Mar 24 18:50:28 2007 GMT
         Not After : Mar 24 18:50:28 2008 GMT
      Subject: C=CZ, ST=Czech rep, L=Brno, O=AbikoSoft,
CN=localhost/emailAddress=abiko@abikosoft.cz
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:c4:d6:23:ef:b1:dc:fb:42:2d:8f:07:7e:b5:ec:
               9c:84:8c:16:c9:9e:50:5f:c8:1f:8e:49:70:16:0c:
               94:1b:38:b9:e6:58:0b:77:06:c4:bc:04:76:92:ae:
               fe:b7:a5:5a:a3:8e:ba:8a:86:73:26:c0:fd:69:b9:
               2b:dd:14:9c:b9:cc:f4:f3:4b:c1:86:14:4f:13:e9:
               a8:f1:c0:0c:28:df:66:fa:ae:0f:c7:ac:13:4d:ec:
               ad:ec:93:57:25:20:69:61:30:fc:3f:d1:16:ff:03:
               37:fe:66:73:95:aa:cc:49:b5:c3:18:35:27:f4:c6:
               f7:0d:4d:f6:aa:ef:35:ac:77
            Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
```

cc:2d:27:b5:7c:0a:b5:7a:e4:67:3b:27:29:db:a8:c2:c5:81:
f6:19:80:29:7b:e1:32:b0:df:38:d4:9e:47:1c:02:22:c3:52:
2e:35:a2:04:cb:43:c2:f9:51:ac:05:e4:a8:69:be:c0:59:e7:
9e:3a:7f:a9:93:76:bb:0c:ed:f4:a2:d5:c0:66:b0:d4:61:a5:
60:b7:07:13:1e:f5:bb:41:73:6f:66:7d:40:2f:c0:cc:51:76:
c4:41:96:19:af:66:1c:af:6e:71:65:7c:0f:da:79:b0:fe:49:
39:8c:3d:80:33:e9:46:a8:79:5c:af:75:fb:5b:9d:de:78:ae:
21:c5
-----BEGIN CERTIFICATE-----
MIICczCCAdygAwIBAgIBATANBgkqhkiG9w0BAQQFADCBgzELMAkGA1UEBhM
CQ1ox
EjAQBgNVBAgTCUN6ZWNoIHJlcDEQMA4GA1UEBxMHT2xvbW91YzEVMBMGA
1UEChMM
VGVzdG92YWNpIENBMRUwEwYDVQQDEwxUZXN0b3ZhY2kgQ0ExIDAeBgkqhki
G9w0B
CQEWEWF1dG9yaXRhQGFiaWtvLmN6MB4XDTA2MDMyNDE4NTAyOFoXDTA3
MDMyNDE4
NTAyOFowezELMAkGA1UEBhMCQ1oxEjAQBgNVBAgTCUN6ZWNoIHJlcDENMA
sGA1UE
BxMEQnJubzESMBAGA1UEChMJQWJpa29Tb2Z0MRIwEAYDVQQDEwlsb2NhbGhv
c3Qx
ITAfBgkqhkiG9w0BCQEWEmFiaWtvQGFiaWtvc29mdC5jejCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEAxNYj77Hc+0Itjwd+teychIwWyZ5QX8gfjklwFgyUGzi5
5lgLdwbEvAR2kq7+t6Vao466ioZzJsD9abkr3RScucz080vBhhRPE+mo8cAMKN9m
+q4Px6wTTeyt7JNXJSBpYTD8P9EW/wM3/mZzlarMSbXDGDUn9Mb3DU32qu81rHc
C
AwEAATANBgkqhkiG9w0BAQQFAAOBgQDMLSe1fAq1euRnOycp26jCxYH2GYAp
e+Ey
sN841J5HHAIiw1IuNaIEy0PC+VGsBeSoab7AWeeeOn+pk3a7DO30otXAZrDUYaVg
twcTHvW7QXNvZn1AL8DMUXbEQZYZr2Ycr25xZXwP2nmw/kk5jD2AM+lGqHlcr3
X7
W53eeK4hxQ==
-----END CERTIFICATE-----

1.5.2 Certifikační autorita

Hlavním úkolem certifikační autority (CA) je vydávání certifikátů. CA vydává certifikáty na základě žádosti o certifikát. Vydané certifikáty CA podepíše svým soukromým klíčem. CA klade velmi velké nároky na ochranu svého soukromého klíče. Pokud by byl soukromý klíč CA kompromitován, byla by to pro danou CA katastrofa. Soukromý klíč je většinou uložen ve speciálních zařízeních, které nejsou pro vyšší bezpečnost připojeny k internetu. Takováto specializovaná zařízení se nazývají High

Security Modules (HSM). Kromě svého soukromého klíče si CA musí chránit databázi svých klientů, archiv soukromých klíčů uživatelů. Certifikát CA, který je podepsán sám sebou, neboli má stejný obsah pole vydavatel jako pole předmět, se nazývá kořenový. Certifikáty tvoří stromovou strukturu, přičemž nejníže je certifikát uživatele a nejvýše kořenový certifikát. Je možný i případ, kdy si dvě CA podepíší své certifikáty, pak se jedná o křížovou strukturu.

CA se při vydávání certifikátu řídí určitými pravidly, které jsou shrnuty v certifikační politice. Certifikační politika je dokument certifikační autority, jenž vysvětluje pravidla pro vydávání certifikátů. Například informuje o struktuře a použití vydávaných certifikátů, poskytovaných službách, zásadách nakládání s certifikáty atd.

1.5.3 Odvolání certifikátu

K odvolání certifikátu dochází při kompromitování soukromého klíče, který je do páru s veřejným klíčem obsaženým v daném certifikátu. V tomto případě uživatel, jehož soukromý klíč byl kompromitován, podává žádost na odvolání certifikátu. Tuto žádost je důležité podat co nejdříve po zjištění kompromitace, aby nedošlo ke zneužití uživatelova soukromého klíče. Žádost můžeme podat elektronickou zprávou, kterou podepíšeme soukromým klíčem, příslušejícím k danému certifikátu. Pokud ale daný certifikát není určen pro elektronické podepisování, je nutné volit jiný způsob. Certifikační autority pro takové případy vydávají jednorázová hesla, pomocí kterého můžeme certifikát odvolat telefonem, faxem či přes webový formulář. Nejpomalejší cestou je osobní kontakt u certifikační autority, která si následně pro odvolání certifikátu ověří osobní údaje uživatele.

1.5.4 Seznam revokovaných certifikátů

Pokud dojde k odvolání certifikátu dříve, než vyprší jeho platnost, příslušná certifikační autorita daný certifikát umístí na seznam revokovaných certifikátů, neboli CRL (Certificate Revocation List). V CRL jsou zveřejňována sériová čísla odvolaných certifikátu do té doby, než vyprší jejich řádná platnost. Certifikační autority vystavují své CRL na svých stránkách, mohou ale také určit i jiná distribuční místa pro odvolané

certifikáty. CRL jsou vydávány v pravidelných intervalech, což může být i nevýhoda, jelikož k revokaci certifikátu dojde až za onen interval, kdy vyjde novy CRL. Tento problém je možné vyřešit protokolem Online Certificate Status List (OCSP). Jedná se o protokol klient/server, kdy klient zasílá serveru dotaz obsahující identifikaci příslušného certifikátu. Server mu vrátí odpověď zda je onen certifikát revokován. OCSP server může provozovat certifikační autorita nebo server, kterému certifikační autorita přidělila příslušná práva.

1.6 Úložiště certifikátů

Certifikát je datová struktura, jenž je svázána s příslušným párem klíčů. Vlastník certifikátu potřebuje uchovat nejen certifikát, ale také vazbu na příslušné klíče. Tuto funkci řeší úložiště certifikátů (viz lit. [1], [18]). Úložiště certifikátů mohou odkazovat na různá umístění schránek klíčů. Úložiště se tedy většinou nachází na straně klienta. Existují logická a fyzická úložiště. Fyzickým úložištěm je myšleno konkrétní úložiště a naproti tomu logické úložiště tvoří odkazy na fyzické úložiště. Fyzickým úložištěm může být HDD počítače či externí zařízení, jako například USB token či čipová karta.

Úložiště certifikátů může zprostředkovávat řada specializovaných zařízení. V následujících podkapitolách jsou uvedeny běžné a uživateli často využívané úložiště.

1.6.1 Protected Storage System

Operační systém Windows obsahuje službu nazvanou Protected Storage System, která slouží pro ukládání citlivých dat, jako jsou například digitální certifikáty. Tato citlivá data jsou ukládána do registrů Windows, konkrétně do:

HKEY_CURRENT_USER\Software\Microsoft\Protected Storage System\Provider

Data zde uložená jsou šifrována symetrickou šifrou, která se odvozuje z uživatelova přihlašovacího jména a hesla. Spravovat a procházet úložiště certifikátu ve Windows

umožňuje modul snap-in Certifikáty. Tento modul se spouští: Start → Spustit → Certmgr.msc

1.6.2 Internetové prohlížeče

Prohlížeč Opera obsahuje správce certifikátů, pomocí kterého můžeme spravovat certifikáty v tomto prohlížeči uložené. Certifikáty certifikačních autorit jsou uloženy v souboru opcacrt6.dat. Osobní certifikáty a privátní klíče jsou uloženy v souboru opcert6.dat.

Mozilla i Firefox stejně jako Opera mají úložiště certifikátů, které ukládá certifikáty do následujících souborů. Do souboru cert8.db jsou ukládány veřejné klíče a certifikáty certifikačních autorit. Tento soubor není šifrován. Šifrován je naopak soubor key3.db, který obsahuje soukromé klíče. Ten je chráněn tzv. hlavním heslem.

Internet Explorer využívá úložiště operačního systému Windows.

1.6.3 Čipové karty a USB tokeny

Z bezpečnostního hlediska je vhodné ukládat své soukromé klíče mimo pevný disk počítače, na bezpečné místo, kam se případný útočník těžko dostane. Nejlépe na zařízení, které může mít uživatel neustále u sebe a nad kterým tak bude mít plnou kontrolu. Tyto požadavky splňují čipové karty a USB tokeny. Výhodou čipové karty je její velikost, nevýhodou zase potřeba čtečky čipových karet. Token je bezpečnější, pokud z něj nelze klíč vyexportovat. Klíč lze na tokenu vytvořit, při operacích neputuje do systému a je stále používán jen na tokenu. Pro zjištění zda a jak je daný token bezpečný, je vhodné zjistit jeho certifikaci. Levnější tokeny certifikovány nejsou. Nejčastěji se u tokenů můžeme setkat s certifikací FIPS 140-2 úroveň 2. Podrobněji o standardech FIPS pojednává kapitola 1.7.

Základní možnosti autentizace:

- Znalost – založeno na znalosti určitého hesla (u tokenů je to typicky PIN)
- Vlastnictví – možnost autentizace má pouze vlastník specifického předmětu (USB token, čip. karta)
- Biometrika – tento typ autentizace je založen na tzv. biometrických vlastnostech uživatele, jako jsou například otisky prstů, geometrie ruky, oční sítnice, tvar obličeje nebo rozpoznávání řeči. Příkladem může být USB token či čtečka čipových karet s možností otisku prstu.

Vícefaktorová autentizace je kombinací uvedených metod.

1.7 Bezpečnostní normy

Bezpečnostní norma nám ručí za to, že dané zařízení prošlo bezpečnostními testy a poskytuje garantovanou úroveň bezpečnosti. Mezi nejznámější bezpečnostní normy patří normy Federal Information Processing Standards (FIPS) (viz lit. [5], [6]). Jsou to v podstatě směrnice, které stanovují nejvhodnější praktiky pro softwarové a hardwarové bezpečnostní produkty. FIPS vydává nevládní organizace National Institute of Science and Technology (NIST). Certifikace FIPS je rozšířená po celém světě a je důležitá pro mnoho organizací. Například americké vládní organizace mohou nakupovat pouze produkty s touto certifikací, také řada finančních korporací vyžaduje FIPS certifikaci. Produkty mající FIPS mohou většinou pracovat ve dvou režimech. Jedním je non-FIPS režim a druhým je FIPS režim, kdy zařízení není schopno používat jakékoliv neschválené FIPS metody.

Nejznámější FIPS standardy:

- 186-2 Standard digitálního podpisu RSA and DSA
- 180-1 Secure Hash Standard SHA-1
- 180-2 Aktualizace Secure Hash Standard SHA-1, plus navíc SHA-256, SHA-384, SHA-512

– 140-2 Standard pro bezpečnostní požadavky kryptografických modulů. Rozdělen na čtyři úrovně:

  - Úroveň 1: Nejnižší úroveň. Ukládá velmi málo požadavků.
  - Úroveň 2: Přidává požadavky na fyzické zabezpečení a na autentizaci.
  - Úroveň 3: Dále zvyšuje požadavky na fyzické zabezpečení (např. ztížení a detekce pokusu o fyzický přístup do modulu) a autentizaci.
  - Úroveň 4: Nejvyšší úroveň tohoto standardu, jenž dále zvyšuje fyzickou bezpečnost. Vhodné do prostředí, které není fyzicky chráněno.

2. VPN

VPN neboli Virtual Private Network (viz lit. [8], [17], [19]) lze popsat jako zašifrovaný tunel mezi dvěma počítači přes nezabezpečenou síť. Neboli VPN umožňuje vytvoření bezpečné komunikace přes veřejnou síť, jakou je například veřejný internet. VPN využívá šifrování a autentizaci pro zajištění toho, aby přenášená informace zůstala privátní a důvěrná. To znamená, že můžeme sdílet data a zdroje mezi více místy, bez obavy o ztrátu integrity dat. Schopnost využívat veřejnou síť je velkou výhodou VPN, protože jinak by pro zajištění stejné funkce bylo nutné si například pronajmout linku bod-bod (např. E1), která je finančně nákladná, obzvláště když body komunikace jsou od sebe geograficky vzdáleny. VPN je velmi efektivní pro uživatele na cestách. Tehdy se uživateli stačí připojit k lokálnímu poskytovateli internetu a následně přes VPN se přímo připojit k privátní firemní síti. Veškerá komunikace s privátní sítí pak bude probíhat zabezpečeně.

Důležité rozhodnutí při instalaci VPN je, jaké zařízení bude poskytovat koncový bod VPN tunelu (hraniční router, firewall či specializované zařízení). Je potřeba si také uvědomit, jak velký přibližně VPN provoz bude a zda nebude potřeba šifrovací akcelerátor. Nedostatek výpočetní síly se totiž může projevit značným zpomalením připojení.

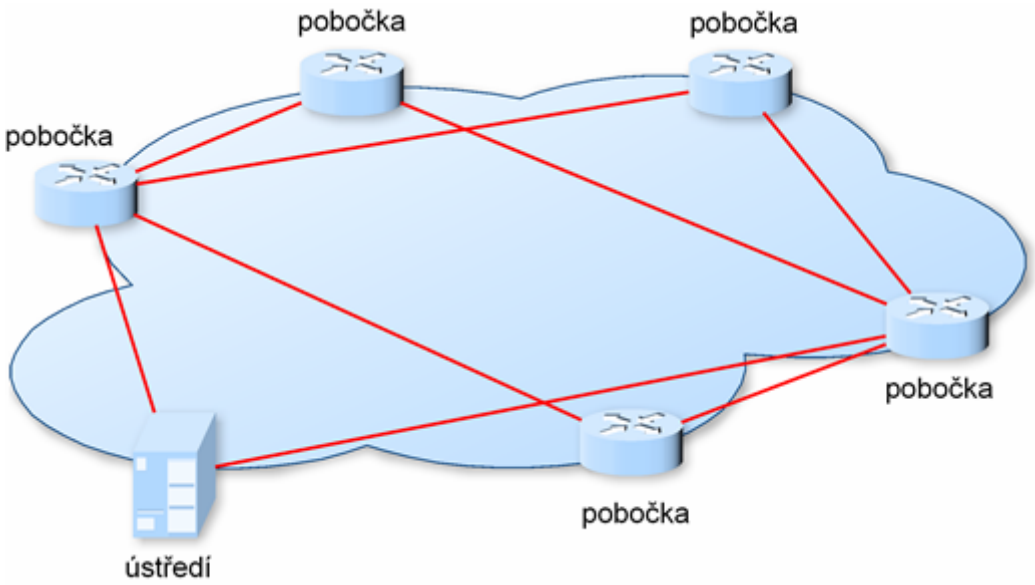
2.1 Topologie

VPN může mít několik rozdílných topologií (viz lit. [8]). Zde jsou čtyři obecné topologie, které budou dále blíže rozebrány :

- Meshed (plně či částečně)
- Hvězda
- Hub and Spoke

2.1.1 Meshed

Tato topologie může být implementována v plné či částečné konfiguraci. Plná meshed topologie má mnoho alternativních cest k cíli. Celkově je v síti s n uzly (n*(n – 1))/2 cest. Tato konfigurace tak poskytuje určitou redundaci, protože každé VPN zařízení je spojeno se všemi ostatními VPN zařízeními. Jednodušší kompromis představuje částečná meshed topologie, kdy nejsou všechny uzly přímo navzájem propojeny. Tuto topologii zobrazuje obr. 4.

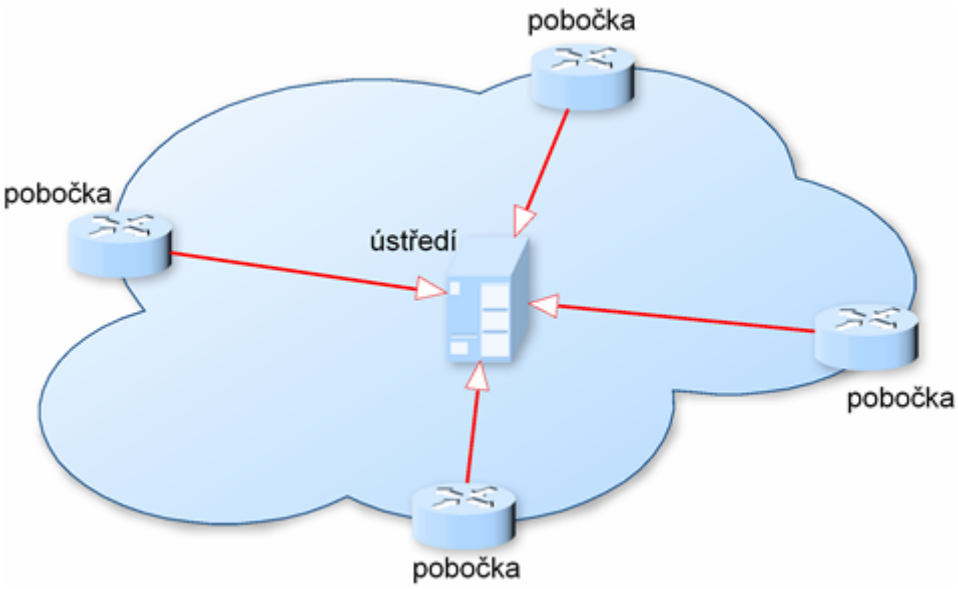


Obr. 4: Příklad částečné Meshed topologie

Meshed topologie poskytuje podstatnou výhodu v tom, že celkový výkon není závislý na jednom konkrétním uzlu. Pokud se například stane v jednom uzlu porucha, ostatní uzly stále mohou mezi sebou komunikovat. Další výhodou je, že uzly, které jsou geograficky blíže sobě, mohou spolu komunikovat bez nutnosti komunikovat přes centrální uzel. Hlavní nevýhodou této topologie je její správa a údržba. Pro plně meshed variantu platí, že pokud přidáme nový uzel, všechny ostatní uzly musí být aktualizovány. Další nevýhodou je cena, jelikož musíme pořídit VPN zařízení pro každý uzel.

2.1.2 Hvězda

V tomto typu topologie mohou vzdálené počítače či sítě (např. pobočky firmy) zabezpečeně komunikovat s centrálním serverem (např. ústředím firmy). Komunikace mezi jednotlivými pobočkami navzájem není povolena (viz obr. 5).

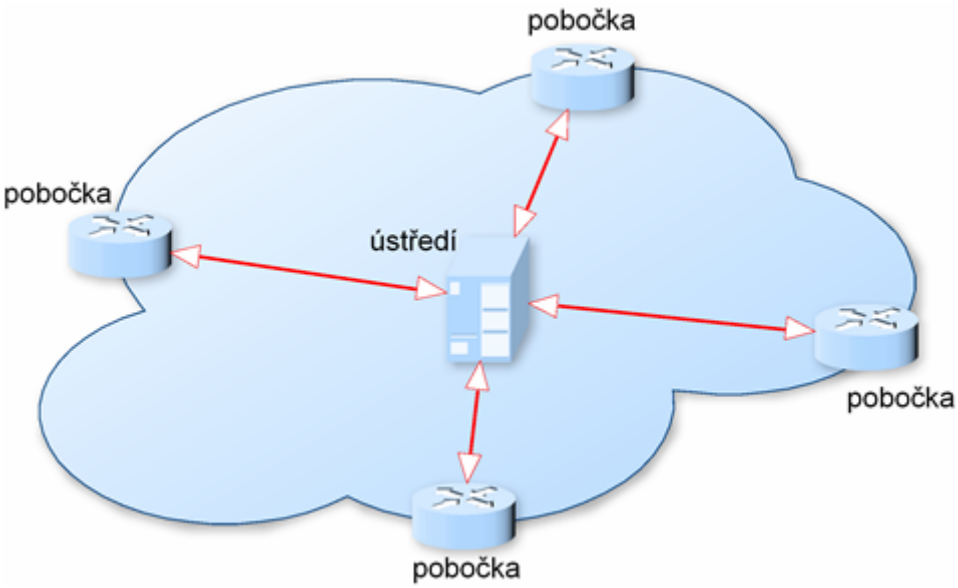


Obr. 5: Příklad topologie hvězda

Tento typ topologie je vhodný například pro banky, kdy kompromitace jedné z poboček nevede ke kompromitaci dalších poboček. Přidání nového uzlu je jednoduché, není potřeba konfigurovat ostatní uzly v síti, což je značná výhoda oproti předchozí topologii. Centrální uzel, se kterým komunikují ostatní uzly, má důležitou roli. Jeho výkon ovlivňuje výkon celé VPN sítě. Pokud centrální uzel selže, všechna spojení v síti selžou také.

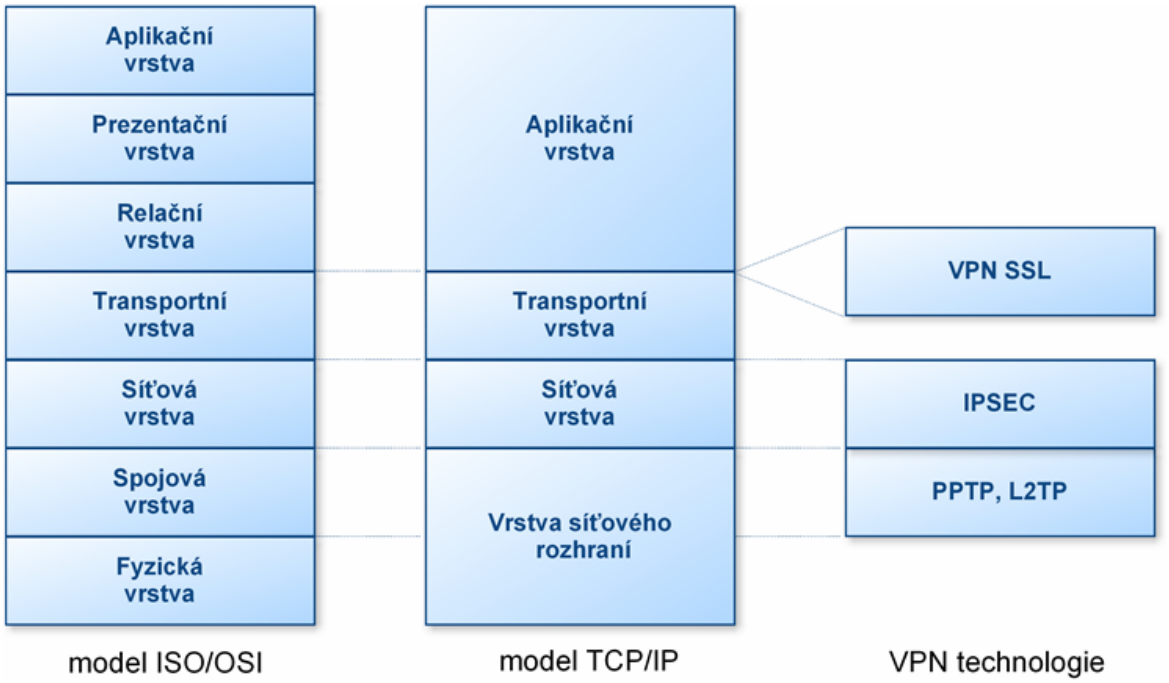
2.1.3 Hub and Spoke

Tato topologie je velmi podobná předchozí topologii. Je zde však několik podstatných rozdílů. Oproti topologii hvězda všechny okolní uzly mají přístup k ostatním uzlům (např. pobočkám). Centrální uzel pracuje jako jednoduchý tranzitní bod pro veškerou komunikaci z jednoho konce sítě na druhý. V centrálním uzlu se data dešifrují, prozkoumají a znovu zašifrují pro odeslaní do konečného cíle. V této topologie se naskýtá větší bezpečností riziko oproti topologii hvězda. Pokud útočník kompromituje jednu pobočku, může zaútočit na další bez nutnosti ovládnutí centrálního uzlu sítě. Nevýhodou oproti meshed topologii je fakt, že dva uzly musí komunikovat přes centrální uzel i pokud se nachází geograficky blízko.



Obr. 6: Příklad topologie Hub and Spoke

2.2 Typy VPN dle síťového modelu



Obr. 7: Technologie VPN v síťovém modelu

2.2.1 PPTP

Point-to-Point Tunneling Protocol (PPTP) (viz lit. [25]) je protokol tunelového propojení operující na druhé vrstvě referenčního modelu OSI. Byl vytvořen společností Cisco a dále rozvíjen zejména firmou Microsoft, v jejíchž produktech je nativně podporován. PPTP je rozšíření protokolu Point-to-Point Protokol (PPP), které zdokonaluje mechanismy komprese, ověřování a šifrování protokolu PPP. Šifrování zajišťuje Microsoft Point-to-Point Encryption (MPPE), který využívá algoritmů RSA a RC4. MPPE používá pro autentizaci uživatele a šifrování 40, 56 a 128 bitové klíče. Klíče jsou odvozeny z uživatelského hesla, což představuje bezpečnostní riziko. Šifrování dat je tak bezpečné, jak bezpečné je uživatelské heslo. Protokol IPSec (viz kapitola 3) je mnohem bezpečnější z pohledu šifrování dat. Klíče jsou totiž generovány náhodně a jsou nezávislé na autentizačních informacích.

2.2.2 L2TP

Protokol Layer Two Tunneling Protokol (L2TP) (viz lit. [16]) je také jako PPTP protokol tunelového připojení. Je založen na standardu RFC. Oproti protokolu PPTP nevyužívá tento tunelovací protokol k šifrování dat standard MPPE, ale využívá službu protokolu IPsec. Protokol L2TP nachází své využití především při připojování vzdálených klientů k firemní síti, kdy L2TP většinou vzdálenému klientovi přidělí IP adresu odpovídající vnitřní síti. Obecně lze říci, že VPN realizované na druhé vrstvě (PPTP,L2TP) je vhodné pro uživatele, kteří chtějí mít dohled nad vnitřním směrováním, překladem adres a správou bezpečnosti. Má se za to, že uživatel využívající VPN na třetí vrstvě přenechá dozor nad VPN provozovateli. Nová verze tohoto protokolu nese název L2TPv3.

2.2.3 VPN SSL

VPN pomocí protokolu SSL je poměrně nová záležitost. Záměr je vytvořit co nejtransparentnější šifrovaný tunel za pomoci protokolu SSL (viz lit. [8], [10]). Velkou výhodou je přítomnost podpory protokolu SSL v každém prohlížeči. Není tudíž nutné instalovat na klientské počítače specifický klientský program. Uživatel se tak může připojit například i z internetové kavárny. K rozšíření schopností VPN řešení slouží Java aplety či prvky ActiveX, které jsou také součástí webových prohlížečů. Pomocí nich můžeme přes SSL protunelovat konkrétní port. Základní funkcí tohoto řešení je umožnit zabezpečený přístup k vnitřním informačním zdrojům firmy. Vytváří se tak šifrovaný tunel mezi SSL VPN bránou a webovým prohlížečem, jak můžeme vidět na obr. 8. SSL VPN také umožňuje využití protokolu Common Internet File System (CIFS), který umožňuje práci se soubory ve vnitřní síti. Nejedná se o náhradu IPSecu, SSL VPN například není vhodná pro propojení jednotlivých sítí. Dle odborných studií by VPN SSL měla převažovat při přístupu jednotlivých uživatelů.

Obr. 8: Princip VPN SSL

2.2.4 IPSec

Dalším typem VPN je protokol IPSec, který funguje na síťové vrstvě modelu OSI. Tento protokol je podrobněji rozebrán v kapitole 3.

2.3 Softwarové vs. Hardwarové řešení VPN

Obecně nelze říci, které z těchto řešení je lepší (viz lit. [8], [24]). Každá síť má své vlastní specifické požadavky. V následujícím textu budou popsány výhody, nevýhody a vlastnosti obou řešení.

Nízkou cenou vyniká softwarové řešení, obzvláště když základní software je již obsažen v operačním systému. Například stanice s OS Windows mají nezbytný software pro použití VPN Windows serveru.

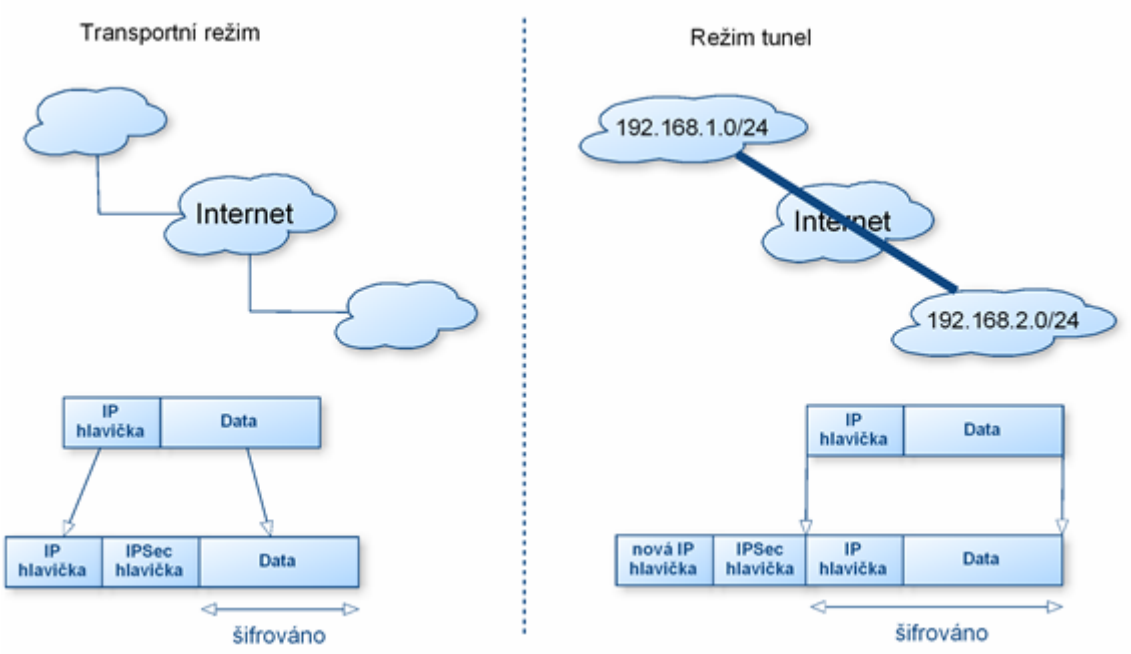
Avšak existuje řada silných argumentů proti používaní softwarového (SW) řešení VPN. Bezpečnost je hlavním z nich. Na serveru může běžet, a většinou také běží, spousta aplikací. Pokud se operační systém či kterákoliv aplikace stane zranitelnou, tak je zranitelný celý systém. Z toho vyplývá, že pokud kompromitujeme systém, můžeme tak kompromitovat i VPN službu a opačně. Nebezpečný SW, jakým jsou například viry a tzv. červi, mnohem více ohrožují SW řešení.

Autentizace a šifrování ve VPN může také hrát významnou roli v zatížení serveru určeného pro všeobecné účely. SW varianta je spíše vhodnější pro situaci, kdy VPN občas využívá několik uživatelů. V jiných případech je vhodnější zvolit pro realizaci VPN

specializované HW zařízení. HW řešení také vyniká nad SW v rychlosti a umožňuje větší počet současně běžících VPN spojení.

3. IPSec

IPSec je skupina protokolů zajišťujících zabezpečení komunikace na úrovni vrstvy IP (viz lit. [4], [11], [14]). Je to ve své podstatě rozšíření protokolů IP, které není závislé na protokolech vyšších vrstev. Není nutné, aby IPSec pracoval na úrovni operačního systému. Je možné, aby pracoval až na úrovni hraničního směrovače a vlastní operační systém jej tak ani nemusí podporovat. Specifikace protokolu jsou uvedeny v RFC-2401 až RFC-2412. Protokol umožňuje dva režimy: transportní režim a režim tunel (viz obr. 9) IPSec je směs více protokolů, které budou v dalším textu popsány.



Obr. 9: Režimy IPSec

Transportní režim je výchozím a jednodušším případem pro protokol IPSec. Při použití tohoto režimu se šifrují pouze data protokolu IP. Mezi záhlaví protokolu IP a záhlaví protokolů vyšších vrstev je vloženo bezpečnostní záhlaví, které specifikuje, jakým způsobem je datová část IP datagramu zabezpečena.

Režim tunel na rozdíl od předchozího režimu šifruje záhlaví i data protokolu IP. V tomto režimu je celý IP paket zapouzdřen, dále je přidána bezpečnostní hlavička

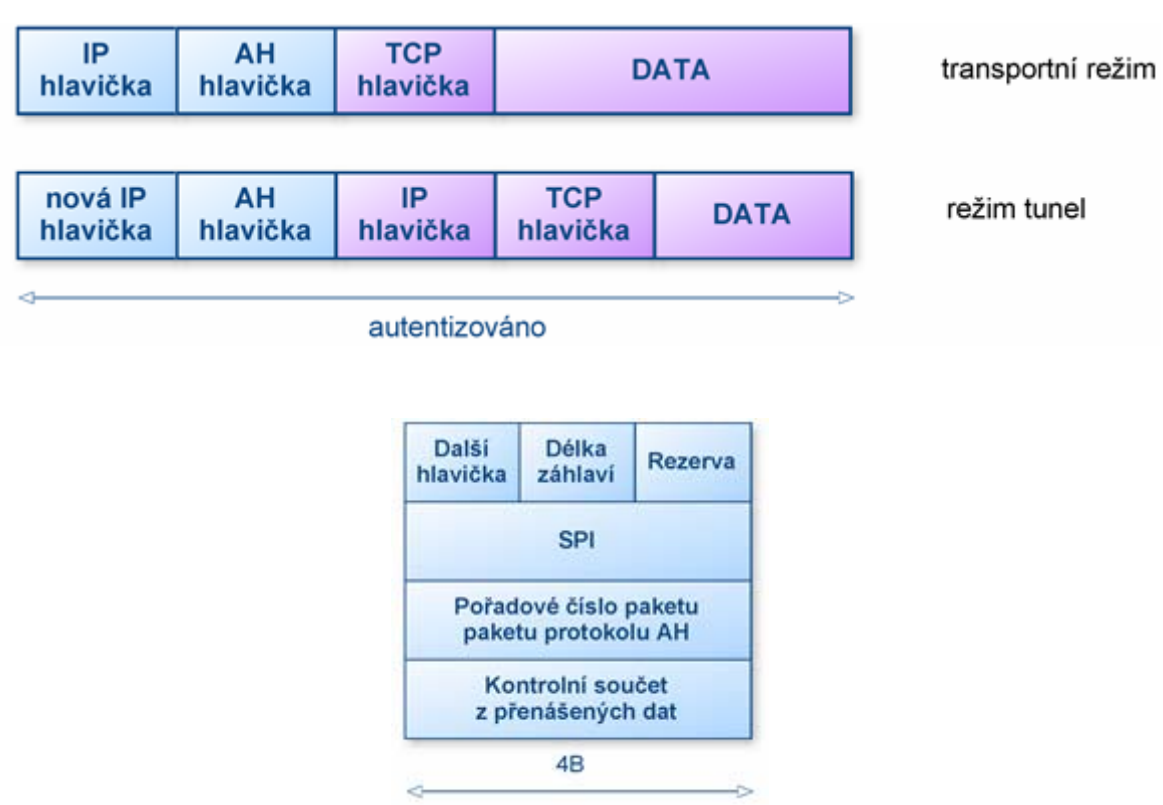
(protokolu AH nebo ESP) a také doplňková hlavička protokolu IP. Celkem jsou tedy zde dvě IP hlavičky - vnější a vnitřní (viz obr. 9).

Základní způsoby IPSec komunikace jsou:

- Mezi dvěma počítači - prakticky nepoužívané.
- Mezi dvěma směrovači - například spojení LAN – LAN.
- Mezi počítačem a směrovačem – například když firemní zaměstnanec přistupuje do sítě organizace z domova.

3.1 Protokol AH

Protokol Authentication Header (AH) zprostředkovává integritu, ověření a ochranu před zneužitím celého paketu (viz lit. [12]). Také AH protokol brání proti útoku zopakováním paketu. Není zde zajištěno utajení, jelikož přenášená data nejsou šifrována. Přenášená data je tedy možné přečíst, ale je znemožněno je pozměnit.



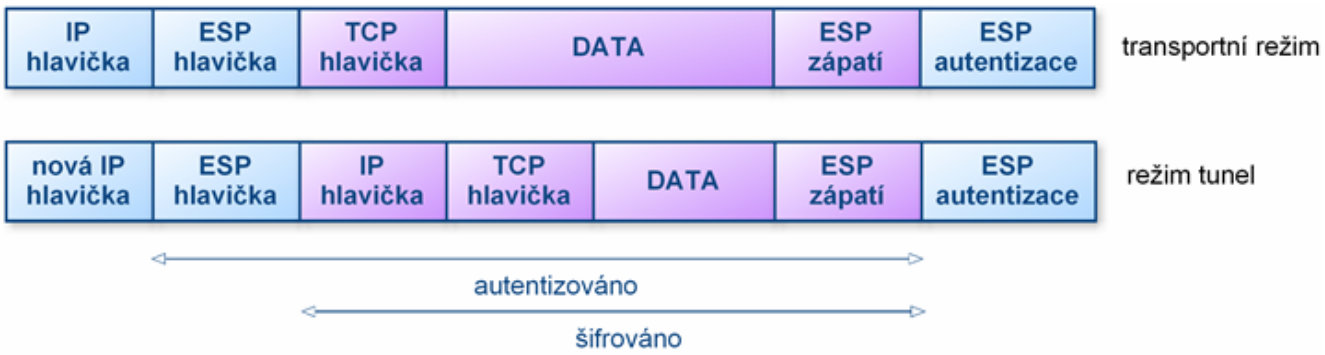
Obr. 10: Protokol AH

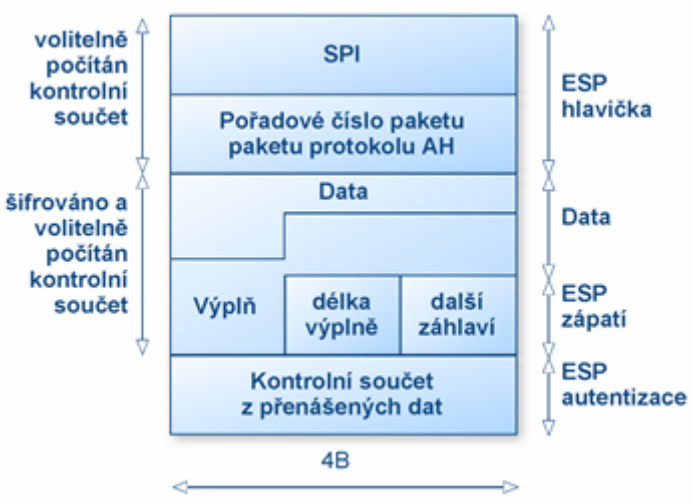
Význam jednotlivých polí protokolu AH:

- Další hlavička – specifikace typu zabezpečených dat.
- Délka záhlaví – délka hlavičky protokolu AH.
- SPI – Security Parameters Index (SPI) je užíván v kombinaci s protokolem AH či ESP a také s cílovou adresou pro identifikaci zabezpečení komunikace. Jedná se v podstatě o ukazatel do databáze, jenž uvádí hodnoty šifrovacích klíčů a sdílených tajemství pro daný spoj.
- Pořadové číslo protokolu AH - poskytuje obranu vůči útoku spočívajícím v opakování paketu. V případě přijetí již dříve přijatého paketu, je paket zahozen.
- Kontrolní součet – toto pole obsahuje kontrolní součet sloužící k ověření zprávy a kontrole její integrity.

3.2 Protokol ESP

Protokol Encapsulating Security Payload (ESP) poskytuje ověření zdroje dat, integritu a také důvěrnost dat (viz lit. [13]). ESP podporuje i konfiguraci bez autentizace, ta ale z důvodu bezpečnosti není doporučena. Oproti protokolu AH není IP hlavička zabezpečena.





Obr. 11: Protokol ESP

Význam jednotlivých polí protokolu ESP:

- SPI – Security Parameters Index (SPI) - význam stejný jako u protokolu AH.
- Pořadové číslo protokolu ESP - význam stejný jako u protokolu AH.
- Data – přenášená a šifrovaná data.
- Výplň – pole zajišťuje, aby se šifrovaná data nalézala v rozsahu bajtů, které požadují šifrovací algoritmy.
- Délka výplně – obsahem je délka pole výplně v bajtech. Pole slouží pro příjemce k odebrání bajtů výplně po dešifrování dat.
- Další hlavička – pole identifikuje typ dat, jenž obsahuje datová část (například TCP či UDP).
- Kontrolní součet – volitelné, využívá stejného algoritmu jako u AH. Hlavička IP protokolu do kontrolního součtu není zahrnuta.

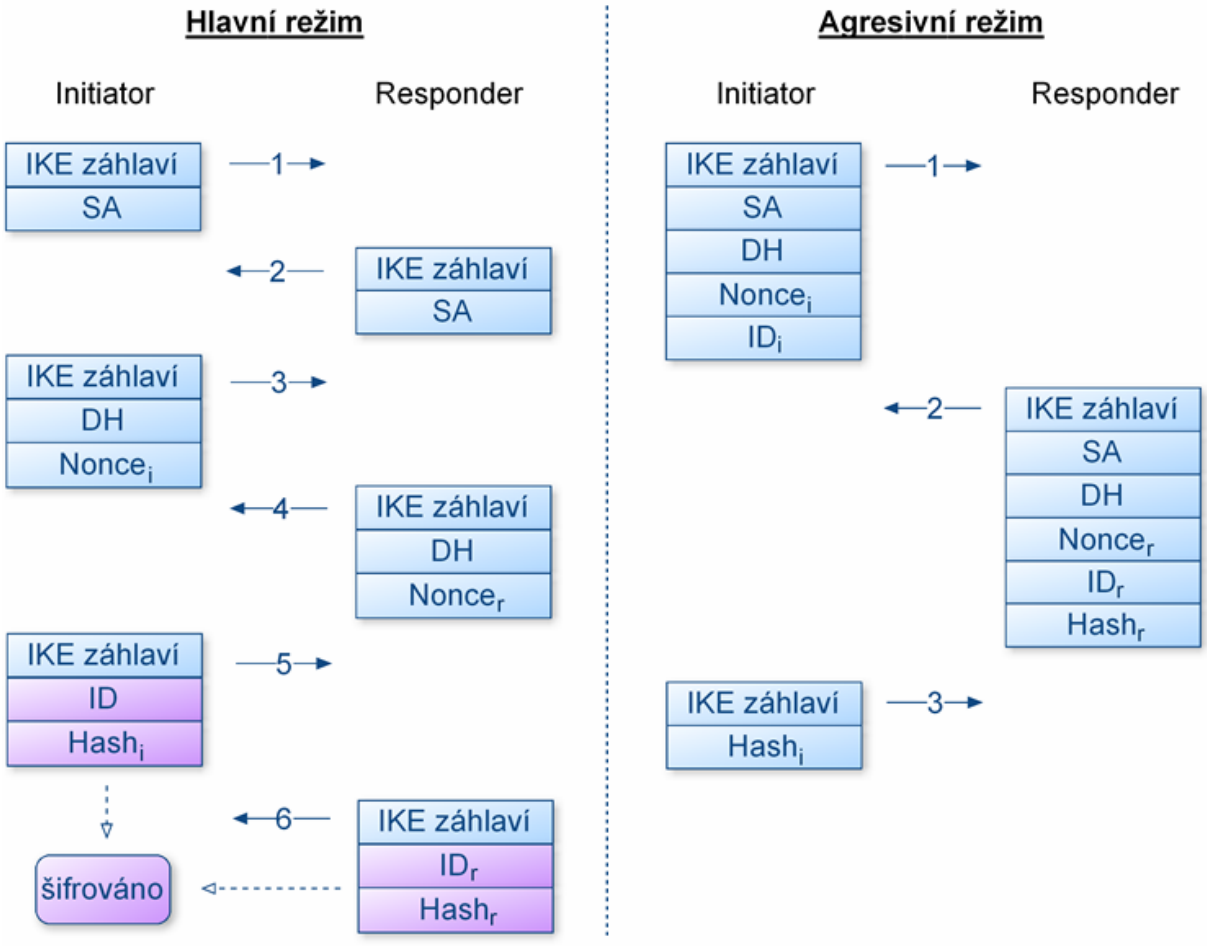
3.3 Protokol IKE a ISAKMP

Před tím, než IPSec vyšle autentizovaná nebo šifrovaná data, vysílací i přijímací strana komunikace se musí dohodnout na určitých pravidlech komunikace (protokolu, šifrovacím algoritmu a klíčích). Pro dohodnutí těchto pravidel se používá protokol Internet

Key Exchange (IKE) (viz lit. [7]) a protokol Internet Security Association and Key Management Protocol (ISAKMP) (viz lit. [20]). Protokol ISAKMP poskytuje strukturu pro protokol IKE, popisující konkrétní výměnu dat mezi oběma konci komunikace. ISAKMP definuje procedury a formáty paketů pro ustanovení a dohodnutí Security Association (SA). Pomocí zpráv SA si strany dohodnou metody, jako například šifrovací či hashovací algoritmy pro sjednání následné zabezpečené komunikace. Protokol IKE je vnořen do protokolu ISAKMP a přenáší se v něm vlastní kryptografická data (klíče, certifikáty, hashe) jako součást vybudování VPN tunelu. IKE využívá algoritmus Diffie-Hellman pro sestavení sdíleného tajemství, ze kterého jsou následně odvozeny klíče pro šifrování komunikace. Komunikace protokolem IKE se sestává ze dvojic zpráv a to z žádosti, jenž je následovaná odpovědí. Je zvykem označovat stranu, která zahajuje spojeni Initiator, a stranu, která odpovídá Responder (pozn. nepoužívá se zde označení klient/server). Pro zajištění spolehlivosti je zde definován časový interval, do kterého musí být přijata odpověď. Pokud přijata není, je vyslána nová žádost. IKE komunikace probíhá ve dvou fázích.

V první fázi se uskuteční vzájemná autentizace (například za pomocí předsdíleného klíče či PKI) a zjedná se šifrovací klíč relace, kterým je následně šifrována další IKE komunikace. Existují dva režimy, jak uskutečnit tuto první fázi - hlavní režim a agresivní režim. Na obr. 12 je uvedena autentizace pomocí předsdíleného klíče (dále PSK – pre-shared key). Jak je možné z obrázku poznat, oba režimy se liší počtem vyslaných zpráv. Hlavní režim si potřebuje vyměnit celkem šest zpráv. Agresivní režim rychleji sestavuje spojení, ale to je jeho jediná výhoda. Velkou nevýhodou je nízká úroveň bezpečnosti a to zejména při autentizaci pomocí PSK. Všechny zprávy přenášené v agresivním režimu jsou nešifrované. Zejména u polí identifikace uživatele a také hash je to nebezpečné. Hash se počítá z pole nonce a z vlastního předsdíleného klíče. Pokud útočník odposlechne tuto nezašifrovanou komunikaci, může se následně pokusit z hashe dostat předsdílený klíč a vzhledem tomu, že se často využívá dnes ne již tak bezpečný hashovací algoritmus MD5, má útočník velkou šanci, že se mu to podaří. Další bezpečnostní slabinu agresivního režimu s autentizací pomocí PSK představuje fakt, že Responder odešle nešifrovaně hash spočítaný ze správného PSK i za situace, kdy Initiator se pokusí autentizovat pomocí nesprávného PSK. Této slabiny může potenciální útočník lehce využít. V kapitole 5.9 bude popsán a prakticky ukázán útok na tento režim. Díky

těmto slabinám se doporučuje raději používat autentizaci pomocí Public Key Infrastructure (PKI). Autentizace pomocí PKI se liší od předchozího typu autentizace tím, že za polem ID se nachází pole certifikát. A také se liší tím, že hash je podepsán privátním klíčem majitele onoho uvedeného certifikátu. Kromě mnohem vyššího stupně zabezpečení (použití asymetrické šifry) má PKI autentizace výhodu ve snadné distribuci autentizačního materiálu (v tomto případě certifikátů). Problémem autentizace pomocí PSK je bezpečné sdělení tohoto klíče druhé straně.



Obr. 12: Hlavní a agresivní režim protokolu IKE s autentizací PSK

Význam jednotlivých polí při komunikace protokolem IKE:

- SA – pomocí těchto zpráv si obě strany komunikace dohodnou jaký budou používat šifrovací algoritmus, jaký algoritmus pro výpočet hashe či jaký protokol (AH / ESP).
- DH – pole, kde se využívá algoritmus Diffie-Hellman pro sestavení sdíleného tajemství, ze kterého jsou následně odvozeny klíče pro šifrování komunikace.
- Nonce – je to náhodné číslo, ze kterého se také počítá hash. Zabraňuje se opakování stejného hashe. Hash není předvídatelný.
- ID – identifikace komunikující strany (např. IP adresa, DNS jméno, uživatelské jméno atd.)
- Hash – otisk z určitých polí zprávy.

Ve druhé fázi se využívá bezpečný kanál, vytvořený v první fázi pro vyjednání bezpečných kanálů (SA) protokolu AH nebo ESP. Většinou se vyjednají dva SA kanály, jeden pro každý směr komunikace.

3.4 Zhodnocení IPSec

Protokol IPSec poskytuje při správné implementaci velmi vysokou úroveň zabezpečení na síťové vrstvě. IPSec využívá tři druhy algoritmů: pro šifrování, pro ověření pravosti a pro ustanovení klíčů. Šifrovat je možné pomocí šifrovacího algoritmu DES s délkou šifrovacího klíče 56 bitů. Tento algoritmus již ale není příliš bezpečný. Z tohoto důvodu je možné využít algoritmus 3-DES s délkou šifrovacího klíče 112 bitů. Také je možné využit AES, který umožňuje mít šifrovací klíč dlouhý 128, 192 nebo 256 bitů. Pro ověření pravosti zprávy jsou k dispozici modifikované hashovací algoritmy MD5 a SHA-1 nazvané HMAC-MD5 a HMAC-SHA-1. Ty vykonávají funkci hash dvakrát, pokaždé odlišně a kombinují zprávu s klíčem. Pro ustanovení klíče přes nezabezpečený kanál využívá protokol IPSec Diffie-Hellmanův algoritmus. Na výběr jsou tři Diffie-Hellman skupiny, kde skupina DH 5 je nejbezpečnější (klíč 1536 bitů), dále skupina DH 2

(klíč 1024 bitů) a poslední skupina DH 1 (klíč 768 bitů). Čím vyšší skupina, tím větší nároky na výkon.

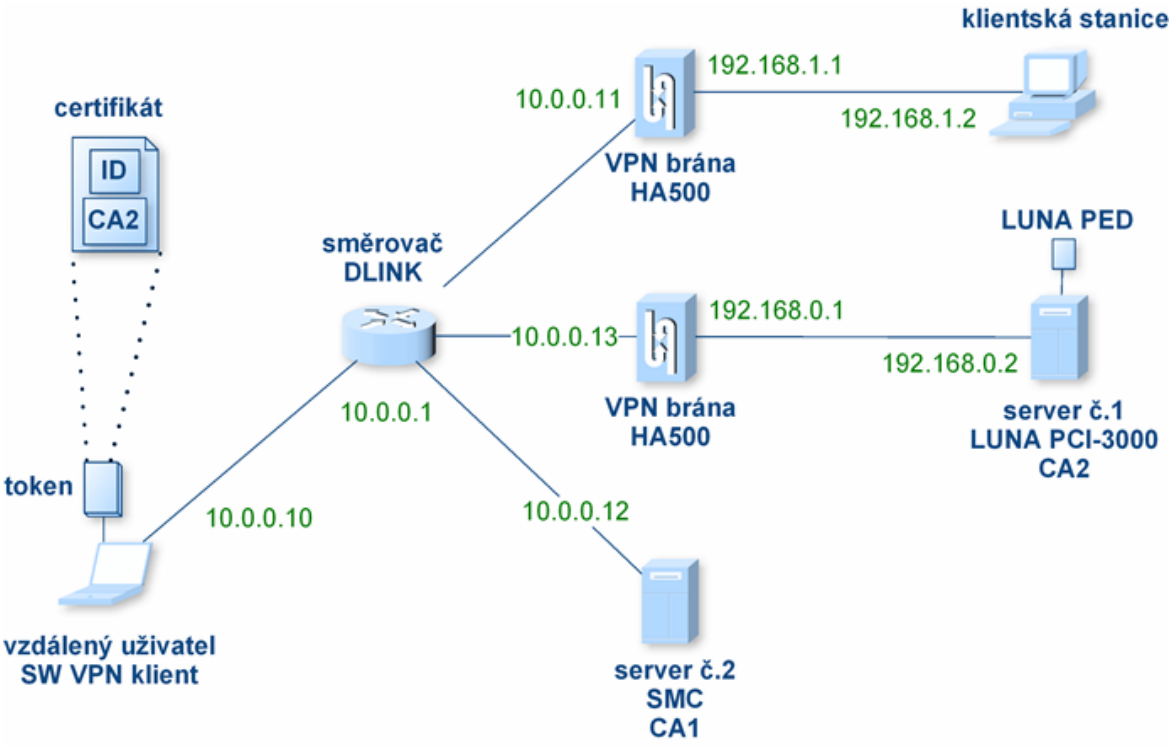
Výhodou IPSec protokolu je, že se jedná o otevřený standard. Není tak svázaný s jedním konkrétním algoritmem či metodou. Další výhodou je jeho implementace v síťové vrstvě. Není tak závislý na konkrétní aplikaci, netřeba konfigurovat jednotlivé aplikace pro bezpečnou komunikaci oproti implementaci v aplikační vrstvě. IPSec také poskytuje ochranu proti Man In The Middle (MITM) útoku, kdy se útočník umístí mezi komunikující body a jednomu z bodů předstírá, že je ten druhý bod a opačně.

To, že je IPSec otevřený standard, může být i nevýhoda. Rozdílné implementace protokolu od různých výrobců mohou způsobit vzájemnou nekompatibilitu. Další nevýhodou je špatná funkce protokolu za Network Address Translation (NAT). Děje se tomu, protože protokol AH spočítá hash z hlavičky IP protokolu. Tato hlavička (zdrojová IP) se ale díky NAT mění. Když paket dorazí na druhou stranu, tak se vypočítá hash, který ale nesouhlasí s uvedeným hashem v paketu. Paket je následně zahozen, jelikož nebyla potvrzena integrita. Problém může být vyřešen použitím režimu tunel s využitím ESP protokolu, kdy se integrita nepočítá z IP hlavičky. Při použití ESP se naskýtá problém s TCP a UDP kontrolním součtem pro ověření paketů. ESP šifruje TCP hlavičky a tak je NAT nemůže změnit (zejména čísla portů). Následně pak nesouhlasí kontrolní součet a paket je zahozen. Tento problém lze také odstranit podporou NAT-T (Network Address Translation Traversal) (viz lit. [15]). NAT-T přidá mezi ESP a IP hlavičku UDP hlavičku. Do ní se uloží porty pro adresaci a také zdrojová IP adresa. To následně dovoluje ověřit kontrolní součet. Často je protokolu IPSec vytýkána přílišná složitost. Avšak i přes všechny nevýhody je IPSec lepší než jakýkoliv jiný bezpečný IP protokol.

4. Návrh konkrétního řešení zabezpečení sítě

V následující kapitole bude ukázáno možné řešení zabezpečení počítačové sítě pomocí produktů firmy Safenet v laboratoři. Budou uvedeny konfigurace jednotlivých zařízení v uvedené architektuře sítě. V jednotlivých podkapitolách budou popsány vlastnosti a také funkce zařízení používaných v laboratoři. Čtenář by tak měl získat představu o vlastnostech hardwarových i softwarových bezpečnostních prvků v síti a o tom, jakou úlohu dané zařízení v síti představuje.

4.1 Schéma zapojení



Obr. 13: Architektura sítě v laboratoři

4.2 Konfigurace jednotlivých zařízení

Název	klientská stanice
IP adresa	192.168.1.2
MAC adresa	00-16-17-CE-1A-32
Použitý HW	CPU Intel Pentium IV. 2.8GHz RAM 504MB
Použitý SW	Microsoft Windows XP Profesional Version 2002 + Service Pack 2 High Assurance Remote VPN klient Bordless Security SW
Pozn.	vzdálený klient jméno: UTKO-1

Název	vzdálený klient
IP adresa	10.0.0.10
MAC adresa	00-16-17-7E-92-20
Použitý HW	CPU Intel Pentium IV. 2.8GHz RAM 504MB
Použitý SW	Microsoft Windows XP Profesional Version 2002 + Service Pack 2 High Assurance Remote VPN klient Bordless Security SW
Pozn.	klient ve vnitřní síti doména: safenet jméno: UTKO-2

Název	server č.1
IP adresa	192.168.0.2
MAC adresa	Private NIC 00-15-17-44-CC-81
Použitý HW	CPU 2x čtyřjádrový Intel Xeon 1.6GHz RAM 2x 1024MB DDR2 800MHz HDD 2x 250GB/7200rpm LUNA PCI-3000+ PED
Použitý SW	Windows Server 2003 Standard Edition + Service Pack 2 SafeNet Borderless Security Administrative Management Center (AMC)
Pozn.	doména: safenet jméno: UTKO-SERVER

Název	server č.2
IP adresa	10.0.0.12
MAC adresa	00-1C-C4-D7-53-7B
Použitý HW	CPU Intel Xeon 3040 - 1,86GHz RAM 4GB HDD 160GB
Použitý SW	Windows Server 2003 Standard Edition + Service Pack 2 Security Management Center (SMC)
Pozn.	server pro správu VPN prvků v síti

4.3 Certifikace zařízení v laboratoři dle normy FIPS

Tab.č.1: Certifikace zařízení v laboratoři dle FIPS

Zařízení	Stupeň normy FIPS 140-2
LUNA PCI-3000	3
LUNA PED (Pin Entry Device)	3
Smart Card Datakey 330	2
Ikey 2032	2
VPN brána HA500	2

4.4 HSM modul

Luna PCI-3000 je zařízení typu Hardware Security Modules (HSM). Modul zajišťuje dvě základní funkce.

První funkcí je akcelerace kryptografických operací. Vhodné použití je kupříkladu na serveru, kde běží zabezpečený webový server, na který se připojují stovky klientů přes protokol SSL. V takovém případě kryptografické operace vykonává HSM modul a procesor serveru tak není zbytečně vytížen.

Podpora PKI je další funkcí. Kritickým bodem bezpečnostního řešení založeného na PKI je kořenový privátní klíč certifikační autority, kterým jsou podepisovány všechny vydané certifikáty dané CA. Pokud by tento klíč získal útočník, čímž by jej kompromitoval, mohl by ho využít k falšování certifikátů (například vydávání svojí osoby za jinou, která dané CA také věří). Pokud by tato situace nastala, důvěryhodnost dané CA by byla zničena, a to pravděpodobně navždy. Aby privátní klíč CA potenciální útočník

nemohl jednoduše nahrát ze zařízení CA, je zde k dispozici právě modul HSM. Ten nabízí bezpečné uložení kryptografických klíčů. Kromě toho, že jsou klíče zabezpečeny pomocí šifrování, jsou zabezpečeny i fyzicky a modul umožňuje detekovat pokusy o fyzické narušení. Pokud takový případ nastane, všechna uložená data jsou zničena.

Pro bezpečné přihlášení k modulu slouží LUNA Pin Entry Device (PED) a sada USB tokenů. Tokeny jsou rozděleny dle práv. Rozeznáváme token inicializační, token Security Officer (SO), token skupinový a token uživatelský. Přihlášení do HSM modulu tedy probíhá vložením příslušného tokenu do zařízení PED a zadáním číselné PIN kombinace daného tokenu.

4.5 Software pro správu síťových prvků v síti

Security Management Center (SMC) je robustní Java aplikace, která umožňuje vzdáleně konfigurovat a monitorovat zařízení, která jsou integrována do architektury sítě SMC. Pomocí této aplikace můžeme do sítě přidávat zařízení jako VPN brány, šifrátory, počítačové stanice či celé sítě. Mezi těmito objekty SMC umožňuje definovat VPN politiky (sada pravidel definující VPN tunel). V laboratoři budou pomocí tohoto programu spravováni VPN klienti a VPN brány HA500. Zařízení, které má být SMC spravováno, musí být certifikováno. Pro tento účel je součástí SMC také vlastní certifikační autorita.

Používání digitálních certifikátů v síťovém prostředí umožňuje vybudovat vzájemnou důvěru mezi komunikujícími stranami. Certifikáty vydané SMC CA se vztahují k výrobním certifikátům (u jednodušších zařízení k sériovým číslům) zařízení a slouží pro ověření autenticity daného zařízení. Každý výrobní certifikát obsahuje unikátní číslo, které je odvozeno z MAC adresy zařízení. Toto unikátní číslo spolu s privátním klíčem a kořenovým certifikátem CA je uloženo ve flash paměti daného zařízení. Výrobní certifikát je vydáván výrobcem daného zařízení a slouží pro získaní oprávnění pro použití v dané síti během instalace. Po instalaci je výrobní certifikát nahrazen certifikátem vydaným CA SMC jako nový základ důvěry. V laboratoři je k dispozici VPN brána typu HA500. Ta se certifikuje na základě jejího sériového čísla, výrobní certifikát k ní není dodáván. Během samotné instalace SMC je vytvořen pár kořenových šifrovacích klíčů a

k nim odpovídající certifikát. Ten je podepsán sám sebou (tzv. self-signed) a slouží jako kořenový certifikát SMC.

4.6 VPN brána

VPN brána HighAssurance 500 (HA500) umožňuje využívat VPN pomocí IPSec tunelů v plně duplexním provozu s rychlostí 1.5 Mbps. Mezi další funkce patří firewall a směrování (podpora směrovacích protokolů RIP a OSPF). HA500 umožňuje až 500 současně probíhajících IPSec tunelů. Správa brány je možná přes službu Telnet, SSH, sériový kabel a SMC. V laboratoři je využívána centralizovaná správa těchto bran pomocí SMC.

4.7 Autentizační předměty

V laboratoři jsou k dispozici dva typy autentizačních zařízení, která budou zejména využívána pro přihlášení do VPN a také do systému.

Prvním je USB token Ikey 2032. Obsahuje osmibitový procesor pro vykonávání kryptografických operací a dále 32kB paměti pro ukládání digitálních certifikátů a šifrovacích klíčů. Token podporuje dvoufaktorovou autentizaci, kdy druhým faktorem je PIN, který uživatel musí zadat pro použití tokenu. Token podporuje algoritmy RSA, DSA, DES, 3 DES, RC2, SH-1 a MD5. Privátní klíč jde vygenerovat přímo na tokenu a při vlastním používání nikdy neopouští token. Výhodou USB tokenu oproti čipovým kartám je mobilita. Uživatel jej může teoreticky využívat na každém počítači obsahujícím jen USB port. U čipových karet potřebujeme čtečku.

Dalšími autentizačními zařízeními jsou čipové karty SmartCard 330M a 330U. Ty obdobně jako USB token Ikey 2032, obsahují 32kB paměti a osmibitový procesor. Podporují také stejné kryptografické algoritmy. Model 330M má oproti modelu 330U podporu biometriky. Pro použití čipové karty je nutná čtečka. V laboratoři jsou k dispozici čtečky Omnikey CardMan USB 3121 a Precise Biometrics 200MC. Poslední jmenovaná čtečka umožňuje biometrickou autentizaci pomocí otisku prstu.

Oba druhy autentizačních předmětů je možně využít pro tzv. Single Sign-On (SSO). Jedná se o metodu řízení přístupu, která uživateli umožňuje se jednou autentizovat a získat tak přístup k prostředkům více softwarových aplikací (například operační systém, firemní intranetové aplikace, webové servery apod.). Toto řešení je vhodné například do firmy, kdy administrátor sítě může nastavit velmi přísnou politiku pro přihlašovací hesla. Hesla budou generovaná náhodně zvlášť pro každého uživatele, budou dostatečně silná a budou se v pravidelných intervalech měnit. Uživatel si tak nemusí pamatovat složitá hesla a pravidelně je měnit. Pro přihlášení do systému mu stačí token a PIN k němu.

4.8 Softwarový VPN klient

Na klientských počítačích v laboratoři je nainstalován High Assurance Remote (HARemote) VPN klient. Ten je založen na VPN standardu firmy Safenet. Poskytuje zabezpečenou komunikaci pomocí protokolu IPSec mezi dvěma klienty a nebo klientem a VPN bránou. Součástí tohoto programu je firewall a program pro správu a certifikátů.

4.9 Software pro práci s autentizačními předměty

Na serveru v laboratoři je nainstalován software SafeNet Borderless Security Administrative Management Center (AMC). Slouží k administraci SSO (více informací o Single Sign-On v kapitole 6.7), PKI a nastavení bezpečnostních politik. AMC není klientská aplikace, ale slouží k jejímu vytvoření. Po nastavení bezpečnostních politik aplikace AMC umožňuje vytvořit předkonfigurovaný instalační balíček určený pro klienty. V aplikaci je podpora pro vytvoření více balíčků s rozdílnými bezpečnostními politikami (např. pro klienty ve vnitřní síti LAN méně restriktivní politika a naopak přísnější politika pro vzdáleně se připojující klienty pomocí VPN).

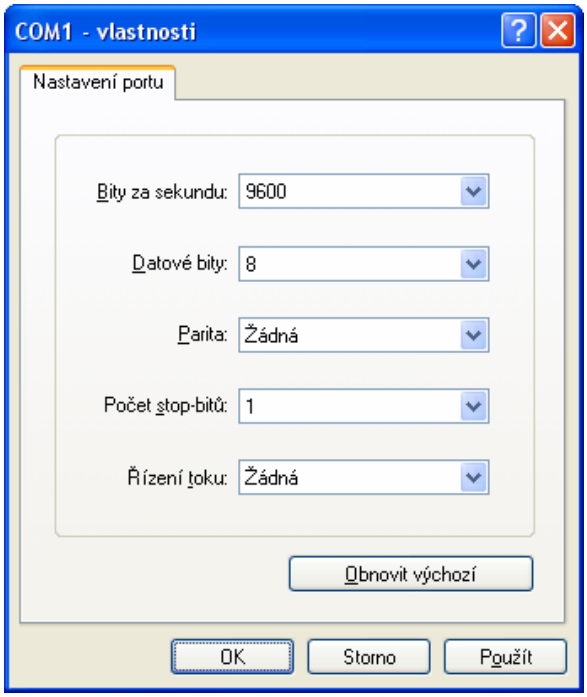
5. Postupy konfigurace

V této kapitole jsou uvedeny postupy, které jsou použity v jednotlivých laboratorních úlohách. Celá kapitola je koncipována jako souhrn modulárních postupů, ze kterých je možné jednoduše dle aktuálních požadavků a situace, složit laboratorní úlohy. Jsou zde uvedeny postupy, ve kterých si studenti vyzkouší konfiguraci a také funkci jednotlivých zařízení a aplikací v laboratorní síti. V této kapitole jsou také uvedeny postupy, ve kterých se studenti seznámí s bezpečnostními aspekty v síti, možnými útoky a také ochranou proti těmto útokům.

5.1 Konfigurace HA500 brány přes CLI

- Než je možné bránu HA500 v prostředí SMC nakonfigurovat, certifikovat a následně ji přiřazovat VPN politiky, je nutná tzv. prvotní konfigurace přes Command Line Interface (CLI).

- Pomocí sériového kabelu připojíme bránu ke stanici. Na stanici spustíme aplikaci HyperTerminal (*Start* → *Spustit*... → *hypertrm*). Parametry spojení nastavíme dle obr. 14. Po stisku klávesy Enter se připojíme.



Obr. 14: Parametry spojení

- Samotné prostředí je téměř identické s Cisco Internetwork Operating System (IOS). Do brány se přihlásíme pomocí uživatelského jména *admin* a hesla *safenet*.

- Tímto krokem jsme se dostali do tzv. neprivilegovaného módu, ve kterém jsou velmi omezená práva a není možné téměř nic měnit. Do privilegovaného módu se dostane po zadaní příkazu *enable* a následně zadáním hesla *safenet*.

- Stejně jako v Cisco IOS a podobně jako v UNIXových systémech příkazy není nutné psát celé. Po napsaní počátečních písmen příkazu a následně stisknutí tabulátoru zapříčiní vypsání celého příkazu. Pro další urychlení práce lze příkazy psát ve zkratkách (např. sh run = show running config, conf t = configure terminal). Pro zjištění možných atributů u příkazu, Odeslání příkazu, za který následuje mezera následovaná *?* nám poradí možné atributy příkazu.

- Pomocí příkazu *show running-config* si ověříme, že brána je v tzv. továrním nastavení. V konfiguračním souboru by tak neměly být uloženy žádné certifikáty, a žádné VPN politiky apod. Konfigurační soubor by měl vypadat takto:

```
hostname Security
enable password md5 encrypted dcf8fc099873b98c54d445aa62e0dc04
ip subnet-zero
ip classless
ip routing
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
logging email priority-level info
username admin password safenet
ip firewall
ip crypto
crypto ike policy 5000
 no initiate
 respond main
 peer any
 attribute 5000
  encryption 3des
  authentication pre-share
  group 2
  lifetime 86400
crypto ike remote-id address 0.0.0.0 255.255.255.255 preshared-key
12345678 no-m
ode-config no-xauth
crypto ipsec transform-set SMC-HIGH-SECURITY esp-3des esp-sha-hmac
 mode tunnel
crypto map PUBLIC-CRYPTO 5000 ipsec-ike
 match address SMC-PUBLIC
 set transform-set SMC-HIGH-SECURITY
 set security-association lifetime kilobytes 1024000
interface eth 01
 description Public
 access-policy PUBLIC-POLICY
 crypto map PUBLIC-CRYPTO
 no shutdown
interface eth 02
 description Private
 ip address  192.168.0.1  255.255.255.0
 access-policy PRIVATE-POLICY
 no shutdown
ip access-list extended MATCH-ALL
 permit ip any  any    log
 remark Match all IP packets
ip access-list extended SMC-PUBLIC
 permit ip any  any    log
 remark Secure from SMC to Public interface
ip policy-class PRIVATE-POLICY
```

```
  allow list MATCH-ALL
ip policy-class PUBLIC-POLICY
  allow list SMC-PUBLIC
  allow list MATCH-ALL
no ip n-form agent
no ip http server
no ip http secure-server
ip snmp agent
no ip ftp agent
snmp-server location SafeNet, Inc.
snmp-server enable traps
snmp-server source-interface ethernet 01
snmp-server community private RO
line con 0
  login local-userlist
line telnet 0 4
  login local-userlist
end
```

- Pokud zjistíme, že brána v tomto továrním nastavení není, tak jí do něj pomocí příkazu *faktory-default* v privilegovaném módu přivedeme.

- Můžeme vidět, že i v tomto továrním nastavení jedna VPN politika je. Ta slouží pro vytvoření šifrovaného kanálu, přes který proběhne prvotní připojení SMC k VPN bráně (přes veřejné – public rozhraní).

- Aby bylo možné toto prvotní připojení uskutečnit, je nutné nastavit veřejnému rozhraní správnou IP adresu, kterou dle naší architektury sítě budeme používat. Vnitřní rozhraní pak již můžeme nastavit v SMC.

- Z privilegovaného módu se přesuneme do globálního konfiguračního módu pomocí příkazu *configure terminal*. Zde pomocí příkazu *interface ethernet 0/1* se přesuneme do nastavení veřejného rozhraní. IP adresu nastavíme příkazem *ip address i.i.i.i m.m.m.m* kde i.i.i.i je ip adresa a m.m.m.m je maska. Příkazem *no shutdown* rozhraní zapneme. Naše nastavení uložíme příkazem *copy running-config startup-config* v privilegovaném módu.

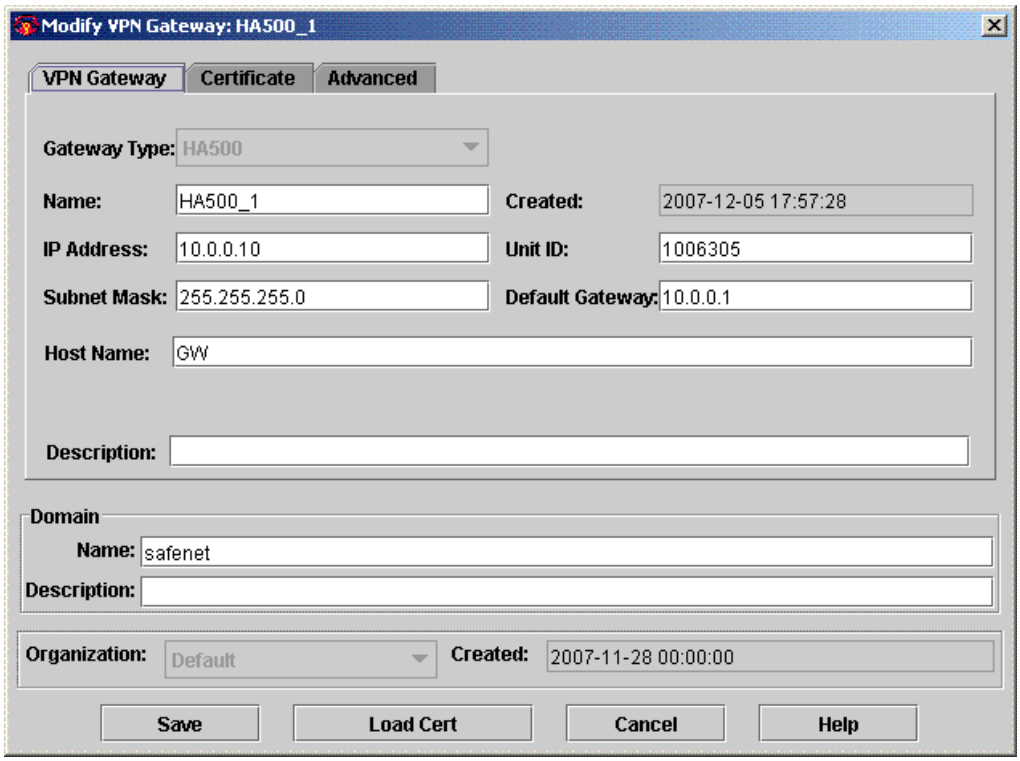
- V konfiguračním módu nastavíme výchozí bránu pomocí příkazu *ip route 0.0.0.0 0.0.0.0 i.i.i.i*, kde i.i.i.i je ip adresa výchozí brány. Nepoužívat příkaz *ip default-gateway*.

- Pozn. z vyššího módu do nižšího se vrátíme příkazem *exit*. Přímo do privilegovaného módu z vyššího se vrátíme pomocí *end*.

5.2 Přidání a nastavení parametrů VPN brány v prostředí SMC

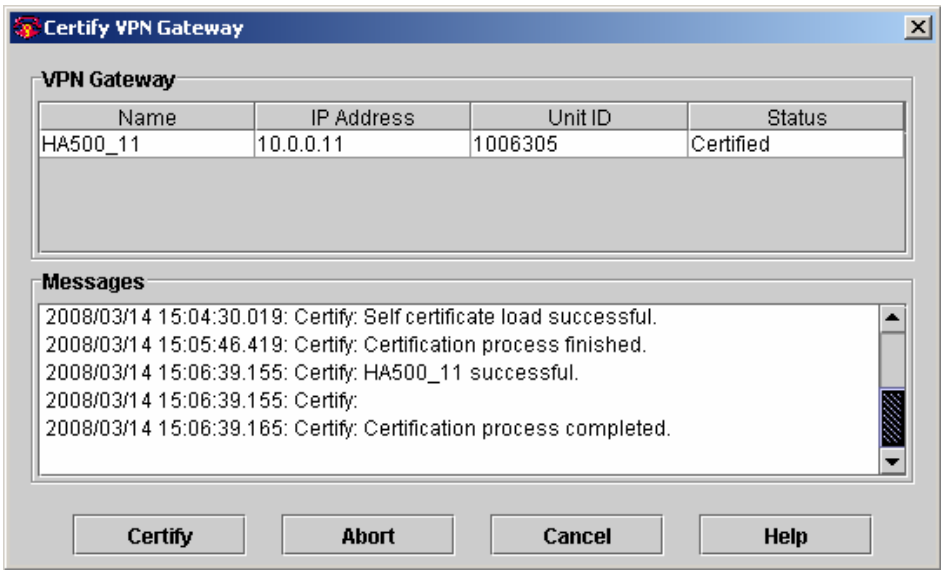
- Spustíme SMC (uživatelské jméno *admin* heslo *Safe.Net3*)

- Na root mapě pomocí předposlední ikony z pravé strany *Add VPN Gateway* přidáme VPN bránu. Do okna, které se objeví vyplníme hodnoty dle zapojení naší sítě (pozn. IP adresou je myšlena adresa vnějšího rozhraní, tj. rozhraní přes které SMC s bránou komunikuje). Typ brány změníme na HA500. Unit ID je sériové číslo VPN brány, které zjistíme ze štítku, umístěného na spodní straně konfigurované brány. Příklad nastavení viz obr. 15.



Obr. 15: Příklad nastavení brány HA500

- Nastavení uložíme a okno zavřeme. Na základě sériového čísla Unit ID, které jsme vyplnili v minulém kroku, interní CA vydá bráně certifikát. Po této certifikaci je brána tzv. v síti důvěryhodná a je jí možné využívat. Certifikaci provedeme tak, že na root mapě klikneme na bránu pravým tlačítkem a z nabídky vybereme volbu *Certify*. V nově otevřeném okně spustíme proces certifikace pomocí možnosti *Certify*. Tento proces trvá cca několik minut. Okno zavřeme až když uvidíme hlášku *Certification process completed* (viz obr. 16).



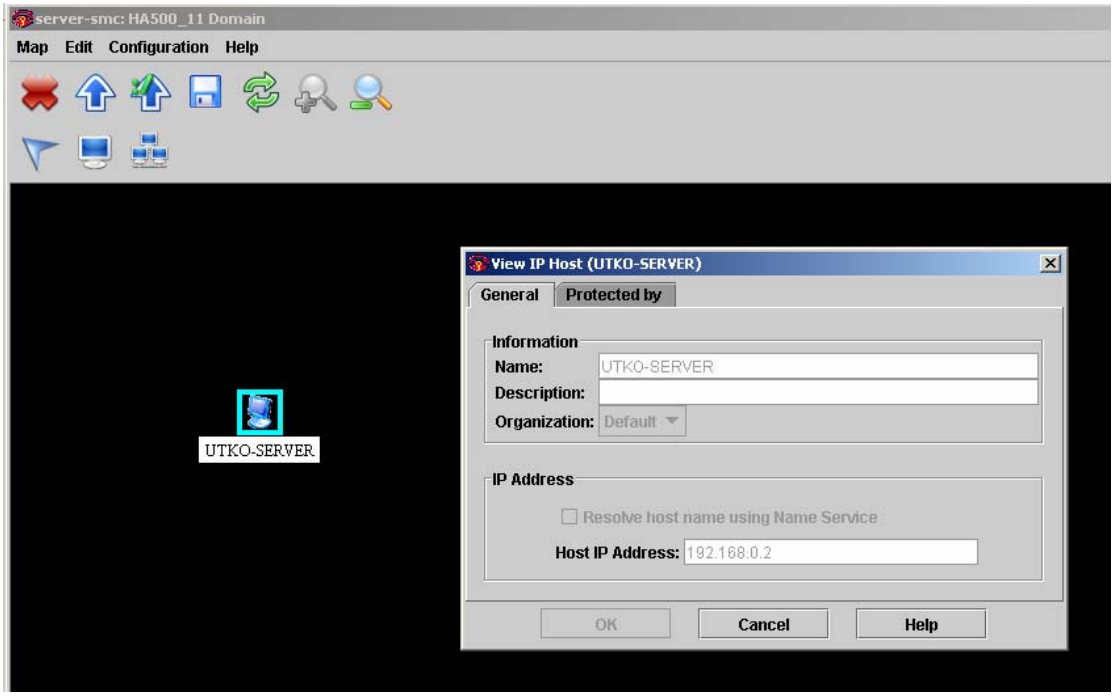
Obr. 16: Certifikace brány HA500

- Pokud se opět připojíme přes aplikaci Hyperterminal a zobrazíme si konfigurační soubor (postup viz začátek kapitoly), měli bych vidět v konfiguračním souboru dva certifikáty. První certifikát je kořenový certifikát interní CA a druhý je certifikát vydaný interní CA pro danou bránu.

- Dvojklikem na ikonu brány v root mapě se na bránu připojíme. V sekci *Configuration* nastavíme správnou IP adresu a také masku vnitřnímu rozhraní.

- Na root mapě vidíme, že ke každé bráně je připojena doména. Do domény přidáme stanici, která je za onou bránou (stanice připojená na privátní rozhraní brány). Dvojklikem na rootmapě na ikonu domény příslušné brány se

dostaneme do podmapy, do které vložíme danou stanici a nastavíme jí příslušnou IP adresu. Příklad na obr. 17.

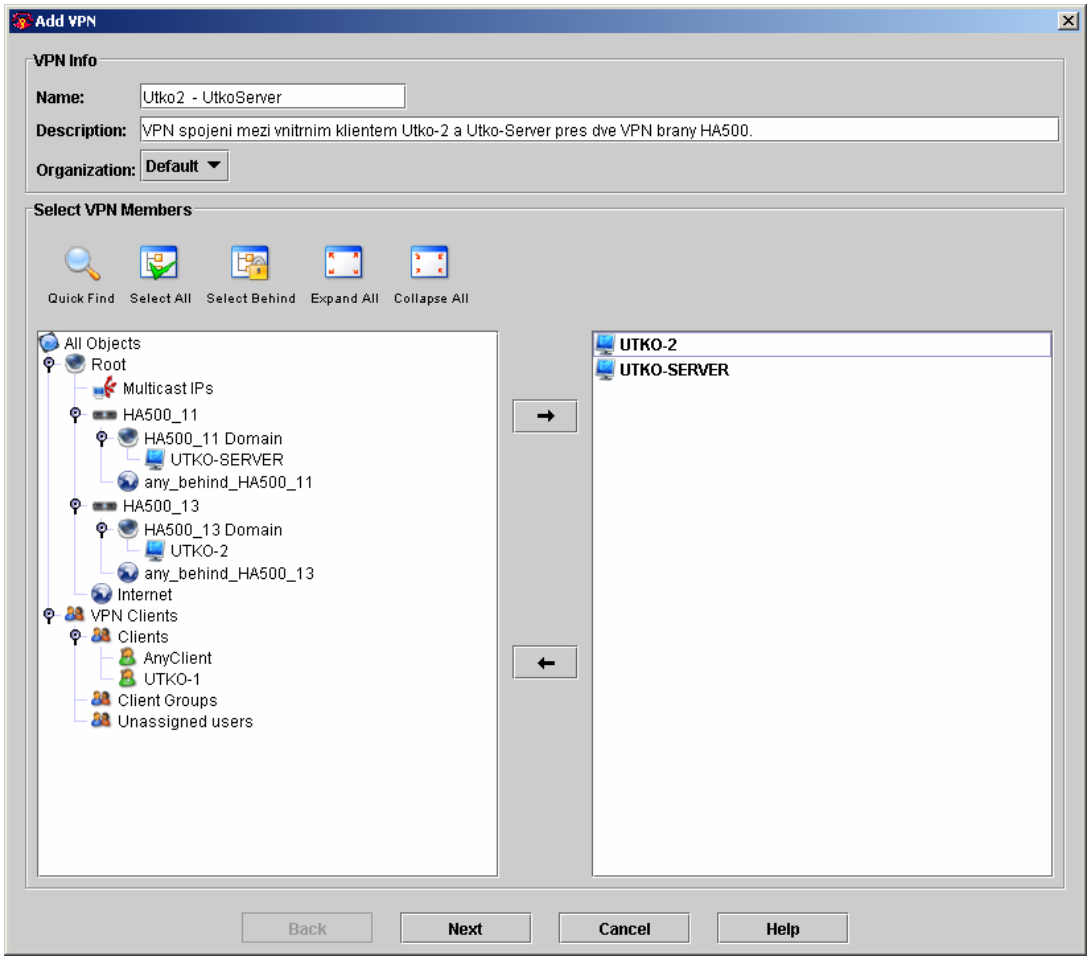


Obr. 17: Přidání stanice do domény

- Po těchto krocích je brána připravena pro nahrání VPN politik.

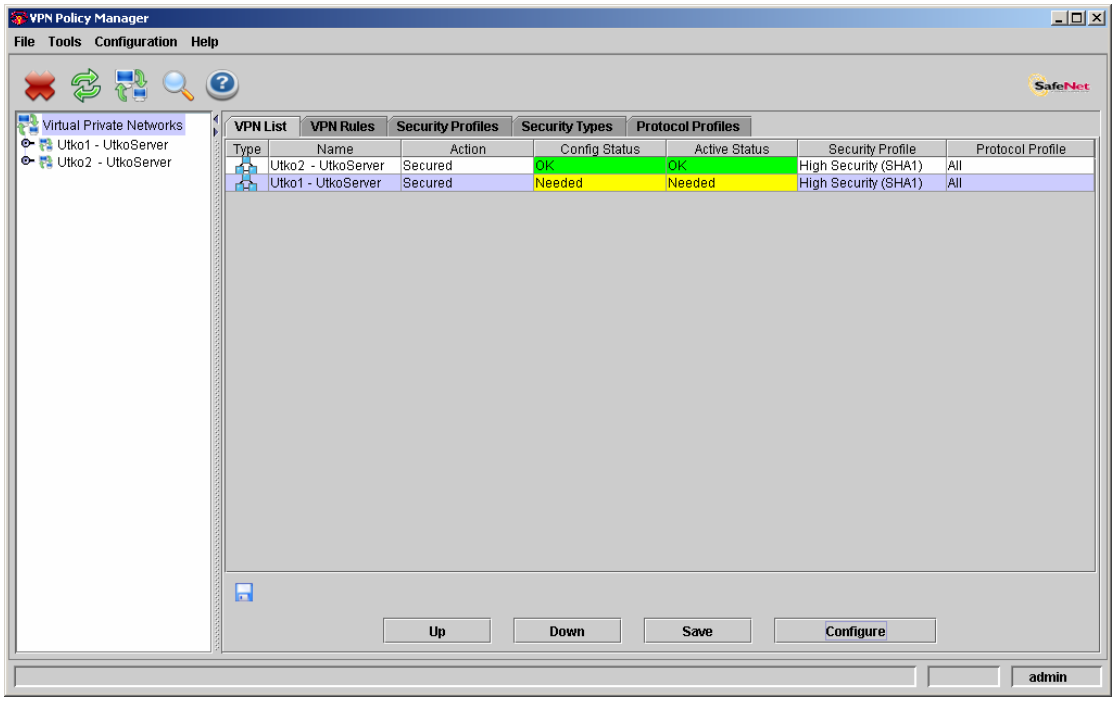
5.3 Definice VPN politik

- Abychom mohli data posílat po zabezpečeném kanále, musíme nejdřív tento kanál definovat, respektive definovat VPN politiku. VPN politika je v podstatě sada pravidel, která definuje VPN tunel. Je v ní uvedeno mezi jakými zařízeními je VPN tunel realizován, jaká autentizační metoda bude použita, či jaké šifrovací metody se použijí. V horní liště v aplikaci SMC v sekci *Security* vybereme *VPN Policy Manager*. Zvolením *Create a new VPN* vytvoříme novou definici VPN. Politiku si pojmenujeme a popřípadě přidáme i popis. Z nabídky na levé straně vybereme ty objekty, mezi kterými chceme vytvořit šifrovaný tunel. Např. viz obr. 18.



Obr. 18: Definice politiky

- Na následující obrazovce ponecháme volby jak jsou a volbou *Finish* se nám daná definice VPN spojení vytvoří. Následně je nutné tuto definici nahrát do VPN brány, popřípadě do více bran pokud se daná definice týká více bran. To provedeme tak, že v hlavním okně sekce *VPN Policy Manager* klikneme na levé straně na položku *Virtual Private Networks*. V pravém okně se zobrazí seznam VPN politik, které jsou v SMC nakonfigurované. Zelenou barvou jsou označeny ty, které jsou již nahrané v příslušných branách a jsou aktivní. Nově přidaná definice je označena žlutou barvou. Viz obr. 19.



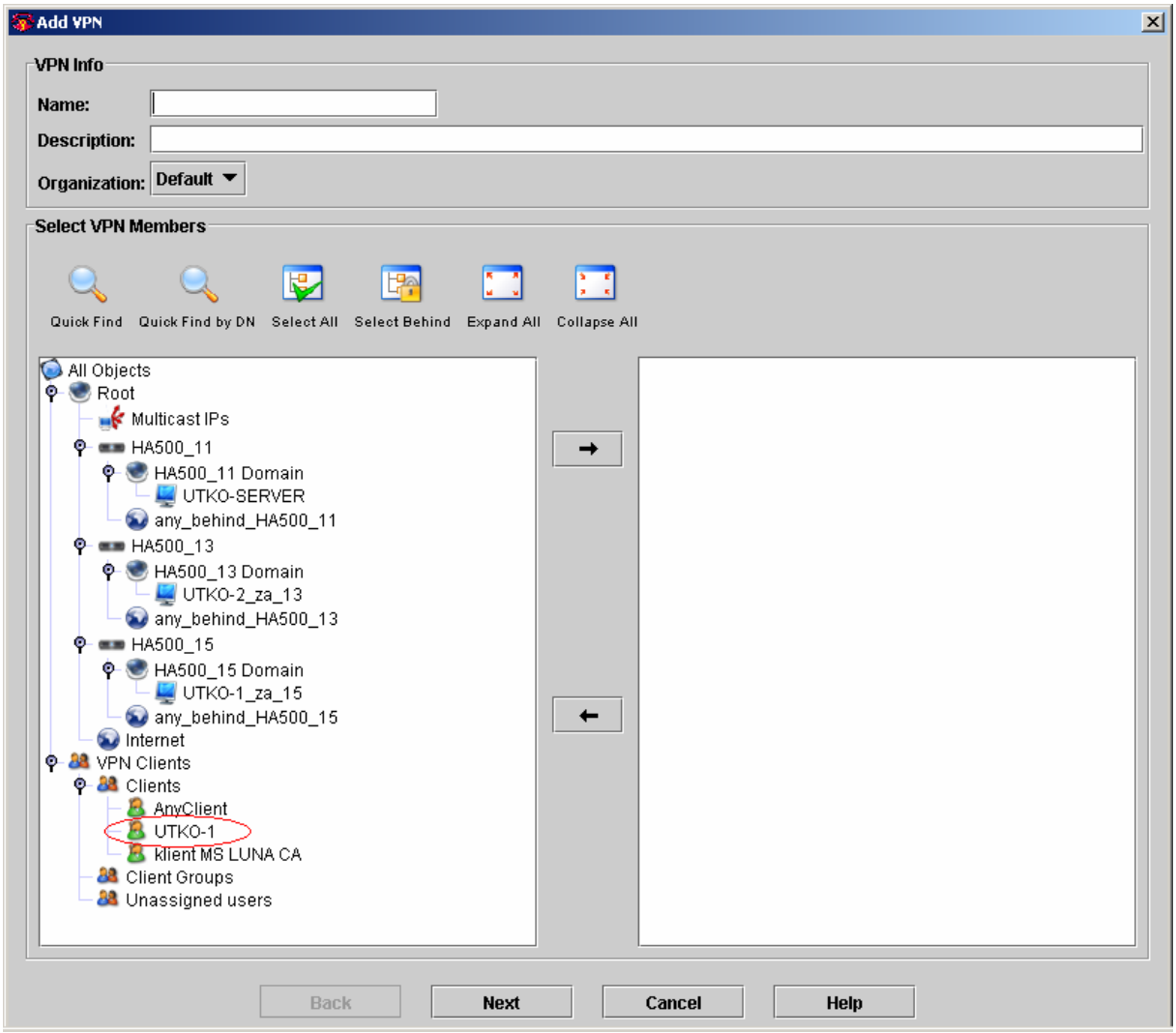
Obr. 19: Zobrazení VPN politik

- Na spodní liště zvolíme položku *Configure*. V nově otevřeném okně zvolíme brány, ve kterých chceme VPN politiky aktualizovat. Pro aktualizování jen těch bran, kterých se týkají nově přidané VPN definice, zvolíme volbu *Changed Devices*. Položkou *Download* VPN definici nahrajeme do daných bran. Tento proces může trvat i několik minut.

- Ověříme zda příslušné stanice mají mezi sebou konektivitu. Například pomocí programu Ping.

5.4 Vytvoření klienta HARemote

- Aby se do sítě mohli připojovat i vzdálení uživatelé pomocí svých SW VPN klientů, je zapotřebí v prostředí SMC definovat tyto klienty. V menu *Security* vybereme *VPN Client Manager*. Zvolíme *Create a new Client*. Další volby neměníme a necháme v jejich základním nastavení. Tímto je klient vytvořen a od této chvíle je klient k dispozici pro definici příslušné VPN.



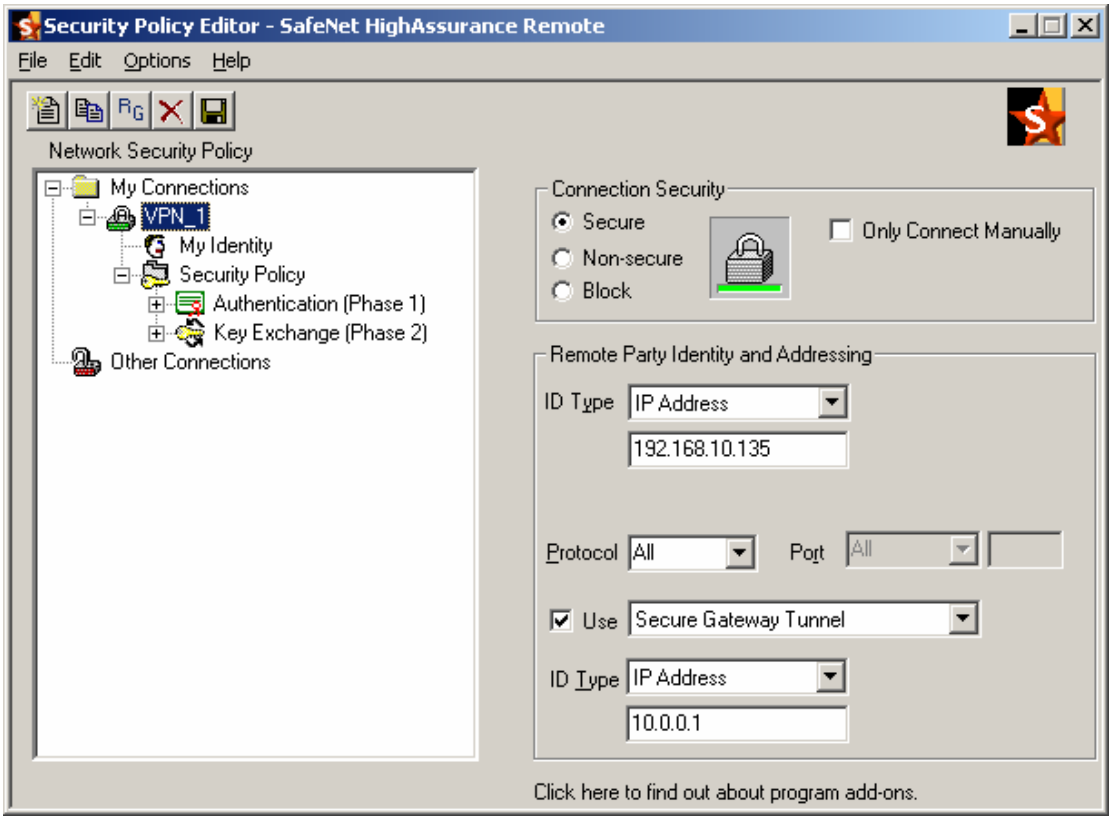
Obr. 20: Přidání VPN klienta

5.5 SW VPN klient HARemote

- Vzdálený uživatel se připojuje do sítě pomocí SW klienta, kterým je v tomto případě program HARemote. Ten se automaticky spouští hned po startu systému. Pokud je správně nadefinováno spojení, tak se v případě požadavku i sám připojí (je možná i volba manuálního spojení. Pro definováni spojení vybereme volbu *Security Policy Editor* po kliknutí pravým tlačítkem myši na ikonu SW VPN klienta v tray liště.

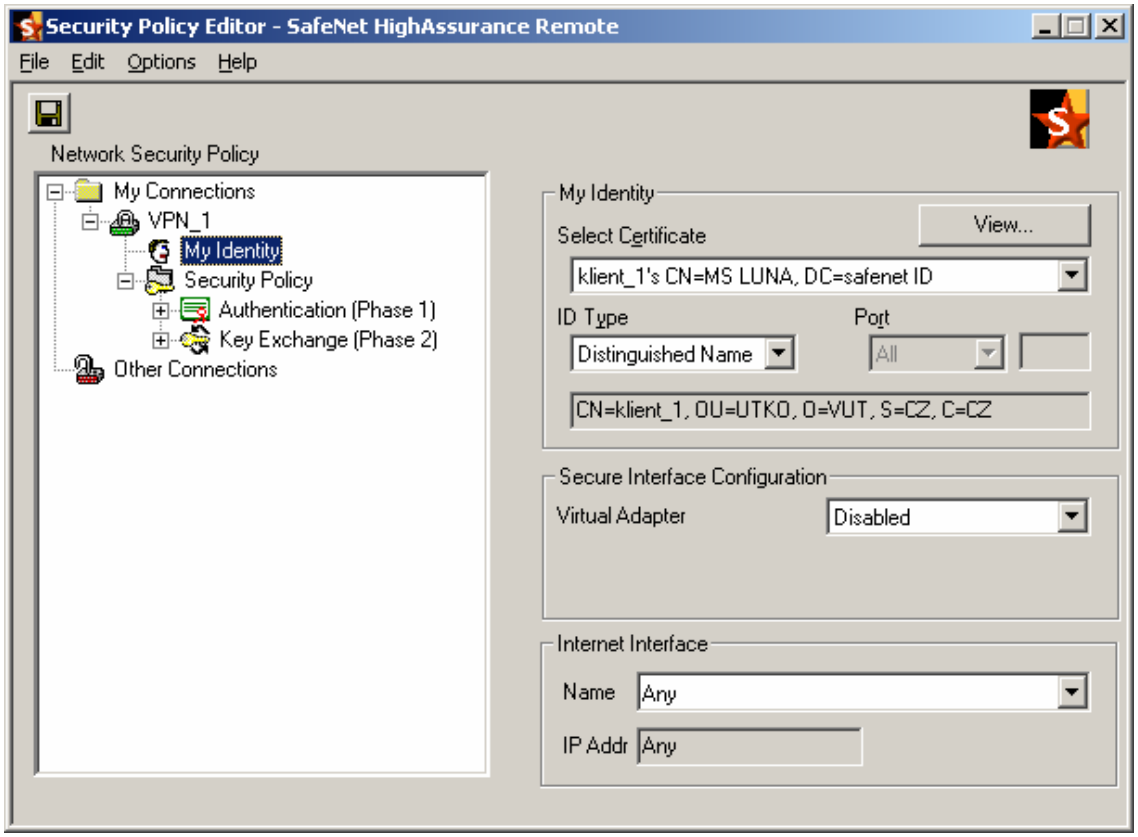
Obr. 21: HARemote tray lišta

- V menu *Edit* zvolíme *Add→Connection*. Položku *Connection Security* nastavíme na *Secure* protože se budeme připojovat pomocí zabezpečeného IPSec tunelu. U položky *Remote Party Identity and Addressing* zvolíme možnost *IP Address* a definujeme IP adresu stanice, ke které se chceme připojit. Protože se budeme připojovat přes VPN bránu, tak zvolíme *USE→Secure Gateway Tunnel* a *ID Type* změníme na typ *IP Address*, a do tohoto pole napíšeme IP adresu vnějšího rozhraní VPN brány přes kterou se chceme připojovat.



Obr. 22: HARemote definice spojení

- Pod položkou *My identity* máme možnost si zvolit, jakým způsobem se budeme protější straně autentizovat. Možnosti jsou tu dvě - pomocí certifikátu nebo pomocí předsdíleného klíče (na obou stranách komunikace musí být zvolen stejný klíč). Pokud zvolíme možnost certifikátu, můžeme dále blíže specifikovat svoji identitu parametry uvedenými v našem certifikátu (např. Common name, Organization unit, E-mail apod). Pokud jsme si v SMC na definovali klienta tak, že se bude kontrolovat hodnota pole e-mail v certifikátu, musíme ji zde nastavit. V opačném případě bychom se nemohli připojit, a to i přesto že máme důvěryhodný certifikát, kterému opačná strana komunikace věří.

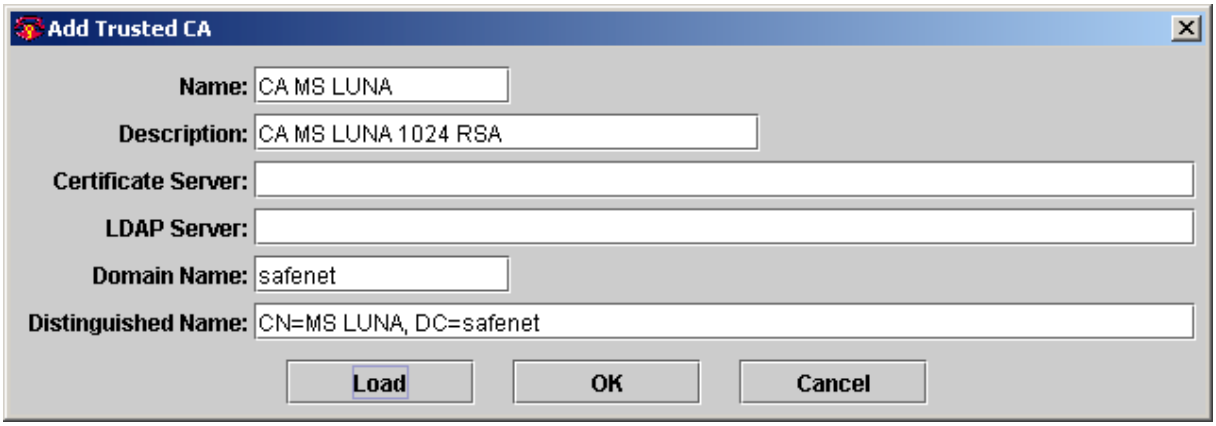


Obr. 23: HARemote definice klienta

- U položky *Other Connections* je možné zvolit, co má integrovaný firewall HARemote provést s ostatní komunikací (tj. veškerou ostatní komunikací mimo VPN). Pokud chceme, aby naší stanici opouštěla a také do ní vstupovala pouze zabezpečená komunikace, tak zvolíme možnost *Block*.

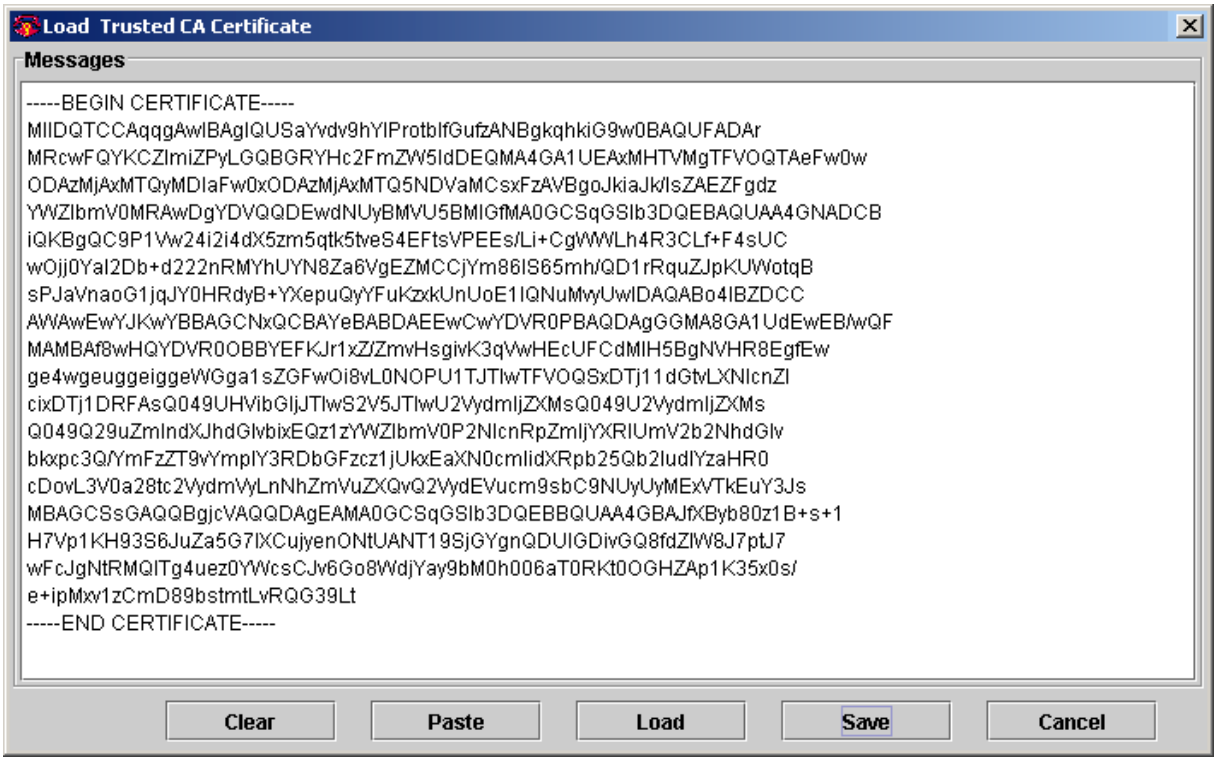
5.6 Klient s certifikátem vydaným jinou CA

- Aby se do sítě mohli přihlašovat vzdálení klienti pomocí certifikátů, které nejsou vydány vnitřní CA SMC ale jinou CA, musíme této CA důvěřovat. Respektive VPN brány a SMC jí musí důvěřovat. To provedeme tak, že v SMC v sekci *Security* zvolíme položku *Certificate Authority*. Tímto jsme se ocitli v sekci interní CA. Zde je potřeba přidat onu zatím nedůvěryhodnou CA do seznamu důvěryhodných CA. Ve složce *Trusted CAs* zvolíme *Add*. Vyplníme položky dle dané CA (např. jako na obr. 24).



Obr. 24: Přidání CA

- Následujícím krokem je nahrání kořenového certifikátu nové CA (položka *Load*). Zde je důležité poznamenat, že je nutné do SMC vložit kořenový certifikát s klíčem o maximální délce 1024 bitů a to i přesto, že z hlediska bezpečnosti je dobré volit klíče kořenové CA delší. Důvodem je fakt, že v laboratoři máme VPN brány HA500, které podporují maximální délku klíče 1024 bitů. Delší klíče je možné nahrát až do bran HA4000. Pro nahrání kořenového certifikátu zvolíme možnost *Load* a vybereme onen certifikát (pokud jej nemáme, tak si jej vyexportujeme ve formátu BASE64 ze serveru, kde daná CA je spuštěna).



Obr. 25: Nahrání kořenového certifikátu

- Uložíme a potvrdíme. Po tomto kroku již SMC důvěřuje dané CA. Jelikož při navazování spojení provádí autentizaci brána, je nutné, aby i brána o této CA věděla (SMC bohužel automaticky tento certifikát do brány nenahraje a je nutné jej nahrát manuálně).

- V CLI se dostaneme do konfiguračního módu brány. Vytvoříme nový CA profile příkazem *crypto ca profile „název profilu“*. Příkazem *crypto ca certificate chain "název řetězce"* definujeme certifikační řetězec. Do tohoto řetězce nyní vložíme certifikát dané CA. Napíšeme příkaz *certificate ca X*, kde X je sériové číslo daného certifikátu. Nyní si v textovém editoru otevřeme onen kořenový certifikát CA (pozn. musí být v Base64 kódování). Obsah zkopírujeme do schránky a vložíme do CLI brány. Sled příkazů by měl vypadat například takto:

crypto ca profile "MS LUNA"

!

crypto ca certificate chain "MS LUNA"

certificate ca 03

-----BEGIN CERTIFICATE-----

MIIDQTCCAqqgAwIBAgIQUSaYvdv9hYlProtbIfGufzANBgkqhkiG9w0BAQUFAD
ArMRcwFQYKCZImiZPyLGQBGRYHc2FmZW5ldDEQMA4GA1UEAxMHTVMgT
FVOQTAeFw0wODAzMjAxMTQyMDlaFw0xODAzMjAxMTQ5NDVaMCsxFzAVB
goJkiaJk/IsZAEZFgdzYWZlbmV0MRAwDgYDVQQDEwdNUyBMVU5BMIGfMA0
GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC9P1Vw24i2i4dX5zm5qtk5tveS4EF
tsVPEEs/Li+CgWWLh4R3CLf+F4sUCwOjj0YaI2Db+d222nRMYhUYN8Za6VgEZ
MCCjYm86lS65mh/QD1rRquZJpKUWotqBsPJaVnaoG1jqJY0HRdyB+YXepuQyY
FuKzxkUnUoE1lQNuMvyUwIDAQABo4IBZDCCAWAwEwYJKwYBBAGCNxQCB
AYeBABDAEEwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wHQ
YDVR0OBBYEFKJr1xZ/ZmvHsgivK3qVwHEcUFCdMIH5BgNVHR8EgfEwge4wg
euggeiggeWGga1sZGFwOi8vL0NOPU1TJTIwTFVOQSxDTj11dGtvLXNlcnZlcixD
Tj1DRFAsQ049UHVibGljJTIwS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ0
49Q29uZmlndXJhdGlvbixEQz1zYWZlbmV0P2NlcnRpZmljYXRlUmV2b2NhdGlvbk
xpc3Q/YmFzZT9vYmplY3RDbGFzcz1jUkxEaXN0cmlidXRpb25Qb2ludIYzaHR0cD
ovL3V0a28tc2VydmVyLnNhZmVuZXQvQ2VydEVucm9sbC9NUyUyMExVTkEuY3
JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEBBQUAA4GBAJfXBy
b80z1B+s+1H7Vp1KH93S6JuZa5G7IXCujyenONtUANT19SjGYgnQDUIGDivGQ
8fdZlW8J7ptJ7wFcJgNtRMQITg4uez0YWcsCJv6Go8WdjYay9bM0h006aT0RKt0O
GHZAp1K35x0s/e+ipMxv1zCmD89bstmtLvRQG39Lt

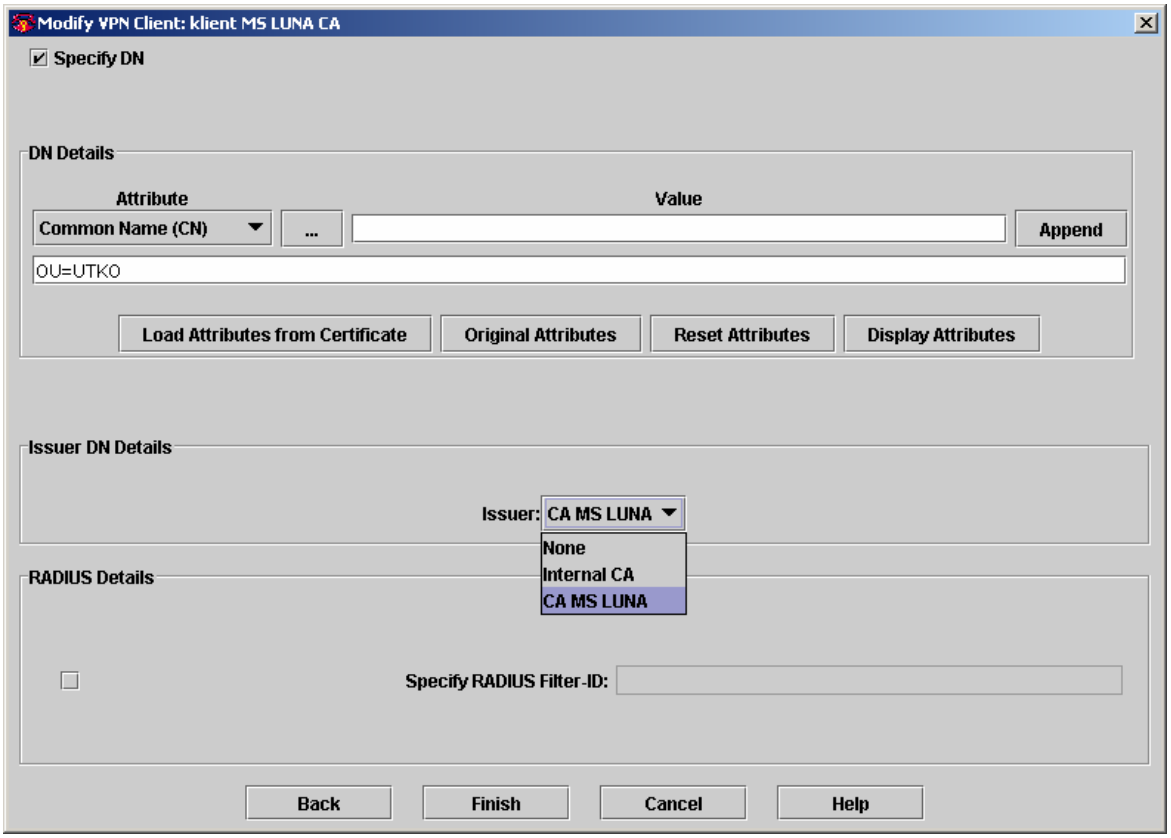
-----END CERTIFICATE-----

quit

!

- Dalším krokem je definice klienta (klientů), který bude mít certifikát vydaný onou novou CA. V menu *Security* vybereme *VPN Client Manager*. Vybereme volbu *Create a new Clien*". Vyplníme údaje o klientovi. IP adresu není potřeba specifikovat, klient bude ověřován jen na základě jeho certifikátu. Na následující obrazovce ponecháme zvolenou možnost RSA. Na další obrazovce

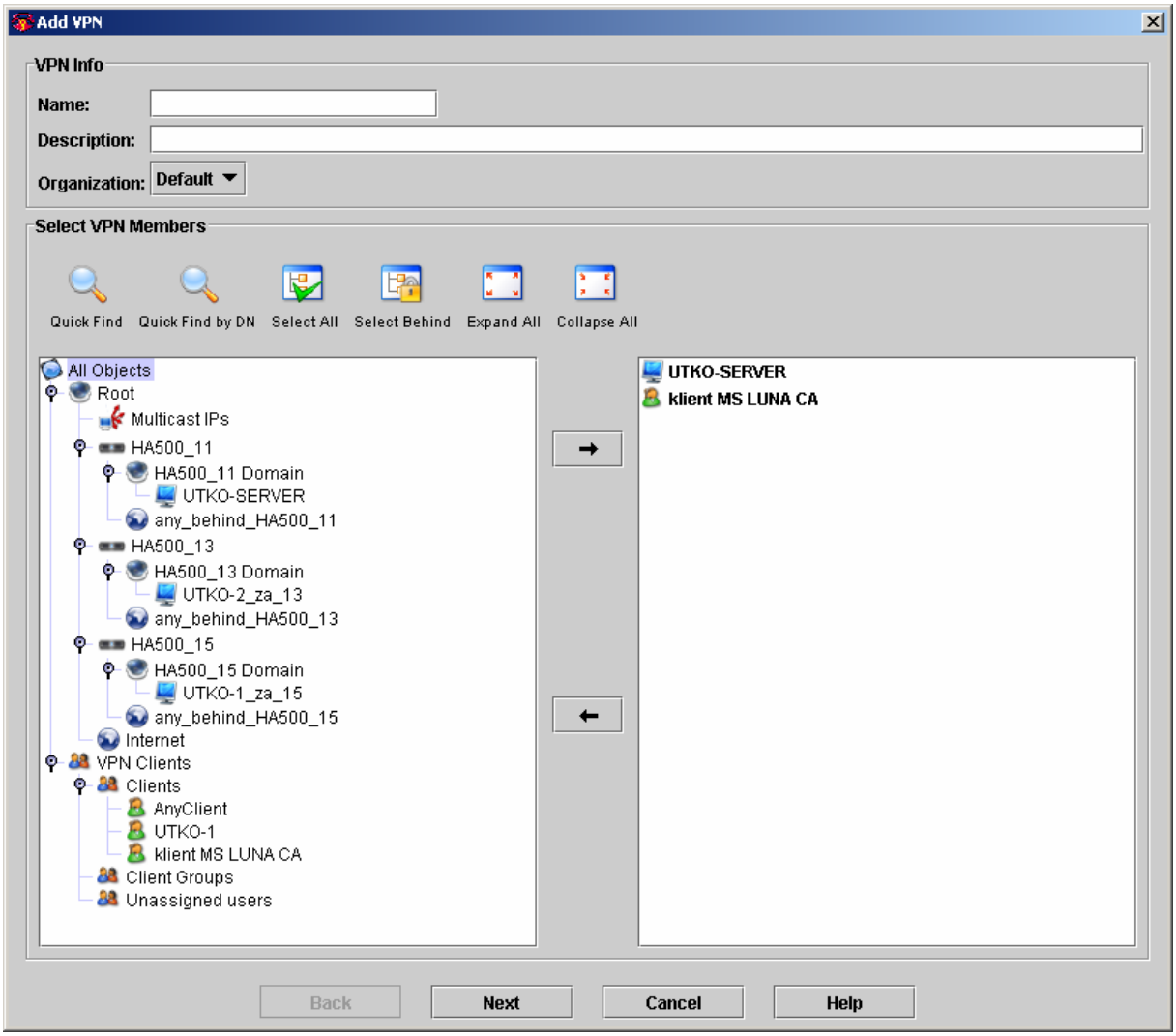
zvolíme možnost *Specify DN* a vybereme jaké prvky certifikátu chceme, aby byly kontrolovány při autentizaci. Dále je potřeba definovat vydavatele certifikátu (onu novou CA). To provedeme tak, že v položce *Issuer* vybereme onu novou důvěryhodnou CA (viz obr. 26).



Obr. 26: Definice klienta

- Na ukázce je zvolen jen parametr *OU=UTKO* a *Issuer CA MS LUNA*. To znamená, že jsme tímto definovali všechny klienty, kteří mají certifikát vydaný certifikační autoritou *CA MS LUNA* a zároveň mají ve svém certifikátu uvedenou položku *OU* (Organization Unit) rovnu hodnotě *UTKO*.

- Nyní je ještě potřeba nadefinovat příslušnou VPN politiku mezi tímto klientem (skupinou klientů) a druhou stranou (viz obr. 27).



Obr. 27: Definice VPN politiky s novým klientem

- V SW VPN klientu HARemote již jen stačí vybrat správný certifikát a připojit se k VPN bráně.

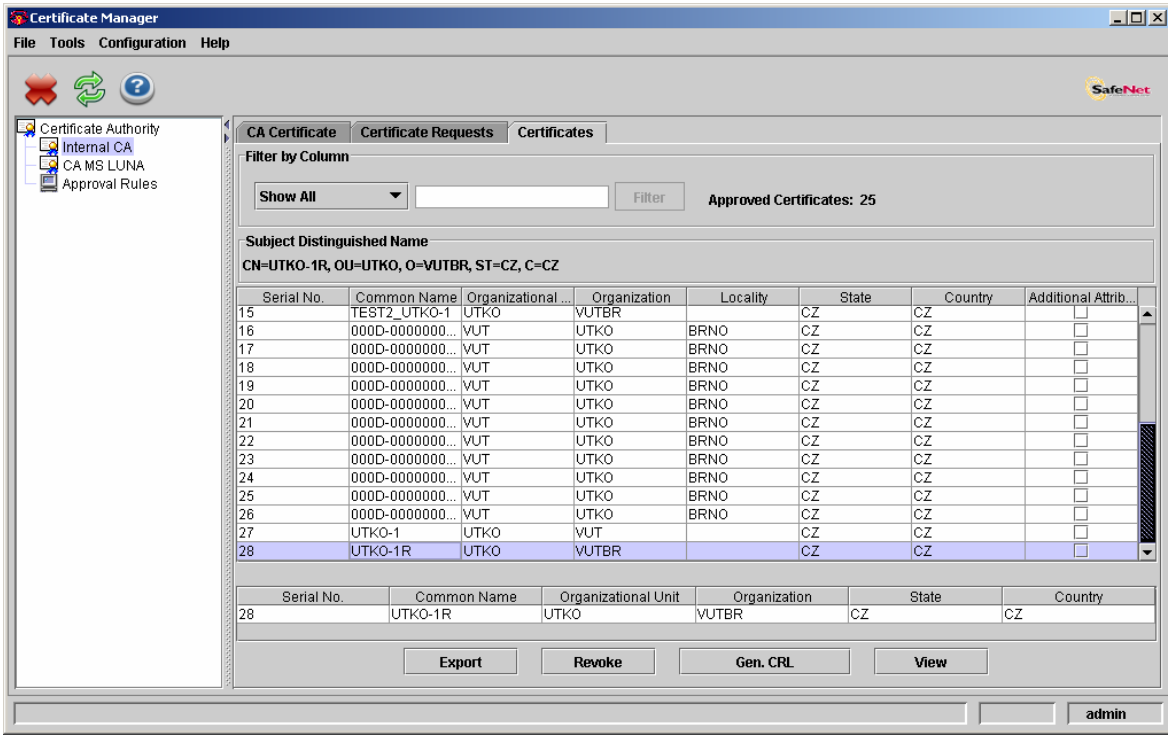
5.7 Ověření CRL

- Může se stát, že dojde ke kompromitaci soukromého klíče, jenž je do páru s veřejným klíčem obsaženým v certifikátu. To může mít katastrofální následky. Je to analogické např. ke ztrátě kreditní karty, která může být následně zneužita. V případě kreditní karty je nutné co nejdříve informovat o této situaci

příslušnou banku, aby danou kartu zablokovala. Stejně tak při kompromitaci certifikátu, je nutné informovat CA, která nám certifikát vydala. CA následně umístí tento certifikát (respektive sériové číslo daného certifikátu) na seznam revokovaných certifikátů Certificate Revocation List (CRL). V případě VPN je nutné tento CRL nahrát do HW VPN bran či do SW VPN klienta, aby tento certifikát při pokusu o spojení odmítli.

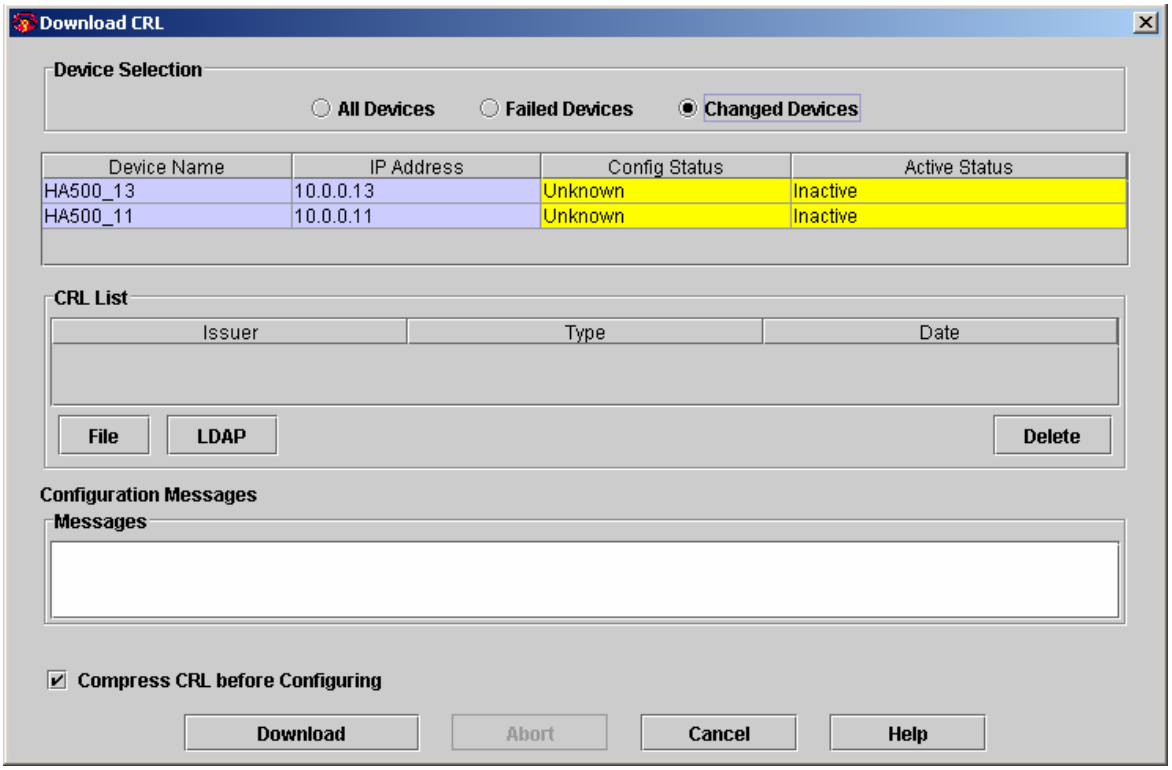
- V následující části si vyzkoušíme celý tento postup. Tedy revokaci certifikátu, vytvoření CRL, importovaní CRL do příslušné VPN brány a jako poslední pokus o připojení s revokovaným certifikátem do VPN.

- V sekci *Certificate Manager* vybereme certifikát, který chceme revokovat, a zvolíme *Revoke*. Pomocí volby *Gen CRL* si vygenerujeme a následně uložíme CRL.



Obr. 28: Výběr certifikátů k revokaci

- V témže okně volbou *Tools* → *CRL Download* se dostaneme do okna, ve kterém budeme následně moci nahrát námi vytvořený CRL do VPN bran.



Obr. 29: Nahrání CRL do VPN brány

- V tomto okně si vybereme VPN brány, do kterých chceme CRL importovat. Tedy ty, přes které by se útočník mohl s kompromitovaným certifikátem přihlásit. Po té vybereme CRL, který jsme si v předchozím kroku vygenerovali a jako poslední volbou *Download* daný CRL importujeme do příslušných VPN bran.

- Po úspěšném nahrání CRL do VPN brány se pokusíme pomocí revokovaného certifikátu přihlásit do naší VPN. V SW VPN klientu vybereme příslušný revokovaný certifikát a pokusíme se připojit na VPN bránu, do které jsme importovali CRL.

- Připojení se nám nepodaří. V logu SW VPN klienta zjistíme, že používáme neplatný certifikát:

4-10: 11:19:49.765 My Connections\VPN - Using configured machine certificate "UTKO-1R's UTKO VUT ID".

*4-10: 11:19:49.812 My Connections\VPN - SENDING>>>> ISAKMP OAK MM *(ID, CERT, CERT_REQ 2x, SIG, NOTIFY:STATUS_REPLAY_STATUS, NOTIFY:STATUS_INITIAL_CONTACT)*

*4-10: 11:19:49.906 My Connections\VPN - RECEIVED<<< ISAKMP OAK INFO *(HASH, NOTIFY:**INVALID_CERT**)*

5.8 Analýza protokolu IKE

- V této kapitole se prakticky podíváme, jak IPSec navazuje spojení a vytváří zabezpečený tunel a jaké jsou možnosti při jeho vytváření. Teoretický úvod je uveden v kapitole 3.3.

- Pro vlastní analýzu navazování spojení použijeme program Wireshark (dříve Ethereal) (viz lit. [29]). Jedná se o grafický nástroj sloužící pro odchytávání paketů, analýzu síťových protokolů, pro studium síťového provozu či ladění problému v počítačové síti. Jedná se o open source produkt šířený pod licencí General Public License (GNU). Wireshark podporuje Unixové i Windows operační systémy.

- Budeme často měnit nastavení VPN. Protože při použití VPN bran by každá změna nastavení spočívala v nahrání nové politiky do brány a tato operace trvá poměrně dlouho, budeme vytvářet zabezpečený VPN tunel jen mezi dvěma SW VPN klienty (HARemote). Pro ukázku funkce jednotlivých možností nastavení je tento postup dostačující.

Obr. 30: Schéma zapojení pro analýzu protokolu IKE

- V nastavení politiky v HARemote zvolíme blokování nezabezpečené komunikace. Tzn. bude povoleno vysílání i příjem pouze VPN dat, ostatní komunikace bude zahozena. V *Security Policy Editor* u položky *Other Connections* vybereme volbu *Block*.

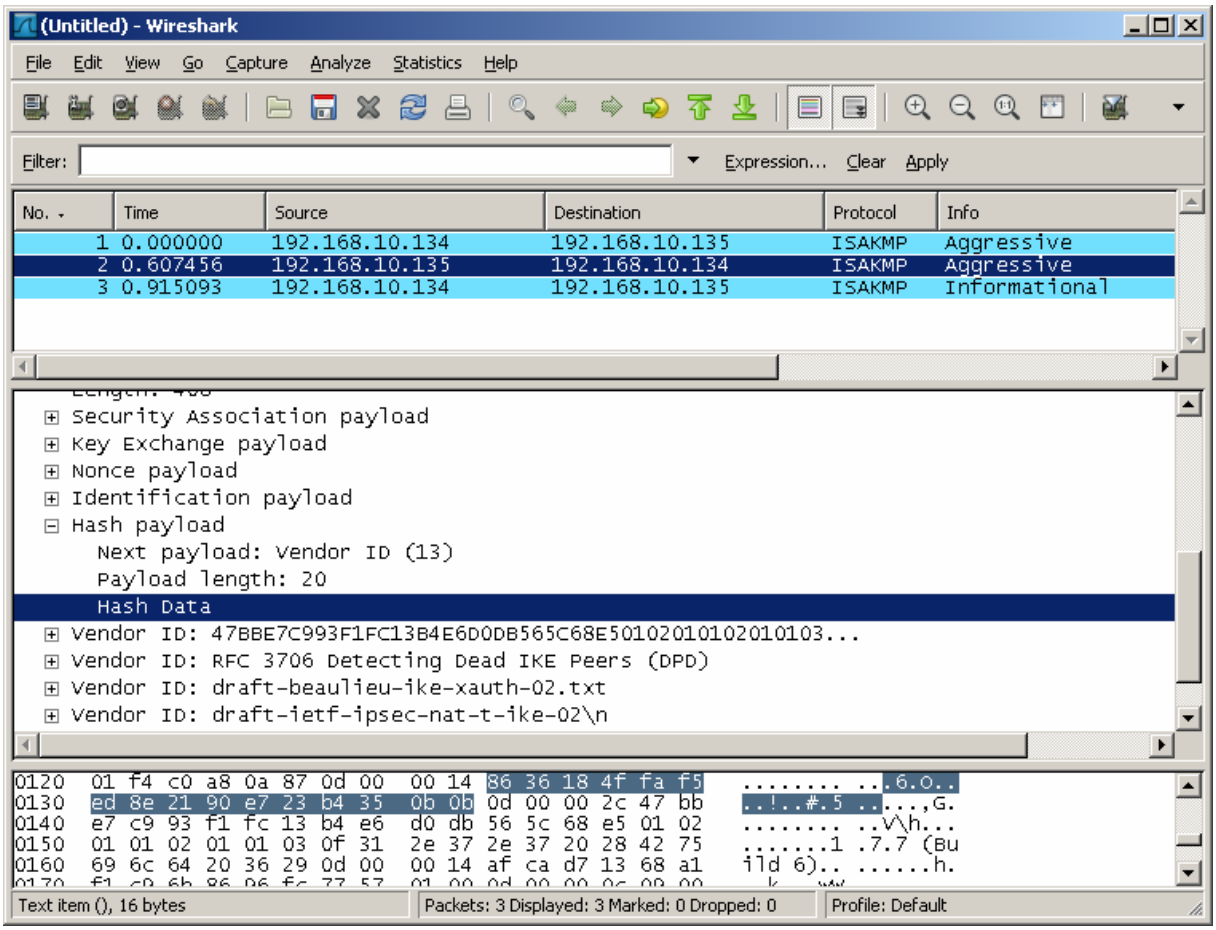
- Spustíme program Wireshark. V sekci *Capture* zvolíme možnost *Options*. Vybereme si rozhraní na kterém chceme odchytávat komunikaci a odškrtneme možnost *Capture packets in promiscuous mode* (zajímají nás pouze pakety určené pro naši stanici). Volbou *Start* spustíme odchytávání. Připojíme se svým klientem na druhý počítač a vygenerujeme nějakou komunikaci (např. Ping). V programu Wireshark uvidíme zachycené pakety, jenž můžeme analyzovat.

- Vyzkoušejte odchytit navazování spojení a následnou komunikaci při těchto možnostech:

    - protokol ESP / AH
    - hlavní režim / agresivní režim
    - autentizace pomocí předsdíleného klíče / certifikátu
    - režim tunel / transparent

- Pokuste se určit rozdílnosti komunikace při nastavení jednotlivých možností. Zaměřte se na to, ve kterých paketech se vyměňují informace o nabízených možnostech komunikace, ve kterých paketech se ustavuje pomocí algoritmu Diffie-Hellman klíč relace, kolik se celkem vymění paketů při navazování

spojení v jednotlivých režimech, kdy začne být komunikace šifrována a v jakém nastavení se nešifrují přenášená data.



Obr. 31: Ukázka zachycení komunikace programem Wireshark

5.9 Útok na agresivní mód protokolu IKE a předsdílený klíč

- V této kapitole využitím slabin protokolu IKE zjistíme tajný předsdílený klíč PSK (Pre-Shared Key) VPN serveru (viz lit. [28]).

- Teoretický úvod viz kapitola 3.3.

- Konkrétně využijeme toho, že zprávy přenášené v agresivním režimu jsou nešifrované. Autentizace se uskutečňuje na základě hashe počítaného z PSK a Nonce (náhodné číslo, které se přenáší v podobě čistého textu). Další slabiny,

kterou využijeme je fakt, že na náš inicializační paket odpoví VPN server paketem obsahujícím onen hash. Pokud tedy odposlechneme tuto nezašifrovanou komunikaci, můžeme se pokusit následně pomocí slovníkového útoku či útoku hrubou silou z hashe dostat předsdílený klíč.

- Při tomto útoku využijeme dva následující programy:

- Program IkeProbe (viz lit. [27]), který postupně zkouší různé kombinace šifrovacích metod, hashovacích algoritmů, různé skupiny Diffie-Hellmanova algoritmu pro zjištění, zda je daný VPN server zranitelný.

- Dalším programem je Cain & Abel (viz lit. [23]). Ten obsahuje velmi mnoho nejrůznějších funkcí zaměřených na získání citlivých informací. My tento program využijeme pro odchycení paketu obsahujícího hash z PSK. Hash následně pomocí útoku hrubou silou v tomto programu prolomíme.

- Pro zjednodušení nastavení nám jako VPN serveru poslouží SW VPN klient HARemote. V našem příkladu nehraje roli, zda použijeme jako VPN server SW klienta či HW bránu, pakety budou mít v obou případech stejnou strukturu. Jen je nutné, aby VPN server podporoval agresivní režim a autentizaci pomocí PSK.

Obr. 32: Schéma zapojení pro útok na protokol IKE

- Na straně serveru nastavíme v SW VPN klientu Aggressive mode, hashovací algoritmus zvolíme MD5 a způsob autentizace PSK. Abychom jako útočníci

prolomili PSK rychle, zvolíme PSK jako číselnou hodnotu *12345678*. Na straně útočníka zvolíme stejné parametry připojení. Hodnotu PSK jako útočník neznáme, proto vyplníme hodnotu náhodnou.

- Abychom zjistili, zda je server zranitelný vůči našemu útoku a nesnažili se tak marně, spustíme na našem stroji aplikaci IkeProbe a jako parametr uvedeme IP adresu serveru. Výstup aplikace by měl končit takto:

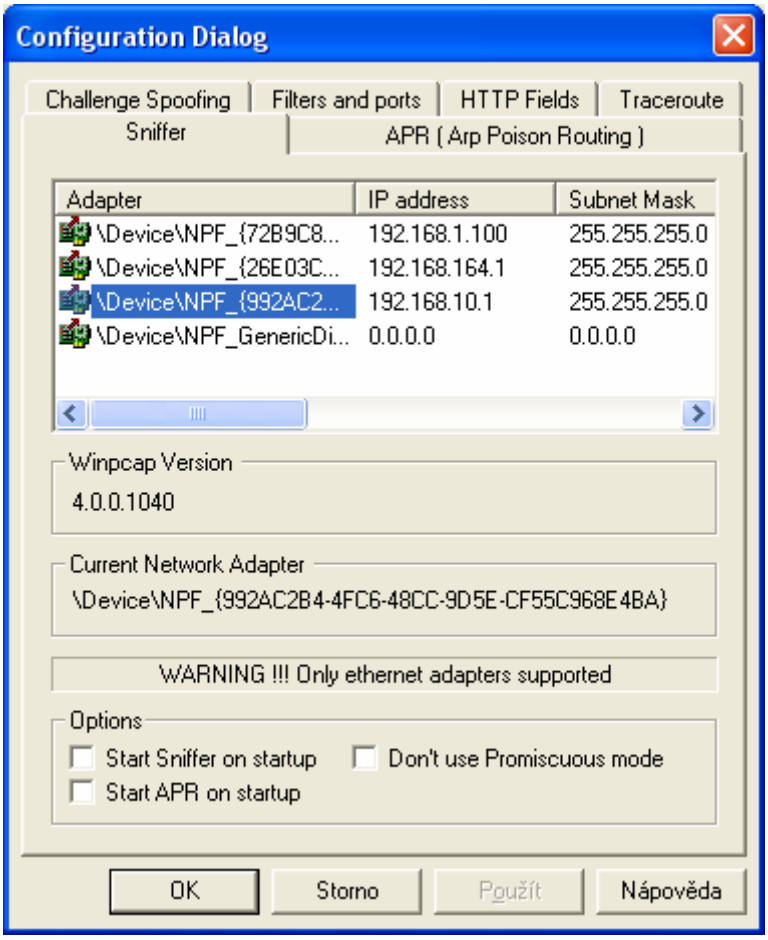
```
.
.
.
Attribute Settings:
Cipher 3DES
Hash MD5
Diffie Hellman Group 2

80.516 3: ph1_initiated(00443ee0, 009b23c8)
80.547 3: << ph1 (00443ee0, 276)
80.906 3: >> 372
80.922 3: ph1_get_psk(00443ee0)

************************************************************************
**********
* System is vulnerable!!
************************************************************************
**********
```

- Tímto jsme zjistili, že je server vůči našemu útoku zranitelný.

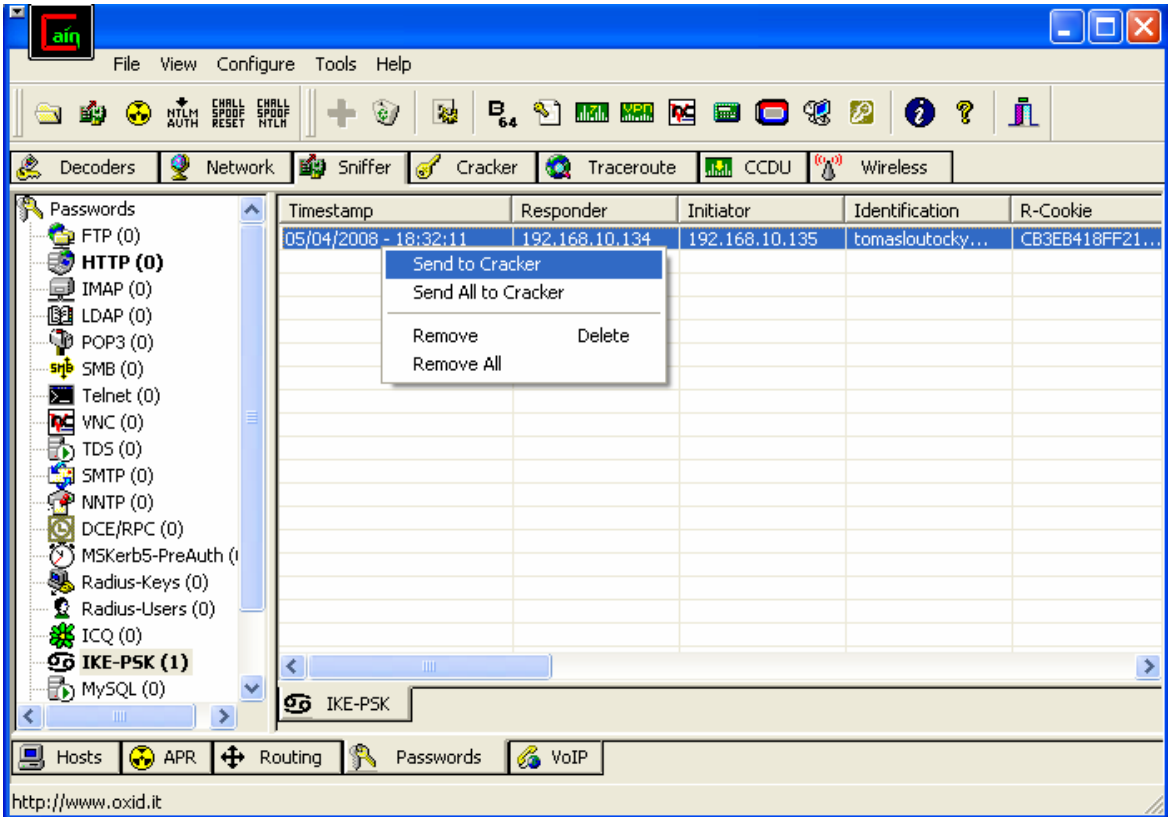
- Spustíme aplikaci Cain & Abel. V menu *Configure* vybereme rozhraní, na kterém chceme odchytávat potřebná data.



Obr. 33: Výběr rozhraní pro odchytávání dat

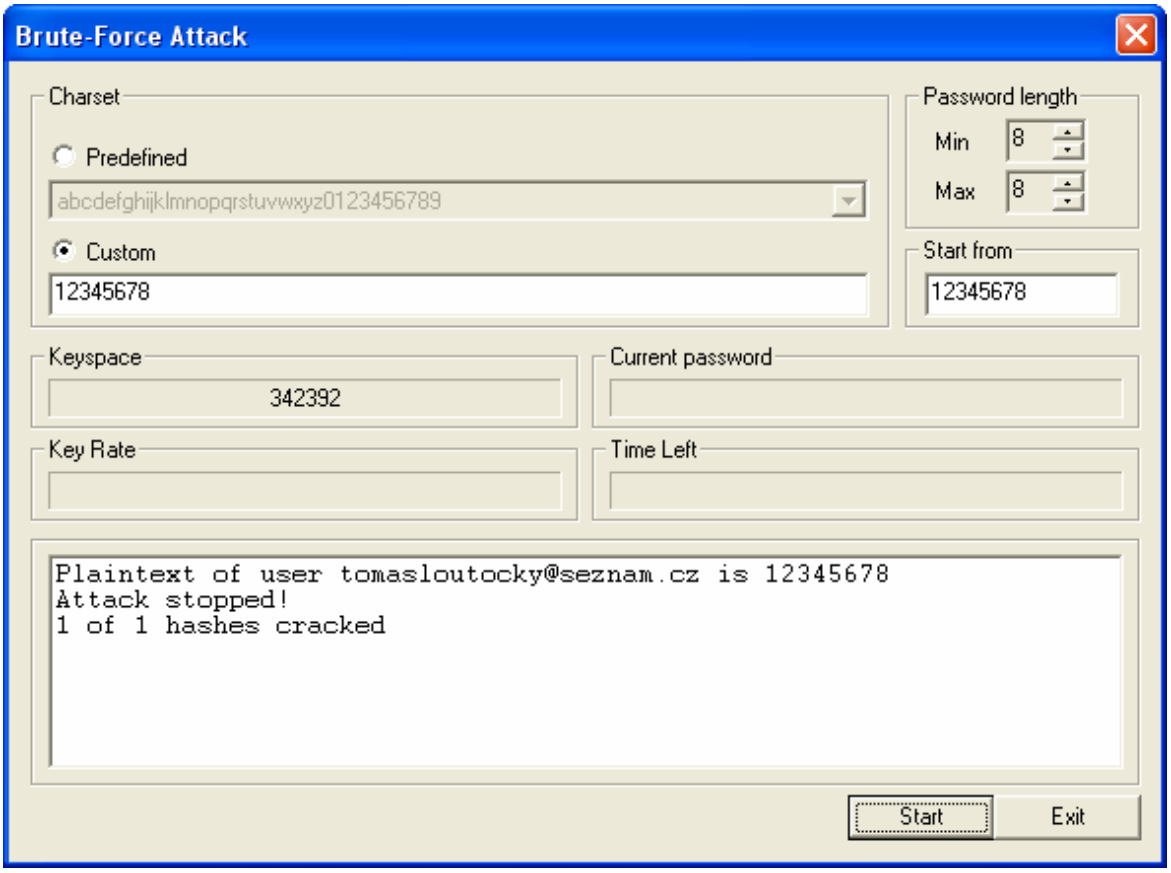
- Na hlavní obrazovce spustíme odchytávání dat druhou ikonou zleva (*Start/Stop Sniffer*).

- Nyní se pokusíme jako útočník připojit k serveru pomocí SW VPN klienta. Připojení se pochopitelně nezdaří, jelikož neznáme správné PSK. Podaří se nám ale v programu Cain & Abel zachytit zhashovanou hodnotu PSK, kterou nám server během pokusu o sestavení spojení zaslal. Na zachycená zhashovaná data klikneme pravým tlačítkem myši a zvolíme *Send to Cracker* (viz obr. 34).



Obr. 34: Zachycení zhashované hodnoty PSK

- Nyní můžeme zastavit zachytávání dat a ze sekce *Sniffer* se přesuneme do sekce *Cracker*, která slouží k prolomení zachycených hesel a jiných citlivých informací. V této sekci si v levém sloupci vybereme oddíl *IKE-PSK Hashes*. V pravém okně vidíme námi zachycený zhashovaný PSK. Klikneme na něj pravým tlačítkem a zvolíme *Brute-Force Attack*, což znamená útok hrubou silou, kdy se zkoušejí všechny možné varianty. Pro urychlení celého procesu prolomení hashe zvolíme, ať program zkouší pouze numerické znaky a délku hesla zvolíme 8 znaků (viz obr. 35). V praxi by útočník neznal délku ani typ znaků použitých k tvorbě hesla, proto by musel vyzkoušet všechny varianty, což je časově mnohem náročnější. V tomto případě útočníci nejdříve zkoušejí slovníkový útok, při kterém jsou testována často používaná hesla (úspěšnost zejména závisí na velikosti slovníku).



Obr. 35: Útok brutální silou

- Po krátké době nám Cracker daný PSK prolomí a zobrazí jej. Teď nám jako útočníkům již nic nebrání k připojení přes VPN k serveru.

- Na závěr je důležité podotknout, že je mnohem bezpečnější používat autentizaci pomocí digitálních certifikátů. PSK autentizaci je vhodné používat například jen při ladění či testování. V praxi díky tomu, že PSK autentizaci je mnohem snadnější nakonfigurovat, není výjimkou se setkat se servery, které PSK autentizaci umožňují. Navíc má řada z nich nastaven tzv. dynamický režim, který dle potřeb klienta umožňuje nastavit jak hlavní tak agresivní režim, čímž umožňuje provést útok, který jsme si předvedli.

5.10 Podvržený certifikát vydaný podvrženou CA

- V této kapitole bude ukázán a prakticky vysvětlen pokus o autentizaci do sítě VPN pomocí podvrženého certifikátu. Pro vytvoření onoho podvrženého certifikátu použijeme prostředí OpenSSL (viz lit. [9]).

- OpenSSL je prostředí obsahující funkce, které implementují protokoly SSL a TLS, dále také implementuje různé kryptografické algoritmy a standardy. Nejčastěji a nejvíce používaným systémem šifrování s veřejným digitálním klíčem je pro SSL kryptografický systém RSA. OpenSSL je projekt open source, na kterém se podílí komunita dobrovolníků po celém světě.

- Ze stránky http://www.slproweb.com/products/Win32OpenSSL.html si stáhneme balíček s posledním vydáním OpenSSL (Win32 OpenSSL v0.9.8g Light), který následně nainstalujeme.

- V adresáři, kam jsme OpenSSL nainstalovali, se nachází adresář *bin*. V tomto adresáři je umístěn jak samotný program, tak konfigurační soubor pro OpenSSL. V adresáři *PEM* se nachází adresář *demoCA* certifikační autority. Tento adresář přesuneme z adresáře *PEM* o adresář výše, tedy do adresáře *bin* (z důvodu nesprávně uvedených cest v konfiguračním souboru). V adresáři *demoCA* nám chybí adresář *newcerts*, kam se budou ukládat nově vydané certifikáty. Adresář proto vytvoříme.

- Chceme, aby certifikát, který posléze vytvoříme, vypadal důvěryhodně - nejlépe tak, že pole certifikátu *Vystavitel* a *Předmět* budou shodná s pravým certifikátem, který hodláme padělat. Otevřeme (dvojklikem) soubor s certifikátem *klient_1_true.cer*. V sekci *Podrobnosti* se podíváme na parametry polí *Vystavitel* a *Předmět*. Zjistíme, že vystavitel (CA, která vydala daný certifikát) má hodnoty:

    - CN = MS LUNA
    - DC = safenet

- Dále zjistíme, že uživatel, kterému byl certifikát vydán, je identifikován parametry (pole *Předmět*):

    - CN = klient_1
    - OU = UTKO
    - O = VUT
    - S = CZ
    - C = CZ

- Protože OpenSSL standardně nenabízí všechny tyto parametry při vytváření certifikátů, je potřeba editovat konfigurační soubor *openssl.cnf*. Obsah direktivy *[req_distinguished_name]* vymažeme a vložíme do ní následující parametry:

    - C = Country
    - ST = State or Province
    - O = Organization Name
    - OU = Organizational Unit Name
    - CN = Common Name
    - DC = DomainController

- Dále je potřeba změnit direktivu *[policy_match]*. Ta nám říká, která pole požadavku na certifikátu se musí shodovat s certifikátem CA. Proto změníme hodnoty všech parametrů v této direktivě na hodnotu *optional*. Výchozí hodnotu (*supplied*) ponecháme pouze u parametru *commonName*.

- Nyní vytvoříme certifikát pro naši podvrženou CA. To provedeme tak, že v příkazové řádce (pozn. musíme být adresáři *OpenSSL/bin*) zadáme příkaz:

openssl req -config openssl.cnf -new -x509 -out demoCA/cacert.pem -keyout demoCA/private/cakey.pem

- Při vytváření certifikátu zadáme heslo, které bude chránit privátní klíč CA. Dále zadáme pouze ty hodnoty, které má i originální CA (tedy v našem případě pouze *Common Name = MS LUNA* a *DomainController = safenet*). Tímto

jsme vytvořili kořenový certifikát naší CA, který je podepsán sám sebou (tzv. Self-signed).

- Pomocí následujícího příkazu vytvoříme žádost o falešný certifikát. Při vytváření zadáme pouze ty parametry, které má pravý certifikát.

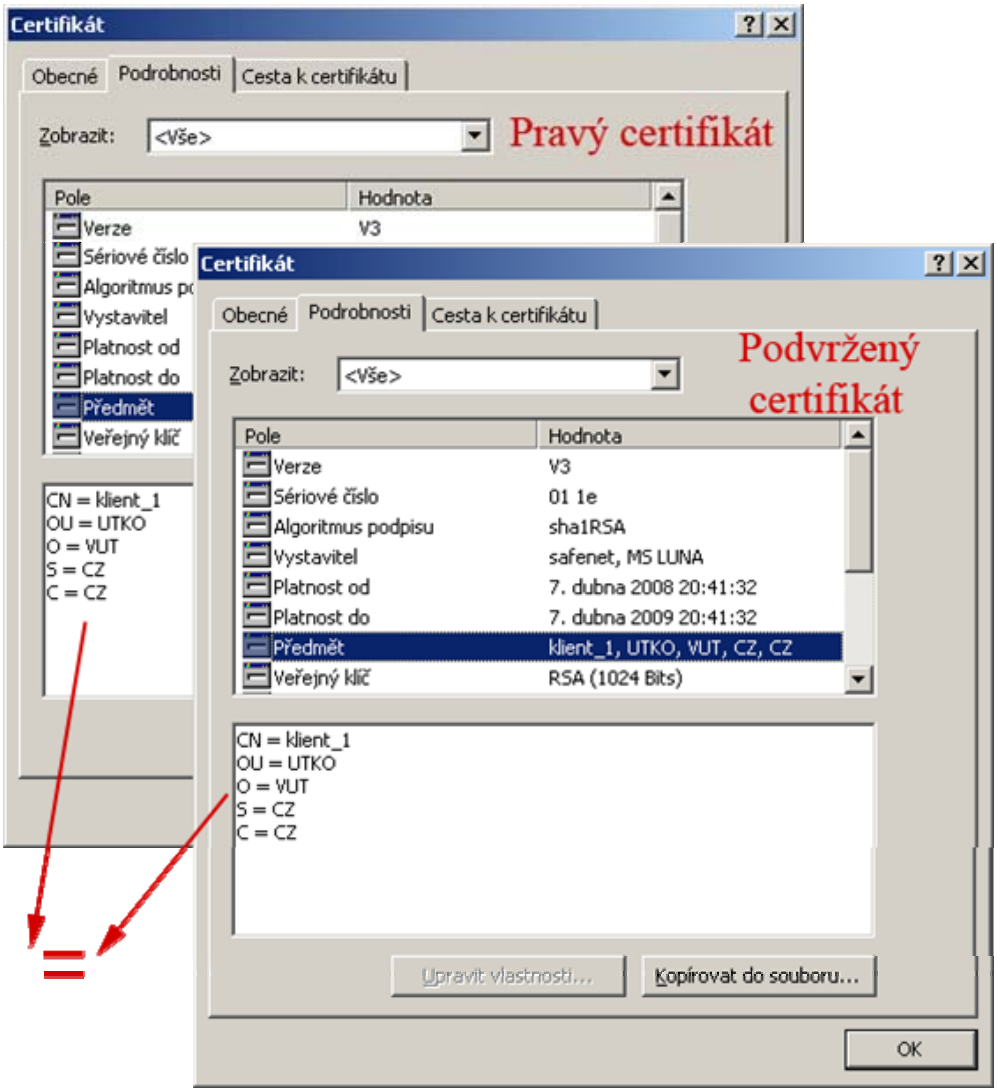
openssl req -config openssl.cnf -new -nodes -out klient_1_fake.pem -keyout klient_1_fake_key.pem

- Po tomto kroku byly vytvořeny dva soubory. V jednom je uložen privátní klíč a ve druhém žádost o certifikát. Tuto žádost nyní potřebujeme podepsat od naší CA. K tomu slouží tento příkaz:

openssl ca -config openssl.cnf -in klient_1_fake.pem -out klient_1_fake.cer

- Při této operaci podepisuje CA žádost svým privátním klíčem, proto budeme vyzvání k zadání hesla, které onen klíč chrání. Heslo bylo vytvořeno při generování kořenového certifikátu CA.

- V souboru *klient_1_fake.cer* máme nyní vytvořen náš falešný certifikát. Můžeme jej porovnat s pravým certifikátem. Na první pohled vypadají totožně.



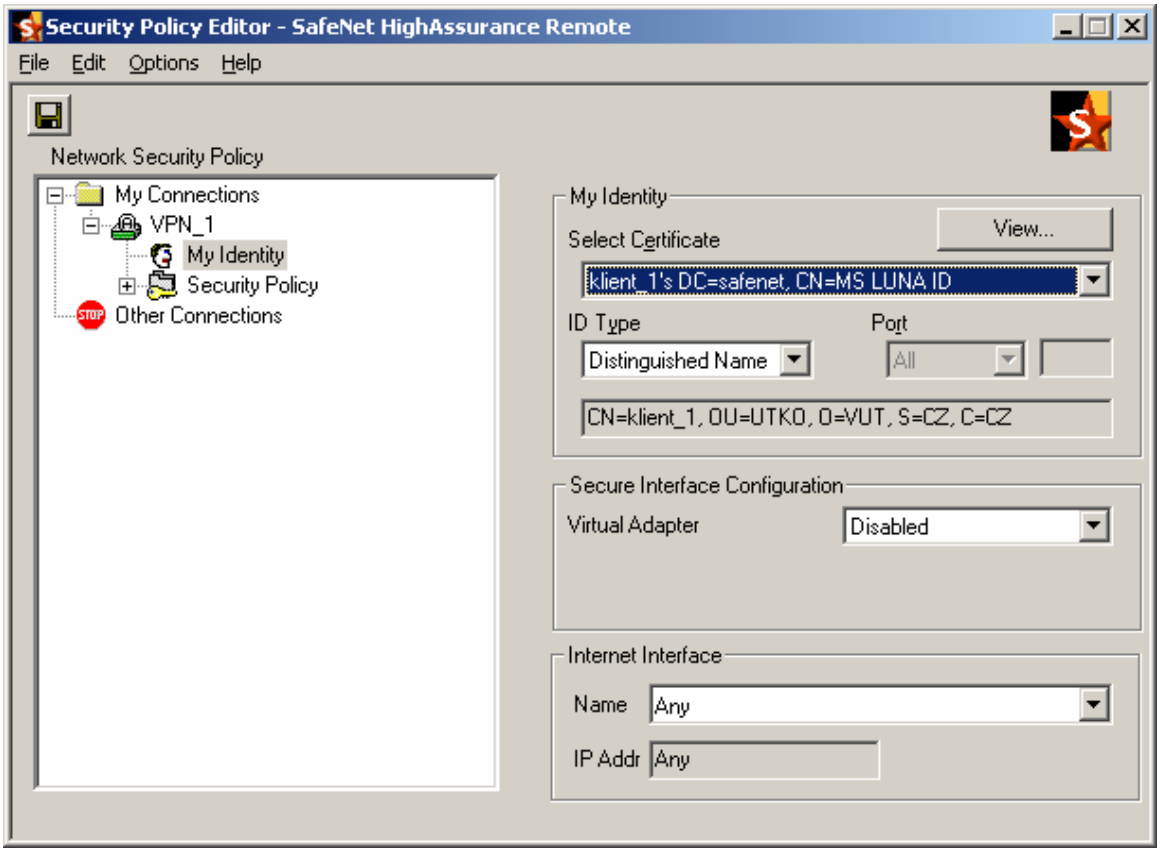
Obr. 36: Pravý a podvržený certifikát

- Pro import do *Certificate manageru* v klientu HARemote musíme náš certifikát převést na formát PKCS#12 (soukromý klíč i certifikát budou v uloženy v jednom souboru). Převod na tento formát provedeme následujícím příkazem:

openssl pkcs12 -export -in klient_1_fake.cer -inkey klient_1_fake_key.pem -out klient_fake_1.p12

- Nyní tento certifikát importujeme do VPN klienta HARemote. Spustíme *Certificate manager* a v sekci *My Certificates* zvolíme *Import Certificate*. Zvolíme typ PKCS#12. Vybereme náš podvržený certifikát a do pole *Password* vložíme heslo, které jsme zadali pří převodu certifikátu na formát PKCS#12.

- V *Security Policy Editoru* zvolíme autentizaci pomocí onoho certifikátu.



Obr. 37: Výběr autentizace pomocí certifikátu

- Aby náš klient důvěřoval certifikátu protější strany, je potřeba naimportovat do našeho *Certificate Manageru* kromě podvržené CA i CA pravou. Certifikáty nejsou tajné (jen privátní klíče), proto není problém je získat. Podobně jako jsme importovali uživatelský certifikát v předchozím kroku, naimportujeme kořenové certifikáty obou CA. Certifikát naší podvržené CA je uložen v adresáři *demoCA* v OpenSSL jako *cacert.pem*. Příponu před importem jen změníme na *.cer*.

- Zkusíme se připojit. Zjistíme, že připojení se nepodařilo. Proč? Pro odpověď se podíváme do logu (*Start → HarAssurance Remote → Log Viewer*).

- Nejdůležitější řádky jsou tyto:

VPN_1 - Using configured user certificate "klient_1's DC=safenet, CN=MS LUNA ID".

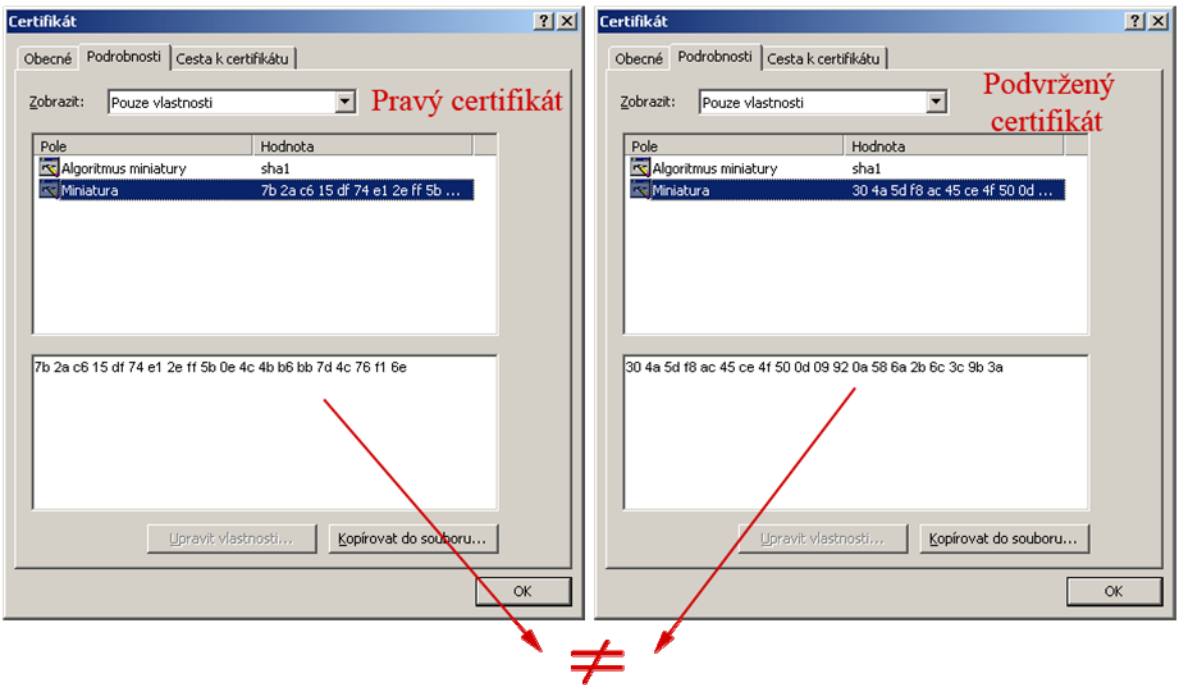
*VPN_1 - SENDING>>>> ISAKMP OAK MM *(ID, CERT, CERT_REQ 5x, SIG, NOTIFY:STATUS_REPLAY_STATUS, NOTIFY:STATUS_INITIAL_CONTACT)*

*VPN_1 - RECEIVED<<< ISAKMP OAK INFO *(HASH, NOTIFY:**INVALID_CERT**)*

VPN_1 - Discarding SA negotiation

- Autentizace byla neúspěšná, protože strana odmítla podvržený certifikát.
- Vysvětlení je následující. Certifikát kromě identifikačních údajů o subjektu, kterému byl certifikát vydán, obsahuje také hash certifikátu, který je podepsán privátním klíčem CA, která certifikát vydala. Ověření certifikátu probíhá tak, že druhá strana komunikace přijme certifikát a zjistí hash pomocí veřejného klíče CA, kterou certifikát udává jako vydavatele. Veřejný klíč CA získá uživatel z kořenového certifikátu této CA, který má uživatel uložen mezi důvěryhodnými CA ve svém úložišti. Zároveň si uživatel sám vypočítá hash z certifikátu. Pokud se oba hashe rovnají, můžeme certifikát považovat za důvěryhodný. V našem případě tedy pošleme protější straně podvržený certifikát, který máme vydaný podvrženou CA. Protější strana komunikace vypočte hash z přijatého certifikátu. Následně zjistí hash uvedený v certifikátu pomocí veřejného klíče uvedené CA MS LUNA. Veřejný klíč získá z kořenového certifikátu CA MS LUNA, který má uložen ve svém úložišti. Protože má ale ve svém úložišti uloženu pravou CA MS LUNA, která má oproti podvržené MS LUNA jiný veřejný klíč, vypočítá i jiný hash. Ten nebude stejný jako hash, který si protější strana vypočítá a tak autentizace proběhne neúspěšně.
- Úspěšnou autentizaci pomocí podvrženého certifikátu by bylo možné provést uložením kořenového certifikátu podvržené CA do úložiště mezi důvěryhodné CA na protější straně komunikace. Například pomocí nějakého viru.

- V případě našeho pokusu byla úroveň bezpečnosti vysoká díky tomu, že protější straně nebyla nabídnuta možnost, zda našemu podvrženému certifikátu věřit či ne. Certifikát byl automaticky zamítnut. To ale neplatí u všech aplikací, kde se certifikáty využívají. Pokud například vytvoříme kopii zabezpečeného serveru (např. banky) a použijeme pro autentizaci serveru podvržený certifikát, tak uživatel, který na tento server přistoupí, bude webovým prohlížečem upozorněn, že se může jednat o podvržený certifikát. Nicméně mu prohlížeč umožní v komunikaci pokračovat. Většina uživatelů po zběžném zkontrolování certifikátu (kdo ho vydal, komu byl vydán) v komunikaci pokračuje. A jak jsme si ukázali, podvrhnout certifikát není nic těžkého. Obrana spočívá v kontrole správnosti hashe certifikátu (nejlépe s hashem, který jsme získali bezpečnou cestou - např. osobním kontaktem).



Obr. 38: Hash u pravého a podvrženého certifikátu

5.11 Útok DoS

- V této části si za využití bezpečnostní slabiny v aplikaci vyzkoušíme útok Denial of Service (DoS).

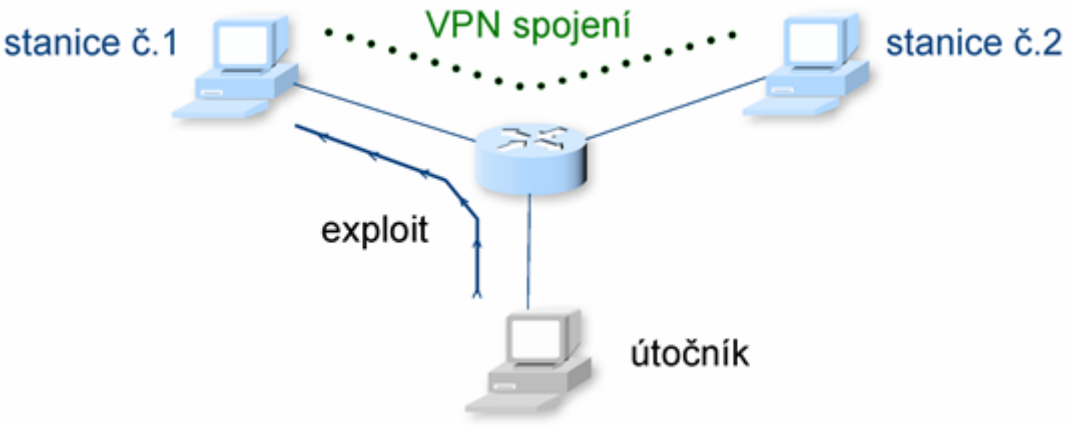
- DoS neboli odepření služby je způsob útoku, při kterém daná konkrétní služba nebude dostupná pro ostatní uživatele. Většinou se pro tento typ útoku využívá bezpečnostní slabina dané služby a nebo také zahlcení oné služby požadavky, tak že služba nezvládá zpracovávat požadavky oprávněných uživatelů. Druhý typ je většinou realizován z více míst, kdy útočník ovládá více stanic. V tomto případě se mluví o distribuovaném odepření služby (DDoS - Distributed Denial of Service).

- Často se při DoS útoku využívá exploitů. Exploit je malá jednoúčelová aplikace využívající bezpečnostní slabiny dané služby. Na internetu je možné najít řadu databází těchto exploitů (např. http://www.milw0rm.com) (viz lit. [22]). Jsou často psány v programovacím jazyku C.

- V našem případě využijeme bezpečnostní slabiny v SW VPN klientu SafeNet HighAssurance Remote. Poměrně jednoduchý exploit pošle SW VPN klientu neplatný paket. SW VPN klient místo toho, aby tento paket zahodil, způsobí nejen nedostupnost aplikace HighAssurance Remote, přerušení probíhajícího VPN spojení, ale také zamrznutí celého systému. Vše proběhne velmi rychle, aniž by se cokoliv stihlo zapsat do logovacího souboru aplikace či operačního systému.

- Daný exploit si vyhledáme na základě klíčových slov *High Assurance Remote* na výše zmíněné adrese http://www.milw0rm.com. Je psaný v jazyku C a je nutné ho nejdříve zkompilovat. K tomu potřebujeme sadu kompilátorů a překladačů. Ty jsou většinou již součástí systému (např. překladač GCC). Dále potřebujeme kolekci knihoven Libnet, které umožňují pokročilou práci s pakety. Pro ušetření času si již zkompilovaný exploit stáhneme z adresy http://upload.loutocky.com/expl/a.out .

- Na obr. 39 je zobrazeno zapojení stanic a jejich rolí při uskutečnění útoku. Mezi stanicemi č.1 a č.2 vytvoříme VPN spojení mezi dvěma SW VPN klienty. Stanice č.2 může být i HW VPN brána, na konečný výsledek by to nemělo vliv. HW VPN bránu je ale časově náročnější nakonfigurovat a proto si vystačíme se SW klientem.



Obr. 39: Zapojení pro uskutečnění útoku DoS

- Na stanici útočníka (OS Linux) spustíme daný exploit. Pro spouštění je nutné mít root práva. Do spouštěcích parametrů uvedeme cílovou adresu (IP stanice č.1) a dále libovolnou zdrojovou IP adresu. Na stanici č.1 je nutné mít vypnutý firewall.

- Jakmile exploit takto spustíme, zjistíme že na stanici č.2 došlo k výpadku VPN probíhajícího VPN spojení se stanicí č.1. Také zjistíme, že stanice zamrzla a je nutné ji restartovat.

- V tomto cvičení jsme se naučili co jsou exploity, jak je jednoduché je získat a jak je jednoduché je zneužít. Je ale důležité vědět, jak se proti nim bránit.

- Hlavní zásadou je mít nainstalovanou poslední aktuální verzi daného programu, která nalezené bezpečnostní slabiny opravuje. Realita je ale taková, že ne vždy tvůrci aplikací dokážou včas vytvářet a distribuovat opravené verze programů

či bezpečnostní aktualizace. Stejně tak administrátoři často neaktualizují aplikace včas. Útočníci tak mohou delší dobu zneužívat bezpečnostních slabin aplikací či systémů.

5.12 Uvedení do původního stavu

- SMC uvedeme do původního stavu vymazáním veškerých VPN politik, které jsme vytvořili. Dále vymažeme objekty, které jsme vytvořili v doménách jednotlivých bran a následně z root mapy i samotné VPN brány. Jako poslední v SMC vymažeme vytvořené VPN klienty.

- VPN brány HA500 uvedeme do původního nastavení přes CLI. V privilegovaném módu napíšeme příkaz *factory-default*. Tím se vymaže celý konfigurační soubor brány a s ním i certifikáty a CRL seznamy uložené v bráně.

6. Závěr

První tři kapitoly této diplomové práce tvoří její teoretickou část. V první z nich je popsán význam základních pojmů počítačové bezpečnosti, z nichž některé jsou následně zmíněny v dalších kapitolách. Ve druhé kapitole je vysvětlena technologie VPN, která je v současné době stále žádanější. Čtenář by tak měl získat dostatek informací, aby byl obeznámen s tím, jak tato technologie funguje. V rámci této kapitoly jsou charakterizovány jednotlivé druhy VPN a to dle několika aspektů. Třetí kapitola je věnována konkrétněji řešení VPN pomocí protokolu IPSec. Zde je vysvětlena jeho funkce, popsány dílčí protokoly a na závěr zhodnoceny klady a zápory tohoto protokolu.

V praktické části tvořené dvěma kapitolami je představena možná implementace tohoto protokolu s využitím autentizačních kryptografických předmětů. Pro tento účel byly vybrány produkty firmy Safenet, která se na tuto oblast zaměřuje a patří mezi špičku v oboru. Čtenář je v této části seznámen s jednotlivými prvky, které se v testovací laboratoři vyskytují, a je mu vysvětlena jejich funkce. Měl by tak získat představu, jakou roli hraje VPN ve spojení s hardwarovými tokeny v dnešní době při zabezpečení počítačových sítí.

Řešení navrhnuté v kapitole 4.1 je možné například použít pro společnost, která má několik poboček a externích pracovníků (tzv. teleworkerů), kteří potřebují bezpečně a vzdáleně komunikovat s ústředím společnosti, přistupovat k vnitřním zdrojům společnosti. Dané řešení vyniká zejména ve vysokém stupni zabezpečení. Produkty používané v této síti jsou certifikované normami FIPS a garantují tak určitou úroveň bezpečnosti. K vysoké bezpečnosti ovšem v praxi nestačí mít specializovaný HW a SW. Je třeba aby technologie byly správně implementovány a zejména bezpečně používány. V tomto případě je tím například myšleno používání autentizace pomocí digitálních certifikátů namísto PSK. Je také vhodné ukládat digitální certifikáty do bezpečných úložišť (např. do tokenů nebo čipových karet), nepoužívat slabé šifrovací a hashovací algoritmy, nepoužívat agresivní režim při navazování spojení a udržovat všechen SW aktualizovaný. Následky nedodržení některých těchto pravidel jsou prakticky ukázány v kap. 5. Uvedené řešení má i několik nevýhod. Jednou z největších je cena, která je vysoká zejména díky normám FIPS, jimiž jsou produkty v síti certifikovány. Další nevýhodou je nedostatek kvalitní dokumentace a s

tím spojená obtížnější konfigurace a ladění chyb. V tomto ohledu vedou open source produkty, které mají dostatek dokumentace a navíc silnou podporu komunity na internetu. U takto specializovaných produktů se většinou počítá s podporou od dodavatele, která je ale placená. Nedostatečná podpora Linuxových OS je další nevýhodou. Většina produktů Linux nepodporuje vůbec. Produkty, které podporu Linuxu mají, podporují jen stará jádra, na která již nejsou vydávány bezpečnostní záplaty.

V kap. 5 byly také vypracovány modulární praktické postupy. V těchto postupech si studenti vyzkouší konfiguraci jednotlivých zařízení v síti a také jejich funkci. Jsou zde také uvedeny postupy, které jsou zaměřeny na bezpečnost. V těch se studenti seznámí s bezpečnostními slabinami a možnou ochranou vůči nim. Studenti si v rámci uvedených postupů mohou vyzkoušet například ověření CRL, analýzu jednotlivých módů protokolu IKE, mohou si také vytvořit podvržený certifikát a pokusit se jej zneužít v dané síti. Dále si vyzkouší útok využívající bezpečnostní slabinu v protokolu IKE nebo útok typu DoS.

7. Použitá literatura

[1] BAY, Robin. *Čipové karty a USB tokeny, aneb bezpečnější autentizace a šifrování* [online]. 2003 [cit. 2007-10-21]. Dostupný z WWW: <http://www.svetsiti.cz/view_list.asp?rubrika=Tutorialy&temaID=264>.

[2] DANSEGLIO, Mike. *Securing Windows Server 2003*. [s.l.] : O'Reilly, 2004. 456 s. ISBN 0-596-00685-3.

[3] DE CLERQ , Jan. *Windows Server 2003 Security Infrastructures*. [s.l.] : Digital Press, 2004. 756 s. ISBN 1555582834.

[4] DOSTÁLEK, Libor. *Velký průvodce protokoly TCP/IP Bezpečnost. 2*. aktualiz. vyd. Praha : Computer Press, 2003. 571 s. ISBN 80-7226-849-X.

[5] Federal Information Processing Standard. *Wikipedia* [online]. 2008 [cit. 2007-12-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Federal_Information_Processing_Standard>.

[6] FIPS 140. *Wikipedia* [online]. 2008 [cit. 2007-12-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/FIPS_140>.

[7] HARKINS, Dan, CARREL, David. The Internet Key Exchange (IKE). *Request for Comments* [online]. 1998 [cit. 2008-03-12]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc2409.txt>.

[8] HENMI, Anne, et al. *Firewall Policies and VPN Configurations*. [s.l.] : Syngress Publishing,Inc., 2006. 504 s. ISBN 1-59749-088-1.

[9] HRUSKA, Thomas. Shining Light Productions : *Win32 OpenSSL* [online]. 2008. 2008 [cit. 2008-04-16]. Text v angličtině. Dostupný z WWW: <http://www.slproweb.com/products/Win32OpenSSL.html>.

[10] IPSec vs. SSL VPNs for Secure Remote Acces. Bitpipe.com [online]. 2006 [cit. 2008-03-08]. Dostupný z WWW: <http://www.bitpipe.com/detail/RES/116535203_903.html>.

[11] *Jak funguje IPSEC* [online]. c2007 [cit. 2007-10-20]. Dostupný z WWW: <http://www.security-portal.cz/clanky/jak-funguje-ipsec-.html>.

[12] KENT, S. IP Authentication Header. *Request for Comments* [online]. 2005 [cit. 2008-03-16]. Dostupný z WWW: <http://tools.ietf.org/html/rfc4302>.

[13] KENT, S. IP Encapsulating Security Payload (ESP). *Request for Comments* [online]. 2005 [cit. 2008-03-16]. Dostupný z WWW: <http://tools.ietf.org/html/rfc4303>.

[14] KENT, S., ATKINSON, R. Security Architecture for the Internet Protocol. *Request for Comments* [online]. 1998 [cit. 2008-03-12]. Dostupný z WWW: <http://tools.ietf.org/html/rfc2401>.

[15] KIVINEN, Tero, et al. *Negotiation of NAT-Traversal in the IKE* [online]. 2005 [cit. 2007-11-02]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc3947.txt>.

[16] Layer 2 Tunneling Protocol. *Wikipedia* [online]. 2008 [cit. 2008-12-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Layer_2_Tunneling_Protocol>.

[17] LEWIS, Mark. *Comparing, Designing, and Deploying VPNs*. [s.l.] : Cisco Press, 2006. 1080 s. ISBN 1-58705-179-6.

[18] LOUTOCKÝ, Tomáš. *Bezpečnostní rizika internetovských prohlížečů*. [s.l.], 2006. 48 s. Vedoucí bakalářské práce Doc. Ing. Václav Zeman, Ph.D.

[19] LUHOVÝ, Karel. *Virtuální privátní sítě VPN* [online]. 2003 [cit. 2007-10-28]. Dostupný z WWW: <http://www.svetsiti.cz/view_list.asp?rubrika=Tutorialy& temaID=219>.

[20] MAUGHAN, Douglas, et al. Internet Security Association and Key Management Protocol (ISAKMP). *Request for Comments* [online]. 1998 [cit. 2008-03-12]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc2408.txt>.

[21] MEHURON, William, et al. FIPS PUB 186-2. *National Institute of Standards and Technology* [online]. 2000 [cit. 2007-11-12]. Dostupný z WWW: <http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>.

[22] *Milw0rm* [online]. 2008 [cit. 2008-04-18]. Text v angličtině. Dostupný z WWW: <http://www.milw0rm.com>.

[23] MONTORO, Massimiliano. *Cain & Abel* [online]. 2008. 2008 [cit. 2008-04-16]. Text v angličtině. Dostupný z WWW: <http://www.oxid.it/cain.html>.

[24] PLOTNICK, Neil. *Software vs hardware VPN* [online]. c2007 [cit. 2007-10-10]. Dostupný z WWW: <http://vpn.sockslist.net/cgi-bin/vpn/virtual_private_network_vpn_hardware_vs_software.html?do=hardvssoft>.

[25] Point-to-point tunneling protocol. *Wikipedia* [online]. 2008 [cit. 2008-12-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Point-to-point_tunneling_protocol>.

[26] STALLINGS, William. *Cryptography and Network Security Principles and Practices. 4th edition*. [s.l.] : Prentice Hall, 2005. 592 s. ISBN 0-13-187316-4.

[27] THUMANN, Michael . Heise : *IKEProbe* [online]. 2004 [cit. 2008-04-18]. Text v němčině. Dostupný z WWW: <http://www.heise.de/security/tools/default.shtml?prg=57>.

[28] THUMANN, Michael, REY, Enno. PSK Cracking using IKE Aggressive Mode. *ERNW* [online]. 2004 [cit. 2008-04-18]. Dostupný z WWW: <http://www.ernw.de/content/e6/e179/e608/download611/pskattack_ger.pdf>.

[29] *Wireshark* [online]. 2008. Gerald Combs, 2008 [cit. 2008-04-22]. Text v angličtině. Dostupný z WWW: <http://www.wireshark.org/>.