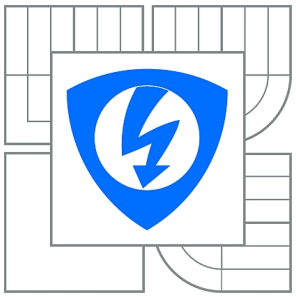
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

KRYPTOGRAFICKÉ PROTOKOLY VYUŽÍVANÉ V POČÍTAČOVÝCH SÍTÍCH

CRYPTOGRAPHIC PROTOCOLS USED IN COMPUTER NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

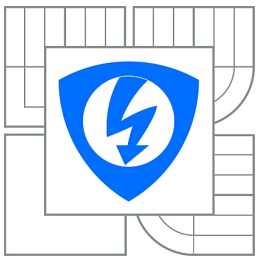
AUTOR PRÁCE — MICHAL LÚDIK

AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK DOLEŽEL

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Michal Lúdik	***ID:***	106599
Ročník:	3	***Akademický rok:***	2009/2010

NÁZEV TÉMATU:

Kryptografické protokoly využívané v počítačových sítích

POKYNY PRO VYPRACOVÁNÍ:

Proveďte rozbor základních kryptografických protokolů používaných pro zabezpečenou komunikaci v počítačových sítích. Prostudujte princip autentizace a autorizace jednotlivých kryptografických protokolů. Porovnejte popisované kryptografické protokoly z pohledu referenčního modelu ISO/OSI a zjistěte možné nebezpečí. Navrhněte model pro ověření funkce, bezpečnosti a měření přenosových a výkonových parametrů kryptografických protokolů. Navržený model prakticky realizujte v laboratorním prostředí. Využívejte nástroje z oblasti Open Source Software.

DOPORUČENÁ LITERATURA:

[1] Boyd, C. Protocols for authentication and key establishmen. Berlin: Springer-Verlag, 2003. 321s. ISBN 3-540-43107-1.
[2] Dostálek, L. Velký průvodce protokoly TCP/IP: bezpečnost. Praha: Computer Press, 2001. 565s. ISBN 80-7226-513-X.

Termín zadání:	29.1.2010	***Termín odevzdání:***	2.6.2010

Vedoucí práce: Ing. Radek Doležel

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Táto práca sa zaoberá rozborom základných kryptografických protokolov, princípom autentizácie a overením ich bezpečnosti. V prvej časti práce je krátke zoznámenie s problematikou – kryptografiou, ďalej niektoré typy útokov, autentizácia a autorizácia. V druhej časti je to rozbor protokolov SSH, SSL, IPsec a Kerberos a overenie ich funkčnosti a bezpečnosti v laboratórnych podmienkach.

KLÍČOVÁ SLOVA

ah, autentizácia, autorizácia, bezpečnosť, esp, https, ike, ipsec, isakmp, kdc, kerberos, kryptografia, ssh, ssl, tls, útoky

ABSTRACT

This work deals with the analysis of basic cryptographic protocols, principle of the authentication and verification of their safety. In the first part of the work is a short introduction to the issue – cryptography, some types of attacks, authentication and authorisation. In the second part is the analysis of protocols SSH, SSL, IPSec, and Kerberos, authentication and verification of their functionality and security under laboratory conditions.

KEYWORDS

ah, authentication, authorisation, security, esp, https, ike, ipsec, isakmp, kdc, kerberos, cryptography, ssh, ssl, tls, attacks

LÚDIK, Michal *Kryptografické protokoly využívané v počítačových sítích*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 50 s. Vedoucí práce byl Ing. Radek Doležel

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Kryptografické protokoly využívané v počítačových sítích“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce 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.

Brno ....

(podpis autora)

POĎAKOVANIE

Ďakujem vedúcemu bakalárskej práce Ing. Radkovi Doleželovi za jeho metodickú podporu, rovnako za poskytnutie laboratórnych podmienok pre riešenie praktickej časti.

V Brne dňa 5. mája 2010 Michal Lúdik

OBSAH

Úvod **11**

1 Kryptografia **12**

2 Autentizácia a autorizácia **14**
2.1 Autentizácia 14
2.2 Autorizácia 15

3 Niektoré typy útokov **16**
3.1 Replay attack 16
3.2 IP spoofing 16
3.3 Dictionaty attack 16
3.4 Denial of Service 16

4 SSH **18**
4.1 SSH server 19
4.2 SSH klient 19
4.3 OpenSSH 19
4.4 Inštalácia ssh serveru 20
4.5 Výsledky merania 21
4.6 Priebeh spojenia 21

5 SSL **23**
5.1 Secure Socket Layer 23
5.2 Transport Layer Security 23
5.3 OpenSSL 24
5.4 Vytvorenie zabezpečeného serveru 24
5.5 Výsledky merania 26
5.6 Priebeh spojenia 26

6 IPsec **28**
6.1 Protokol AH 28
6.2 Protokol ESP 29
6.3 Protokol IKE 29
6.4 ISAKMP 30
6.5 Zabezpečená komunikácia 30
6.6 Výsledky merania 32
6.7 Priebeh spojenia 32

7 Kerberos **33**
7.1 Kerberos ako autentizačný protokol 33
7.2 Vytvorenie zabezpečenej relácie 34
7.3 Priebeh spojenia 36

8 Zhrnutie vlastností protokolov **37**

9 Záver **39**

Literatúra **40**

Zoznam skratiek **43**

Zoznam príloh **45**

A SSL certifikát a jeho detaily **46**

B Kerberos **50**

ZOZNAM OBRÁZKOV

1.1 Predstavenie kryptografie 12
2.1 Prenos zašifrovaného hesla 14
2.2 Metóda výzva-odpoveď 15
4.1 Organizácia SSH 18
4.2 Priebeh komunikácie pomocou SSH 22
5.1 Oznámenie neovereného certifikátu 24
5.2 Priebeh komunikácie pomocou SSL 27
6.1 Paket zabezpečený AH v transportnom a tunelovom móde 29
6.2 ESP paket v transportnom a tunelovom móde 30
6.3 Priebeh komunikácie zabezpečenej IPsecom 32
7.1 Proces dohodnutia tajných kľúčov 33
7.2 Priebeh prihlásenia na ftp server 36
A.1 Certifikát SSL (1) 48
A.2 Certifikát SSL (2) 48
A.3 Informácie o zabezpečenej stránke 49
A.4 Prehľad paketov SSL zachytených wiresharkom 49
B.1 Prehľad paketov zachytených wiresharkom pri autentizácii Kerberom 50

ZOZNAM TABULIEK

1 Prehľad protokolov a im príslušných vrstiev 11
4.1 Prehľad merania zabezpečením SSH 21
5.1 Prehľad merania zabezpečením SSL 26
6.1 Prehľad merania zabezpečením IPsec 32
8.1 Komplexný prehľad vlastností protokolov 38

ÚVOD

Cieľom mojej práce je preštudovať a zoznámiť sa s kryptografickými protokolmi, navrhnúť model pre overenie ich funkčnosti a bezpečnosti a zmerať ich výkonové a prenosové parametre.

Aby klient a server mohli spolu komunikovať ľubovoľným protokolom, musia sa najprv autentizovať jeden druhému. Práve v tejto chvíli je ideálna príležitosť pre útočníka napadnúť komunikáciu a odchytiť pre neho užitočné informácie. Útok na komunikáciu nemusí byť vedený iba za účelom získania dát. Jeho podstatou môže byť aj zabránenie komunikácii dvoch počítačov. Tomuto sa snažia zabrániť kryptografické protokoly, ktoré v jednotlivých kapitolách popisujem.

Na začiatku práce je krátky úvod ku kryptografii – čo slovo kryptografia znamená a jej krátka história. Obsahom nasledujúcich kapitol je vysvetlenie kľúčových bodov – autentizácia a autorizácia – ktoré sú nevyhnutnou súčasťou každej nadviazanej bezpečnej relácie.

Nasledujúca kapitola pojednáva o niektorých typoch útokov na kryptografické protokoly. Jej obsahom je stručne priblížiť ich podstatu.

Zvyšné kapitoly sa venujú už popisu jednotlivých kryptografických protokolov, medzi ktoré patria: Secure Shell, Secure Sockets Layer, Transport Layer Security, Kerberos a IPsec. V tab. 1 je krátky prehľad protokolov a vrstiev OSI modelu, súčasťou ktorých sú.

Svoju bakalársku prácu som zameral na oblasť Open Source Software. Praktickú časť riešim v prostredí linuxu, konkrétne Ubuntu 9.10 na notebooku branom ako klient a Ubuntu 10.04 alfa II na počítači, ktorý slúžil ako server. Systém som aktualizoval najnovšími aktualizáciami. Pracoval som na:

- notebooku – Intel DualCore 1.87 GHz, 3 GB RAM a 100Mbps LAN karta,
- stolnom PC – Intel Core2Duo 2.8 GHz, 1 GB RAM a 1Gbps LAN karta.

Pre zisťovanie využitia CPU som používal východzí program pre sledovanie systému *gnome-system-monitor*, pri vzdialenej práci konzolový príkaz *top* a pre analýzu siete program *Wireshark*. V práci su písané rôzne IP adresy a mená užívateľov, pretože nastavenia som vykonával vzdialene aj z iných ako hore spomínaných počítačov. Meranie výkonových a prenosových vlastností som vykonal iba na spomínaných PC.

Pri praktickom riešení tejto práce v laboratórnych podmienkach nedošlo k ohrozeniu bezpečnosti osôb ani majetku.

Tab. 1: Prehľad protokolov a im príslušných vrstiev

Protokol	SSH	SSL	TLS	Kerberos	IPsec
Vrstva	Aplikačná	Prezentačná	Transportná	Vyššie vrstvy	Sieťová

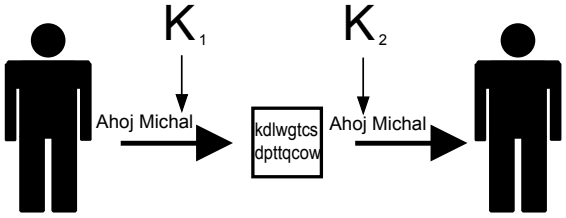
1 KRYPTOGRAFIA

Úlohou kryptografie (z gréčtiny kryptos – skrytý, gráphein – písať) je utajiť obsah správy tak, aby bola čitateľná iba so špeciálnou znalosťou [20].

Korene kryptografie siahajú do dávnej minulosti, kedy si potrebovali dve strany vymeniť informáciu bez toho, aby si ju prečítal niekto iný. Na toto im slúžili rôzne spôsoby utajovania, napr. neviditeľné atramenty a pod. Hlavnou úlohou kryptografie je utajenie obsahu správy, zabrániť jej pozmeneniu a zabrániť tomu, aby sa „Jano“ vydával za „Jozefa“.

A na čo vlastne potrebujeme kryptografiu? Kryptografia je základný prostriedok bezpečnosti nielen na internete. Je založená predovšetkým na matematike (rôzne šifrovacie algoritmy sú komplexné matematické funkcie). Delíme ju na (obr. 1.1) [7]:

- **symetrickú** – odosielateľ aj príjemca používajú jeden tajný kľúč ($K_1 = K_2$),
- **asymetrickú** – potenciálny príjemca správy vlastní súkromný kľúč, ktorý je známy len jemu a odosielateľ verejný kľúč, ktorý je verejne známy a zverejnený ($K_1 \neq K_2$).



Obr. 1.1: Predstavenie kryptografie

S nástupom modernej techniky, hlavne počítačovej, nastal problém s bezpečnosťou. Každý užívateľ by mal dbať na to, aby nejakým spôsobom neohrozil samého seba – svoje citlivé údaje, heslá a pod. Problémom modernej generácie je spôsob získavania dát. Väčšina ľudí je schopná vymeniť svoje dôležité čísla účtov či telefónov pre niečo nezmyselné napr. na warez fórach (diskusné fóra, hlavnou úlohou ktorých je zdieľanie autorsky chránených diel), odkiaľ smeruje najväčšie percento útokov na užívateľa (phishing, vírusy, spyware atď.).

Najnovším príkladom môžu byť sociálne siete napr. Twitter (`http://twitter.com`), Facebook (`http://www.facebook.com`), alebo slovenský Azet (`http://www.azet.sk`) či české Libimseti (`http://libimseti.cz`). Na jednej strane sa všetci snažíme byť superanonymní, ale na strane druhej veľká väčšina ľudí si do svojho profilu uvedie informácie, ktoré sa tam nehodia.

Veľkú výhodu v bezpečnosti má Open Source Software, pretože doň môže prispievať ktokoľvek a preto sa rýchlo vyvíja. Na druhej strane vďaka tomuto systému môže útočník do kódu pridať svoj škodlivý kód a vydávať ho za korektný software.

Preto sa treba rozhodnúť, odkiaľ si program stiahnete. Linuxové distribúcie majú tento problém ošetrený certifikátmi. Rovnako sú aj aktualizácie overované a preto nie je možné ohroziť systém.

2 AUTENTIZÁCIA A AUTORIZÁCIA

Táto kapitola vysvetľuje, čo je to autentizácia a autorizácia, a aké typy autentizácie poznáme.

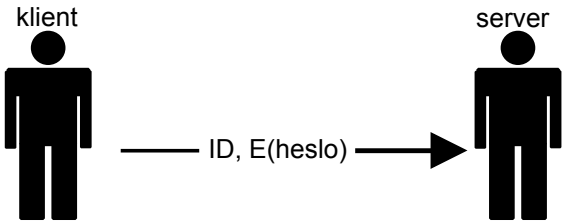
2.1 Autentizácia

Úlohou autentizácie je overenie identity užívateľov (užívateľ je naozaj ten, za ktorého sa vydáva) a autenticity (integrity) dát (dáta pri prenose nikto nezmenil). Stretávame sa s ňou pravidelne na každom kroku.

Poznáme jednostrannú a obojstrannú autentizáciu [2]. Najbežnejším príkladom jednostrannej autentizácie je podpisovanie dokumentov. Našim podpisom takto potvrdzujeme svoju identitu. Ďalším príkladom môže byť zadávanie PIN kódu do mobilného telefónu či bankomatu, zadávanie hesla do operačného systému a pod. Najmodernejšie výdobytky technológie umožňujú autentizovať sa čipovou kartou, odtlačkami prstov, očnou sietnicou, štýlom chôdze, prítlakom na klávesy pri písaní na klávesnici atď. V prípade predaja nehnuteľnosti je potrebná obojstranná autentizácia – podpis predajcu a kupca.

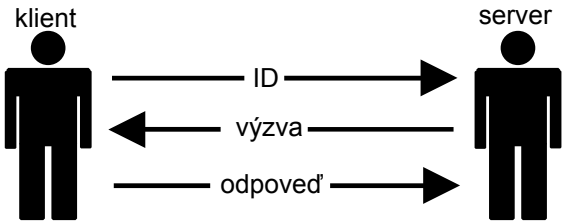
Poznáme viacero spôsobov autentizácie [7]:

- Prenos zašifrovaného hesla (obr. 2.1), kde klient zašle serveru svoje identefikačné údaje a zašifrované heslo *E(heslo)*. Útočník si však môže tieto informácie odchytiť a pomocou tzv. replay attacku sa dostane na klientov účet bez toho, aby potreboval poznať skutočné ID a heslo.



Obr. 2.1: Prenos zašifrovaného hesla

- Metóda výzva-odpoveď (obr. 2.2). Klient zašle serveru svoje ID na ktoré následne obdrží výzvu. Táto výzva je náhodne generované číslo a preto je vždy iná. Na každú výzvu existuje vždy práve jedna odpoveď. Bezpečnosť tohto spôsobu autentizácie je taká, aký dlhý šifrovací kľúč použijeme. V prípade krátkeho kľúča sa náhodne generované číslo môže znovu vygenerovať skôr ako pri použití kľúča dlhého. V tomto prípade je útok časovo veľmi náročný, ak vôbec realizovateľný.



Obr. 2.2: Metóda výzva-odpoveď

- Metódy, kde sa pre overeniu identity využíva tretia dôveryhodná strana, napr. Certifikačná Autorita alebo Kerberos (kap. 7).

2.2 Autorizácia

Po úspešnom autentizovaní nastáva proces prideľovania práv užívateľom. Ide o autorizáciu, kde administrátor systému pridelí oprávnenia, ktoré súbory a na aký účel môžu použiť [1]. Užívatelia sú rozdelení do skupín, kde každá skupina má iné práva.

Pre lepšie vysvetlenie tejto problematiky uvediem príklad. Máme veľkú počítačovú firmu, ktorá je rozdelená na skupiny:

- **riaditeľ firmy**, ktorý má prístup k celému systému,
- **účtovníčky**, ktoré majú prístup iba k účtovníckemu softwaru,
- **programátori**, ktorí majú prístup iba k tomu, čo sami tvoria,
- **ostatní zamestnanci**, ktorí majú povolený iba prístup na internet.

Autorizácia nefunguje iba vo veľkých firmách, ale je súčasťou napríklad operačného systému – užívateľské kontá. S autorizáciou sa stretávame aj mimo počítačovej techniky, napr. ak máte vodičský preukaz, môžete šoférovať auto (neberieme do úvahy porušenie zákona).

3 NIEKTORÉ TYPY ÚTOKOV

V tejto kapitole sú popísané niektoré typy útokov pri komunikácii cez internet.

3.1 Replay attack

Je druh „man-in-the-middle“ útoku. Útočník odchytí zašifrovaný priebeh komunikácie a môže ho zopakovať s tým, že nepozná jej reálny obsah. V prípade, že server nedetekuje duplikovanú požiadavku, môže to vyvolať napr. opakované zadanie platobného príkazu.

Tomuto nebezpečenstvu môžeme zabrániť napr. použitím časových lístkov, digitálnych certifikátov [22], alebo autentizovaním každého IP paketu napr. využitím autentižačných protokolov IPsec – ESP [13].

3.2 IP spoofing

Ide o druh útoku, kedy sa útočník snaží napodobniť komunikáciu pomocou zmeny IP paketov na falošné.

Veľa programov pri komunikácii cez internet využíva databázu hostov, ktorým dôveruje a nemusí si pýtať žiadne overenie identity. Toto využíva útočník – zmení obsah hlavičky IP paketu a obdrží dáta ktoré mal pôvodne obdržať naozajstný príjemca. Chrániť sa pred tymto typom útoku je možné tzv. viacstupňovým filtrovaním prichádzajúcich paketov [23].

3.3 Dictionaty attack

Slovníkový útok je efektívnejší spôsob odhaľovania hesiel ako tzv. „brute force attack“ (útok hrubou silou – software skúša všetky možné kombinácie)[26].

Užívatelia si v mnohých prípadoch ku svojím účtom dávajú slabé heslá (tzn. jednoduché frázy, napr. meno partnera, dátum narodenia alebo inej dôležitej udalosti, číslo domu alebo jednoduché „123456“) a toto je vhodný prípad použitia tzv. „slovníka“. Program na to určený skúša celý obsah slovníka rad za radom a snaží sa dospieť k pozitívnemu výsledku [25].

3.4 Denial of Service

Tento typ útoku je odlišný od ostatných spomínaných. V predchádzajúcich prípadoch išlo o ovládnutie prihlásenia a komunikácie. V tomto prípade ide o zlyhanie

služby. Využívajú sa slabiny v TCP/IP protokole na obmedzenie komunikácie – buď zhodenie prichádzajúcich paketov alebo odchádzajúcich.

Existuje viacero variant tohto útoku [27]:

- **Flood attack** – útočník zasiela viac paketov ako dokáže postihnutý počítač spracovať a tým zhodí celý systém,
- **Ping of Death Attack** – využíva sa program ping, ktorý vysiela pakety väčšie ako 65,535 bajtov,
- **SYN Attack** – na zahltenie komunikácie použije útočník veľké množstvo SYN správ,
- **Teardrop Attack** – útočník posiela poškodené IP pakety,
- **Smurf Attack** – využíva sa ping na broadcastovú adresu.

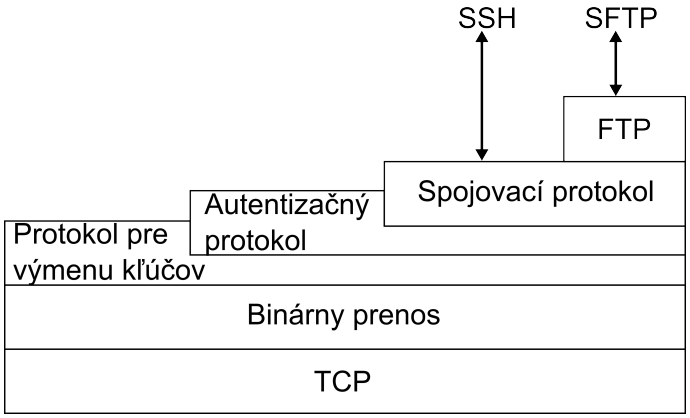
4 SSH

Secure Shell je protokol aplikačnej vrstvy typu klient-server, používaný najmä v systémoch UNIX a Linux [4]. Vytvoril ho Tatu Ylonen [10]. Dokáže naviazať zabezpečené spojenie a spúšťať programy na vzdialenom počítači cez tzv. tunel. Používa sa ako náhrada za telnet, rsh a rlogin.

Pre SSH komunikáciu je potrebný SSH server, SSH klient a kľúč či heslo k pripojeniu sa na server. Jeho zásadnou vlastnosťou je, že sa musí preukázať klient serveru aj server klientovi. Secure Shell existuje v dvoch verziách. SSHv1 podporuje RSA kľúče, SSHv2 RSA aj DSA [9]. Šifrovací kľúč sa prenáša asymetrickou šifrou, dáta naopak symetrickou. K autentizácii sa využíva kombinácia prihlasovacie meno a heslo, alebo RSA kľúč, ktorý nahráme na server a uľahčíme si tak opakované vypisovanie prístupového hesla.

Secure Shell sa skladá z viacerých protokolov. Ich organizáciu môžete vidieť na obr. 4.1 Sú to:

- Protokol pre výmenu kľúčov – Key Exchange Protocol – zabezpečuje autentizáciu serveru, dôvernosť a integritu, voliteľne aj kompresiu,
- Autentizačný protokol – User Authentication Protocol – pomocou protokolu na transportnej vrstve sa stará o autentizáciu zo strany klienta,
- Spojovací protokol – Connection Protocol – multiplexuje šifrovaný tunel do niekoľkých logických kanálov.



Obr. 4.1: Organizácia SSH

Hoci je SSHv2 novší od SSHv1, je napísaný od úplného základu. Hlavným rozdielom je použitie šifrovacích algoritmov – kým SSHv1 podporuje DES, 3DES, IDEA a Blowfish, SSHv2 vypustilo algoritmy IDEA a DES a pridalo podporu Twofish, Arcfour a Cast128-cb [6, 16]. Pri prihlasovaní SSH klienta verzie 1 na SSH server verzie 2 narazíme na problém s kompatibilitou. V tomto prípade sa môže server hlásiť pod verziou 1.99 a problém je vyriešený na oboch stranách.

SSHv2 podporuje protokol SOCKS (protokol, ktorý umožňuje smerovanie paketov medzi klientom a serverom cez proxy server) [2]. Dovoľuje zmenu hesiel. Používa viac metód na výmenu kľúčov a rozšírené dohadovanie algoritmov medzi klientom a serverom. Veľkým prínosom je silnejšia kontrola integrity dát pomocou kryptografických metód a tzv. prekľúčovanie (pravidelná výmena kľúča relácie) [1].

4.1 SSH server

Je to démon (sshd), ktorý vytvára podmienky pre spojenie. To znamená, že vytvorí pre každé prichádzajúce spojenie nový podproces. Úlohou podprocesu je zabezpečiť výmenu kľúčov, autentizáciu a výmenu šifrovaných správ.

Démon si pri spustení a využití verzie 1 vždy vygeneruje tajný kľúč, zvyčajne 768-bitový. Tento sa vo východzom nastavení generuje každú hodinu v prípade, že ho behom tejto doby nikde nepoužil a nikdy nie je uložený na disku v počítači.

Bezpečnosť komunikácie pomocou verzie 2 zaisťuje démon využitím Diffie-Hellmanovho algoritmu, ktorý vytvára tajný kľúč relácie. Ostatná komunikácia je zabezpečená symetrickými šiframi, integrita dát pomocou HMAC (Hash-based Message Authentication Code). Klient si najprv zistí, ktoré algoritmy server podporuje a podľa toho prispôsobí svoju požiadavku na šifrovací algoritmus [9].

4.2 SSH klient

Klientský program vykonáva prihlásenie ku vzdialenému počítaču a spúšťa užívateľské príkazy. Po pripojení na SSH server je overená jeho identita a v prípade jej potvrdenia nasleduje zabezpečená relácia.

Protokol SSH nemá však úplne voľnú licenciu. Preto bola vytvorená jeho obdoba OpenSSH.

4.3 OpenSSH

OpenSSH je otvorený projekt s voľnou licenciou. Vo verzii 3.0 implementuje SSHv1 aj SSHv2 [11] ale vo východzom nastavení je verzia 2. Bol vytvorený projektom OpenBSD. Pracujú na ňom dve skupiny. Prvá sa stará o vytvorenie čistého, prehľadného a hlavne bezpečného kódu, druhá tento kód upravuje pre použitie v rôznych operačných systémoch.

OpenSSH poskytuje silné šifrovanie – využíva algoritmy 3DES, Blowfish, AES, Arcfour. Tak isto ponúka silnú autentizáciu verejným kľúčom a využitím Kerbera.

Podporuje tiež kompresiu prenášaných dát a smerovanie portov a X11. Poskytuje ochranu pred útokmi IP spoofing, fakes routes a DNS spoofing [21].

Najznámejším SSH klientom je PuTTY, pomocou ktorého môžeme pri komunikácii cez FTP vynútiť šifrovaný prenos [19].

Najnovšie je OpenSSH vo verzii 5.4, ktorá vo východzom nastavení úplne zakázala SSHv1. Pre autentizáciu sa využíva nový, minimálny formát OpenSSH [24].

4.4 Inštalácia ssh serveru

Pre správny chod zabezpečenej relácie potrebujeme SSH klient a SSH server. Na počítači, ktorý si zvolíme ako server si nainštalujeme ssh server.

```
$ sudo apt-get install openssh-server
```

Na druhom počítači, zvolenom ako klient, nie je potrebné nič inštalovať, iba vygenerovať kľúče.

```
$ ssh-keygen -t ssh-rsa
```

Pre prihlásenie na server zadáme do konzoly

```
$ ssh <meno-pouzivatela>@<stanica>
```

v našom prípade `ssh mehmed@147.229.148.151`. Systém sa nás opýta, či chceme na server pridať svoj tajný kľúč. Napíšeme `yes`. Ďalej sa nás systém spýta na heslo užívateľa `mehmed`. Zadáme správne heslo a sme prihlásení na vzdialenom počítači.

Pri každom prihlasovaní na vzdialený počítač sa nás systém spýta na heslo, pretože SSH sa zakaždým po vykonaní požiadavky z neho odhlási. Ak chceme tomuto predísť, musíme na server skopírovať svoj tajný kľúč, podľa ktorého si server overí klienta bez zadávania hesla. Dosiahneme to tak, skopírujeme svoj kľúč do adresára na serveri, obsah súboru `id_rsa.pub` pridáme na koniec súboru `authorized_keys`, kde sú uchovávané všetky verejné kľúče, ktoré sa následne porovnávajú s tajným kľúčom na lokálnom počítači. V prípade zhody sme prihlásení.

```
$ scp ~/.ssh/id_rsa.pub <meno-pouzivatela>@<stanica>:~/.ssh/
<meno-pocitaca>.pub
$ ssh <meno-pouzivatela>@<stanica>
$ cd ~/.ssh
$ more <meno-pocitaca>.pub >> authorized_keys
$ exit
```

kde za meno-pouzivatela zadáme `mehmed`, stanica `147.229.148.151` a meno-pocitaca `mehmed`. Systém sa nás ešte opýta na zadanie bezpečnostnej frázy a v prípade, že žiadnu nezadáme, sme schopní prihlásiť sa bez zadávania hesla [28].

4.5 Výsledky merania

Funkčnosť tohto protokolu som si overil prenosom súboru cez zašifrovaný kanál pomocou programu, ktorý je súčasťou balíku openssh – *scp*. Popri otvorenej konzole, pomocou ktorej som spúšťal scp som mal spusteného správcu procesov, ktorý slúžil pre zistenie zaťaženia systému. Zaťaženie siete (rýchlosť prenosu) zobrazoval sám program scp.

Pri meraní prenosových vlastností na počítači, ktorý slúžil ako klient, som prišiel na nasledovné skutočnosti (viď tab. 4.1):

Tab. 4.1: Prehľad merania zabezpečením SSH

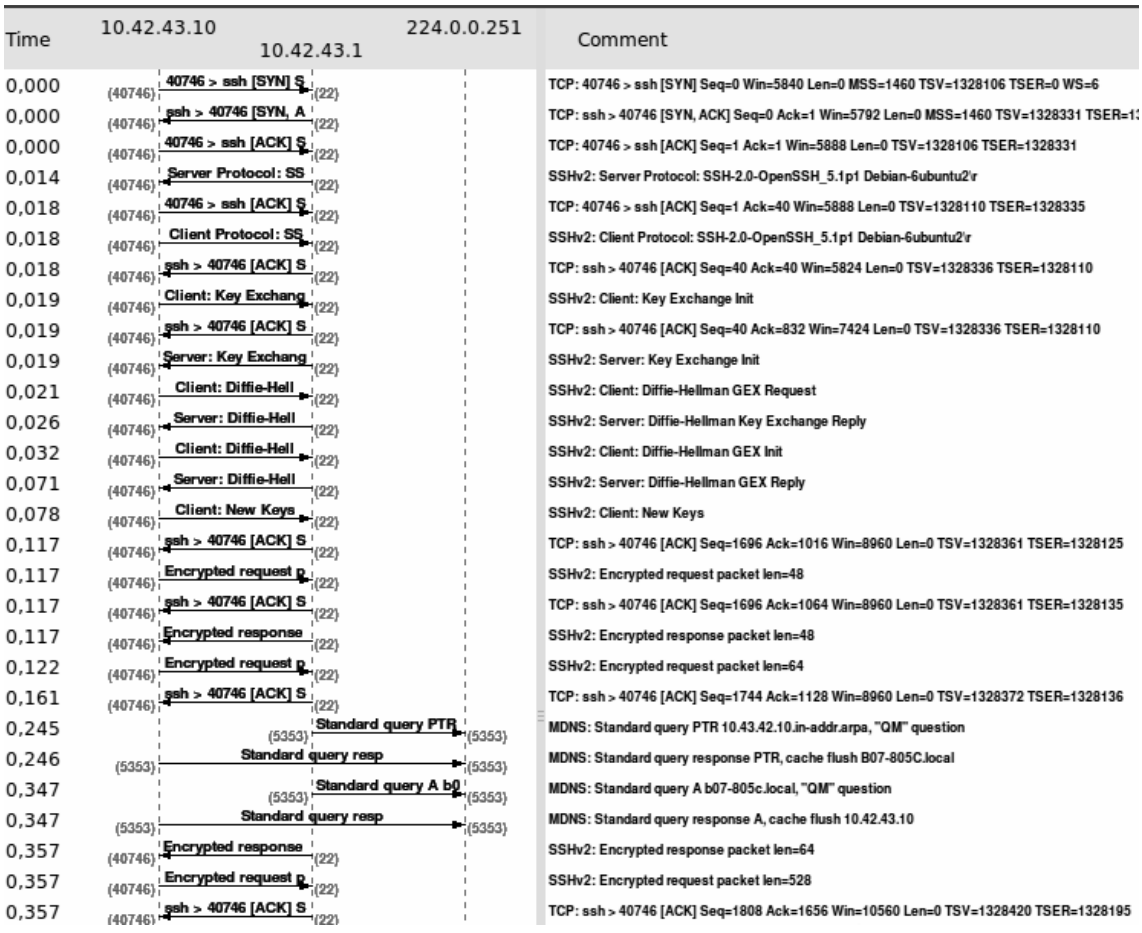
Zabezpečenie	využitie CPU	prenosová rýchlosť
Bez zabezpečenia	10 %	8,5 MB/s
SSH bez kompresie	31 %	11,2 MB/s
SSH s kompresiou	98 %	7 MB/s

Pri prenose bez zabezpečenia som dosial rýchlosť len 8,5 MB/s. Mohlo to byť spôsobené niektorým spusteným procesom na pozadí. Bez kompresie sa rýchlosť vyšplhala na 11,2 MB/s a využitie procesoru vzrástlo vďaka použitému šifrovaniu na 31 %. Rýchlosť pri prenose s využitím kompresie bola obmedzená výkonom procesoru na notebooku, ktorý bol úplne vyťažený, a dosiahla priemernú hodnotu 7 MB/s.

4.6 Priebeh spojenia

Priebeh spojenia je zobrazený na obr. 4.2.

Pri nadväzovaní spojenia zašle klient TCP správu serveru na port 22. Nasleduje trojcestný handshake typický pre TCP. Potom sa obidve strany dohodnú na použitom protokole a zahája výmenu kryptografických kľúčov pomocou Diffie-Hellmanovho šifrovacieho algoritmu. Ďalej klient zasiela požiadavky o pakety, na ktoré mu server pozitívne odpovedá. Po úspešnom prenose súboru klient a server zrušia spojenie.



Obr. 4.2: Priebeh komunikácie pomocou SSH

5 SSL

Protokoly SSL a TLS (komplexne nazývané SSL) sa starajú o bezpečnosť na vyšších vrstvách. Poskytujú plne duplexnú komunikáciu. Dokážu zabežpečiť všetky protokoly, ktoré pre komunikáciu využívajú protokol TCP.

5.1 Secure Socket Layer

Secure Socket Layer je protokol prezentačnej vrstvy OSI modelu, ktorý vytvorila firma Netscape. Je dnes najpoužívanejším zabezpečovacím mechanizmom pre web. Je pomerne dobre vyvinutý.

Pre každú novú reláciu sa vytvorí nové bezpečné spojenie. Jeho prítomnosť spoznáme v paneli s adresou kde sa nachádza „https“. Podporuje viaceré algoritmy na dohodnutie kľúčov, šifrovanie a hashovanie. Pre autentizáciu sa využívajú mená a heslá, mená a token alebo digitálne certifikáty.

Protokol SSL sa skladá z dvoch protokolov [4]:

1. Record Layer – stará sa o fragmentáciu dát, ich šifrovanie a autentizáciu,
2. Change Cipher Spec – slúži na signalizáciu šifrovania a autentizáciu po úspešnom naviazaní spojenia (handshake) a na signalizáciu chýb (alert).

5.2 Transport Layer Security

TLS – protokol transportnej vrstvy – je oficiálnym protokolom Internetu a poskytuje tak isto zabezpečenie webovej relácie. Úlohou tohto protokolu je znížiť nároky na šírku pásma skrátením dĺžky fragmentov [4]. Je súčasťou transportnej vrstvy OSI modelu. TLS pôvodne vychádza z protokolu SSLv3 ale spolu nie sú kompatibilné, hoci majú mnoho spoločných vlastností. TLS používa na výmenu kľúčov, šifrovanie a hashovanie iné algoritmy. Podporuje hlásenie chybových správ. Obľúbili si ho aj tvorcovia GSM aplikácii a preto ho implementovali do svojich aplikácií [2].

Obidva protokoly pri relácii vyžadujú autentizáciu serveru (aj keď TLS podporuje plne anonymnú reláciu). Overenie identity klienta je voliteľné. Na začiatku relácie sa klient a server najprv dohodnú na použitých kryptografických algoritmoch. Následne si vymenia náhodné čísla preto, aby mohlo dôjsť k bezpečnej výmene kryptografických kľúčov pre prenos a autentizáciu. Nasleduje výpočet kontrolných súčtov, ktoré slúžia spolu so zdieľaným tajomstvom na autorizáciu.

Na udržanie bezpečnosti počas komunikácie overuje klient Certifikačnú Autoritu, či sa v jej zozname nachádza server, s ktorým komunikuje, a môže mu dôverovať.

Klient tak isto overuje platnosť certifikátu, a v prípade, že dátum nespadá do obdobia jeho platnosti, znemožní ďalšiu komunikáciu [2]. Na obr. 5.1 je znázornený prípad overovania certifikátu na zabezpečenom serveri. Prehliadač označí certifikát ako neoverený a nie bezpečný. Detaily certifikátu sú uvedené v prílohe A.

Obr. 5.1: Oznámenie neovereného certifikátu

Nevýhodou tohoto protokolu je jeho rýchlosť. Oproti nezabezpečenému TCP prenosu je pomalší z toho dôvodu, pretože tu dochádza k výmene mnohých informácií pre udržanie bezpečnosti.

5.3 OpenSSL

OpenSSL je kryptografický protokol s voľnou licenciou, ktorý v sebe zahŕňa protokoly SSL (v2, v3) a TLS (v1). Jeho najnovšia verzia 1.0.0 bola vydaná 29. marca 2010.

5.4 Vytvorenie zabezpečeného serveru

Pre sfunkčnenie SSL protokolu som zvolil postup nainštalovania serveru Apache2. Ako prvé si nainštalujeme samotný OpenSSL balíček a Apache2 server. Pri vytváraní konfigurácie som používal [5] a [14].

```
$ sudo apt-get install openssl
$ sudo apt-get install apache2
```

Teraz nainštalujeme a zavedieme SSL.

```
$ sudo a2enmod ssl
$ sudo /etc/init.d/apache2 force-reload
```

Ďalej sa presunieme do zložky `/var/www` a vytvoríme si tu jednoduchú html stránku `index.html`. Teraz máme vytvorený nezabezpečený webový server. Pre jeho zabezpečenie potrebujeme vytvoriť tajné kľúče serveru. Presunieme sa do zložky `/etc/apache2` a tu vygenerujeme RSA kľúč.

```
$ cd /etc/apache2
$ sudo openssl genrsa -des3 -out server.key 1024
```

Ďalej si vytvoríme vlastný certifikát.

```
$ sudo openssl req -new -key server.key -out server.csr
```

Do nášho certifikátu môžeme vložiť mnoho informácií. Dialóg na vkladanie informácií vyzerá v mojom prípade nasledovne:

```
root@PA-274:/etc/apache2#  sudo openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CZ
State or Province Name (full name) [Some-State]:Czech Republic
Locality Name (eg, city) []:Brno
Organization Name (eg, company) [Internet Widgits Pty Ltd]:utko VUT
Organizational Unit Name (eg, section) []:student
Common Name (eg, YOUR name) []:Michal Ludik
Email Address []:xludik01@stud.feec.vutbr.cz

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:mehmed
An optional company name []:Home
root@PA-274:/etc/apache2#
```

Nasledujúcim príkazom si vytvoríme samopodpísateľný certifikát, ktorého platnosť bude jeden rok.

```
$ sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt
```

Nasleduje inštalácia kľúča a certifikátu.

```
$ sudo cp server.crt /etc/ssl/certs/
$ sudo cp server.key /etc/ssl/private/
```

Presunieme sa do zložky `/etc/apache2/sites-available`, otvoríme súbor `default-ssl` a odkomentujeme v ňom nasledujúce riadky:

```
SSLEngine on
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

a tým zabezpečíme chod portu 443 pre protokol SSL.

Pri reštartovaní serveru na nás vyskočí chybová hláška. Preto musíme otvoriť súbor `/etc/apache2/httpd.conf` (ktorý bude pravdepodobne prázdny) a pripísať doň `ServerName localhost`. Nakoniec aktivujeme východziu SSL stanicu a reštartujeme apache2 server.

```
$ sudo a2ensite default-ssl
$ sudo /etc/init.d/apache2 restart
```

V tomto okamihu máme funkčný zabezpečený webový server.

5.5 Výsledky merania

Pre získanie potrebných výsledkov som zvolil metódu stiahnutia súboru zo zabezpečenej stránky. Do jednoduchého html kódu som pridal odkaz na stiahnutie súboru. Vo webovom prehliadači som klikol na odkaz tohto súboru a hneď nato sa spustilo jeho sťahovanie.

V tab. 5.1 je stručný prehľad náročnosti SSL protokolu pri prenose súborov.

Tab. 5.1: Prehľad merania zabezpečením SSL

Zabezpečenie	využitie CPU	prenosová rýchlosť
Bez zabezpečenia	10 %	11,3 MB/s
So zabezpečením	98 %	7 MB/s

Pri prenose súborov nezabezpečenou sieťou som dosiahol rýchlosť 11,3 MB/s a nízke zaťaženie systému. Akonáhle som aktivoval protokol SSL, prenos súboru sa spomalil na 7 MB/s a využitie procesoru stúplo na 98 %.

5.6 Priebeh spojenia

Na obr. 5.2 je znázornený priebeh komunikácie medzi počítačmi pomocou protokolu SSL, konkrétne TLSv1.

1. Klient zašle serveru správu *Client Hello*.
2. Server odpovedá *Server Hello*, ktorý obsahuje certifikát a indikáciu výmeny tajného kľúča.

3. Klient potvrdí výmenu kľúča, zašle správu s výzvou o šifrovací algoritmus (protokol *Change Cipher Spec*) a šifrovaný *Handshake*.
4. Server potvrdí výzvu odpoveďou so šifrovaným *Handshake* a *Change Cipher Spec*.
5. Klient zašle informáciu o dáta, ktoré chce stiahnúť.
6. Server odpovedá príslušnými zašifrovanými dátami. Tento proces sa stále opakuje.

Obr. 5.2: Priebeh komunikácie pomocou SSL

6 IPSEC

Na prenos súborov cez internet slúži Internet Protocol (IP). Ten však nie je nijak zabezpečený. Preto bolo preň vytvorené rozšírenie s názvom IPsec.

IPsec je skupina protokolov zabezpečujúca pakety prenášané sieťou na úrovni sieťovej vrstvy [4]. Nezabezpečuje iba protokoly TCP/UDP ako SSL, ale celú komunikáciu (viď obr. 6.3) [10]. Pôvodne bol vyvinutý pre IPv6, ale postupne ho prepracovali aj pre IPv4 [29]. Podporuje všetky šifrovacie algoritmy, ktoré sa v súčasnosti používajú (naproti SSL používa iba jednoduché). Dokáže sa prispôsobiť aj novým [15].

Výhodou tohto protokolu je, že sa vyskytuje na úrovni operačného systému, alebo úrovni smerovača a preto systém ani IPsec podporovať nemusí. Na rozdiel od SSL, ktoré sa stará o zabezpečenie aplikácie, IPsec sa stará o vytvorenie bezpečného kanálu.

IPsec dokáže fungovať v dvoch režimoch:

- **transportný** – tzv. host to host. Vytvorí sa spojenie priamo iba medzi dvomi stanicami v jednej sieti,
- **tunelový** – vytvorí VPN tunel medzi dvomi sieťami.

IPsec sa skladá z dvoch hlavných protokolov – Authentication Header (AH) a Encapsulated Security Payload (ESP). AH má na starosti overenie neporušených odosielaných datagramov a autentizáciu odosielateľa; ESP podporuje aj šifrovanie dát. VPN systémy dokážu obidva protokoly použiť zároveň vďaka čomu sa zvýši zabezpečenie datagramu, ale zníži výkon systému [15].

6.1 Protokol AH

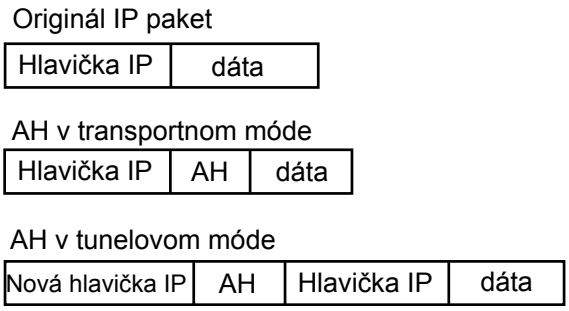
Authentication Protocol je rýchly protokol, pretože jeho úlohou je len autentizácia dát. Šifrovanie, ktoré je náročné na systémové prostriedky má za úlohu ESP.

Protokol AH autentizuje celý datagram aj s hlavičkou. Pre ochranu informácií používa algoritmus MD5 a autentizáciu vykonáva algoritmus HMAC (typ autentizačného kódu správy, ktorý sa vypočíta pomocou hashu a tajného kľúča [17]). Jeho hlavička obsahuje sekvenčné čísla, ktoré zabezpečujú ochranu pred útokom *replay-attack* [15].

Protokol AH je možné použiť dvomi spôsobmi (viď. obr. 6.1):

- **Trasportný režim** – je náchylný na odpočúvanie, pretože informácie v IP datagramie nie sú nijak zabezpečené. Skladá sa z originálnej IP hlavičky datagramu, AH hlavičky a užitočného zaťaženia datagramu. Autentizovaný je celý datagram okrem premenlivých polí.

- **Tunelový režim** – v tomto prípade je autentizovaný celý datagram a je možné overiť, či sa pri prenose nezmenil. Vytvára hlavičky IP za ktorou nasleduje hlavička AH a pôvodný datagram. Tunelový režim podporuje používanie súkromných adries.



Obr. 6.1: Paket zabezpečený AH v transportnom a tunelovom móde

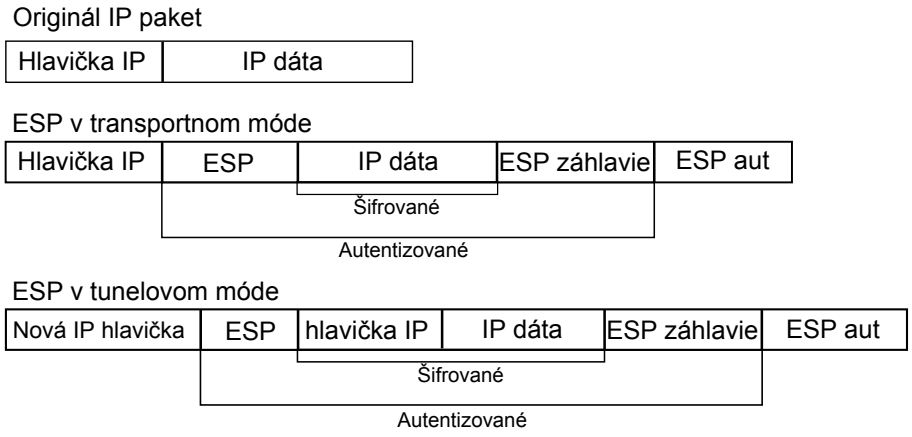
6.2 Protokol ESP

Encapsulating Security Payload protokol je náročnejší na systém, pretože naviac od AH protokolu zabezpečuje šifrovanie dát aj s hlavičkou. Šifrovanie prebieha použitím symetrického šifrovania. Informácie sú chránené rovnako ako u AH – pomocou HMAC.

Protokol ESP môžeme použiť podobne ako AH, v transportnom a tunelovom režime (viď obr. 6.2). V transportnom móde je prvá IP hlavička, potom hlavička ESP, nasledujú šifrované dáta, návestie ESP a autentizácia údajov. Tunelový režim je bezpečnejší ako transportný, pretože autentizuje a šifruje aj hlavičku IP datagramu a tak nie je možné pozmeniť údaje v nej. Rovnako ako pri AH, vytvára ESP hlavičky IP. Za ňou nasleduje ESP hlavička, hlavička IP, šifrované dáta, návestie ESP a autentizácia údajov [15].

6.3 Protokol IKE

Internet Key Exchange je protokol, ktorý zabezpečuje generovanie a výmenu kryptografických kľúčov v protokole IPsec a dohaduje všetko to, čo je potrebné pre úspešný prenos. Deje sa tak pomocou Security Association (SA), úlohou ktorej je napr. identifikovať typy algoritmov, kryptografických kľúčov a dobu ich existencie atď. Implementácia prebieha v dvoch fázach [15]. Pre autentizáciu môže využívať aj digitálne podpisy [4].



Obr. 6.2: ESP paket v transportnom a tunelovom móde

Prvá fáza slúži na vytvorenie bezpečného kanálu. Vytvorí hlavný tajný kľúč, z ktorého sa odvodzujú ďalšie šifrovacie kľúče na ochranu prenosu údajov užívateľa. Takto to prebieha, aj keď medzi dvomi koncovými bodmi ešte neexistuje žiadna bezpečnostná ochrana. Fáza jedna ochraňuje správy fáze 2.

Fáza dva dohoduje SA a kľúče, ktoré ochraňujú aktuálne výmeny údajov aplikácie. Po vyjednaní fázy 2 sa vytvorí bezpečné dynamické spojenie cez sieť a medzi koncovými bodmi, ktoré sú definované v pripojení.

Vyjednanie fázy 1 prebieha raz za deň, pričom fáza 2 je vyjednávaná každú hodinu alebo 5 minút. Použitie kratšej doby prináša vyššiu bezpečnosť ale aj väčšie zaťaženie systému [15].

6.4 ISAKMP

Protokol ISAKMP (Internet Security Association And Key Management Protocol) je dynamický aplikačný protokol, ktorý slúži na plnenie SA databázy kryptografickými údajmi na oboch stranách spojenia [2]. Jeho súčasťou je protokol IKE popisovaný v kap. 6.3. Pre výmenu informácií používa protokol UDP na porte 500. Môže byť implementovaný v ktoromkoľvek transportnom protokole [18].

6.5 Zabezpečená komunikácia

Ako je známe, IPsec je schopný fungovať v dvoch režimoch – transportnom a tunelovom. Pre overenie funkcie som si zvolil prvý – transportný režim označovaný aj ako „host-to-host“. V prvom rade je potrebné na obidva počítače nainštalovať balík `ipsec-tools`.

```
$ sudo apt-get install ipsec-tools
```

Ďalej je potrebné upraviť konfiguračný súbor `/etc/ipsec-tools.conf`. Mal by vyzerať nasledovne:

```
#!/usr/sbin/setkey -f

# Configuration for 147.229.148.151

# Flush the SAD and SPD
flush;
spdflush;

# AH SAs using 128 bit long keys
add 147.229.148.151 147.229.196.145 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 147.229.196.145 147.229.148.151 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add 147.229.148.151 147.229.196.145 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 147.229.196.145 147.229.148.151 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd 147.229.148.151 147.229.196.145 any -P out ipsec
esp/transport//require
ah/transport//require;

spdadd 147.229.196.145 147.229.148.151 any -P in ipsec
esp/transport//require
ah/transport//require;
```

V prípade druhého počítača nám stačí konfiguračný súbor skopírovať a vymeniť položky „out“ a „in“. Zmeníme prístupové práva tak, aby nebolo možné zmeniť konfiguračný súbor,

```
$ sudo chmod 750 /etc/ipsec-tools.conf
```

na každom počítači aktivujeme konfiguračný súbor

```
$ sudo setkey -f /etc/ipsec-tools.conf
```

a spustíme samotný protokol.

```
$ sudo /etc/init.d/setkey start
```

6.6 Výsledky merania

Na vykonanie tejto časti som využil už vytvorený ftp server. V tab. 6.1 je prehľad výsledkov merania priemerného zaťaženia systému a priemerná prenosová rýchlosť. Rýchlosť prenosu súboru bez použitia IPsecu bola maximálna, akú prepustila sieťová

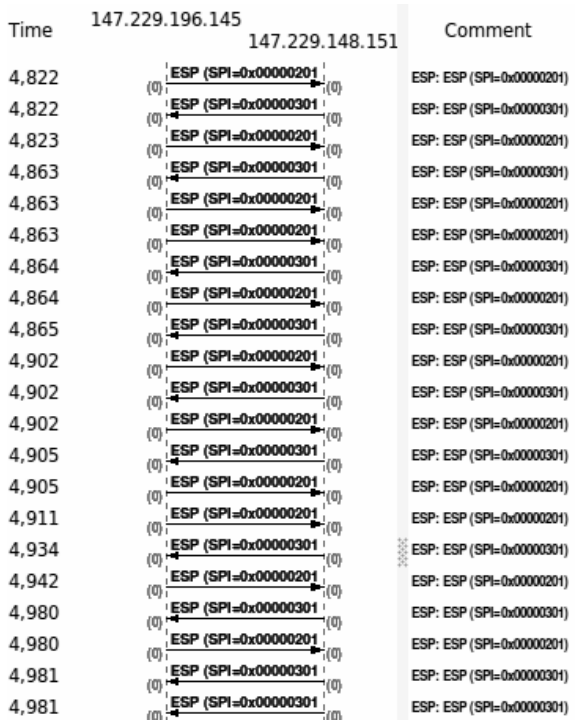
Tab. 6.1: Prehľad merania zabezpečením IPsec

Zabezpečenie	Využitie CPU	Prenosová rýchlosť
Bez zabezpečenia	7 %	11,3 MB/s
So zabezpečením	90 %	6,3 MB/s

karta – 11,3 MB/s. Priemerné zaťaženie systému bolo 7 % na strane notebooku so slabším procesorom. Po aktivovaní zabezpečenia sa priemerná rýchlosť znížila na 6,3 MB/s a zaťaženie systému stúplo na 90 %.

6.7 Priebeh spojenia

Priebeh komunikácie zachytenej Wiresharkom je možné vidieť na obr. 6.3. Ako je možné všimnúť si, z existujúcej komunikácie nie je možné nič vyčítať.



Obr. 6.3: Priebeh komunikácie zabezpečenej IPsecom

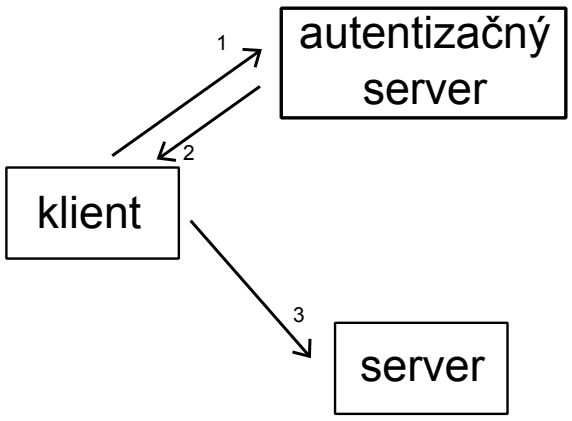
7 KERBEROS

Kerberos je pomerne starý autentizačný protokol s vyše 20-ročnou históriou a vďaka tomu je dobre prepracovaný. Výhodou tohto protokolu je to, že nie je viazaný na jednu vrstvu OSI modelu, ale dokáže sa prispôsobiť podľa toho, aký protokol zabezpečuje. Pre komunikáciu využíva symetrické šifrovanie založené na DES a vyžaduje tretiu stranu, samostatný server tzv. Key Distribution Center (KDC).

7.1 Kerberos ako autentizačný protokol

Autentizácia v Kerbere funguje na základe lístkov, databáza ktorých sa udržiava práve v KDC. V KDC sa udržiavajú ako heslá klientov, tak aj serverov a preto musí byť dobre zabezpečený.

Pri komunikácii – obr. 7.1 – zašle užívateľ žiadosť autentizačnému serveru o autentizáciu, kde označí seba a server, s ktorým chce nadviazať spojenie. Užívateľovi príde odpoveď vo forme náhodného čísla, ktoré autentizačný server vygeneroval a zašifroval kľúčom užívateľa aj serveru. Užívateľ prepošle serveru jeho správu. Obidvaja si svoje správy dešifrujú a toto náhodné číslo slúži ako tzv. session key, kľúč pre jedinú reláciu.



Obr. 7.1: Proces dohodnutia tajných kľúčov

V prípade použitia kľúča relácie tu existuje isté bezpečnostné riziko. Ak útočník zachytí správy, ktoré odchádzajú od užívateľa, dokáže ich pozmeniť a tak sa vydávať za server. V tomto prípade má prístup ku všetkým informáciám. Reakciou na tento bezpečnostný problém je využitie Needham-Schroederovho protokolu. Ten vkladá do správ z KDC okrem kľúča relácie aj identifikáciu užívateľa a servera, ktorú môže spätne overiť [3].

Jeho najnovšia verzia je pod číslom 5. Staršie verzie neboli oficiálne zverejnené, iba verzia 4 [3]. Toto najnovšie vydanie zlepšuje kompatibilitu s Microsoft Windows, kvalitu kódu, vývoj kódu [19].

Hlavnou nevýhodou Kerbera je to, že ak vypadne hlavný server, nie je možné sa prihlásiť. Tento problém sa dá vyriešiť použitím viacerými záložných KDC serverov. Ďalším problémom sú neplatné časové lístky pri autentizácii. Tento prípad nastáva vtedy, ak nemáme vzájomne zosynchronizovaný KDC server a komunikujúce počítače.

7.2 Vytvorenie zabezpečenej relácie

Na začiatku je potrebné nainštalovať server a KDC databázu. Ja som si tieto dve položky zvolil na jednom počítači. Pre sfunkčnenie tejto časti som používal [8].

```
$ sudo apt-get install krb5-admin-server krb5-kdc
```

Teraz je potrebné upraviť konfiguračný súbor Kerbera v `/etc/krb5.conf`. Vytvoríme východzí realm (doména, úlohou ktorej je autentizácia užívateľov)

```
[libdefaults]
default_realm = MICHAL.LUDIK

[realms]
MICHAL.LUDIK = {
kdc = kdc
admin_server = kdc
}
```

ďalej vytvoríme doménu

```
[domain_realm]
.michal.ludik = MICHAL.LUDIK
michal.ludik = MICHAL.LUDIK
```

a pridáme zaznamenávanie

```
[logging]
kdc = FILE:/var/log/kerberos/krb5kdc.log
admin_server = FILE:/var/log/kerberos/kadmin.log
default = FILE:/var/log/kerberos/krb5lib.log
```

Pre korektné fungovanie konfiguračného súboru si ešte vytvoríme súbory pre záznamy a pridelíme im práva

```
$ sudo mkdir /var/log/kerberos
$ sudo touch /var/log/kerberos/{krb5kdc,kadmin,krb5lib}.log
$ sudo chmod -R 750  /var/log/kerberos
```

Ešte je potrebné do `/etc/hosts` pripísať nasledujúce údaje

```
147.229.148.151 kdc.michal.ludik kdc
```

Nasleduje vytvorenie realmu,

```
$ sudo krb5_newrealm
```

prihlásime sa a vytvoríme nového užívateľa `root/admin`

```
$ sudo kadmin.local
kadmin.local: addprinc root/admin
```

zoznam účtov si môžeme overiť

```
kadmin.local: listprincs
```

Ďalej potrebujeme upraviť súbor `/etc/krb5kdc/kadm5.acl`, ktorý obsahuje prístupové práva. Je tu definovaný užívateľ, ktorý bude môcť ovládať KDC ako administrátor. Na konci súboru musí byť `*admin *`. Znamená to, že administrátorské práva získa každý užívateľ, ktorý bude mať vo svojom účte reťazec „/admin“, u nás konkrétne root/admin.

Nakoniec reštartujeme admin-server a KDC databázu

```
$ sudo /etc/init.d/krb5-admin-server restart
$ sudo invoke-rc.d krb5-kdc restart
```

Pre konfiguráciu klienta potrebujeme nainštalovať príslušné balíky.

```
$ sudo apt-get install krb5-user krb5-clients
```

Upravíme konfiguračný súbor `/etc/krb5kdc/kdc.conf` tak, že upravíme východzí realm na ten náš (MICHAL.LUDIK). Pre plne funkčný model Kerbera je potrebné zosynchronizovať čas medzi počítaťmi potmocou protokolu NTP.

Keďže Kerberos je predovšetkým autentizačný protokol, pre overenie jeho funkčnosti som zvolil protokol ftp. Jedná sa o program `krb5-ftp` – špeciálny klient ftp, ktorý podporuje autentizáciu Kerberom. V tomto príklade som si zvolil overenie Kerbera pri prihlasovaní sa na ftp server. Prenos dát ale prebieha nezabezpečene, viď príloha B.

Kerberos je známy tým, že poskytuje autentizáciu aj iným protokolom. V mojom prípade som vykonal zabezpečenie autentizácie protokolu SSH. Jedná sa o nahradenie autentizácie pomocou kľúčov Kerberovými lístkami. Je potrebné upraviť konfiguračný súbor SSH démona tak, že odkomentujeme riadok s `KerberosAuthentication yes`. Teraz je potrebné zmazať lístok `kdestroy` a nadviazať spojenie `ssh mehmed@kdc` [12].

Autentizácia v tomto prípade funguje tým spôsobom, že klient sa pripája na server pod užívateľským menom a ten žiada od klienta Kerberovský lístok. V tejto fáze prebehne `kinit` a server si od užívateľa vyžiada kerberovské heslo.

7.3 Priebeh spojenia

Na obr. 7.2 je zobrazený priebeh prihlásenia sa na ftp server. Pred samotným prihlasovaním je možné vidieť synchronizáciu času mezdi naším klientom a serverom. Nasleduje inicializácia klienta `kinit michal` a jeho autentizácia pomocou hesla. Prihlásime sa na náš ftp server `kbr5-ftp kdc`, zadáme prihlasovacie meno `michal`, od TGS obdržíme lístok a sme prihlásení na ftp serveri. Je možné si tu všimnúť, že ftp server od nás nežiada žiadne heslo. Je to kvoli lístkom, ktoré fungujú ako RSA kľúč pri SSH, kedy pri prihlasovaní sa na vzdialený počítač od nás server nevyžaduje heslo (viď kap. 4.4).

Obr. 7.2: Priebeh prihlásenia na ftp server

8 ZHRNUTIE VLASTNOSTÍ PROTOKOLOV

V tab. 8.1 je komplexný súhrn vlastností všetkých protokolov – vrstva OSI modelu, autentizácia, bezpečnosť a nároky na systém.

Protokol SSH zabezpečuje aplikácie na aplikačnej úrovni. Rovnako je na tom aj SSL, ktoré dokáže zabezpečovať aj protokoly transportnej vrstvy referenčného ISO/OSI modelu. Ďalším protokolom je autentizačný protokol Kerberos, ktorý zabezpečuje vyššie vrstvy podľa toho, kde potrebujeme zaistiť overenie identity. Kerberos je možné implementovať napr. ako súčasť SSH. IPsec je protokol, ktorý dokáže zabezpečiť nielen celú komunikáciu, ale môže byť súčasťou smerovača či iného zariadenia a preto nemusí byť v systéme implementovaný a tak nezaťažuje náš počítač.

Autentizáciu protokolu SSH je zaistená heslom alebo RSA kľúčom. Protokol SSL využíva pre autentizáciu Certifikačnú autoritu od ktorej si vyžiada potvrdenie platnosti certifikátu. V prípade, že svoj certifikát nemáme podpísaný od CA, je len na užívateľovi, či mu bude dôverovať. Kerberos na overenie identity používa lístky, ktoré musia byť časovo zosynchronizované, inak dôjde k zamietnutiu požiadavky o autentizáciu. U IPsecu zaisťujú autentizáciu tajné kľúče, ktoré sú na oboch počítačoch zhodné a sú definované pre každý smer komunikácie.

Secure Shell využíva pre autentizáciu heslá, ktoré je možné ohroziť replay attackom. Niektorí užívatelia si k svojím účtom dávajú veľmi slabé heslá, ktoré by sa dali odhaliť slovníkovým útokom. SSH tiež možno ohroziť útokom hrubou silou. V prípade SSL je to rovnako replay attack a man-in-the-middle, ale sú tu možné útoky na znovuvyvolanie relácie a zlyhanie služby. Kerberos má veľmi silné zabezpečenie avšak je možné ho ohroziť replay attackom v prípade, že nemáme zosynchronizovaný čas. Lístky majú platnosť väčšinou niekoľko hodín a v prípade odcudzenia počítača má dotyčný prístup do systému. U IPsecu je možný útok na protokol ESP alebo tzv. IV attack.

Žiaden protokol nie je ideálny a obsahuje nejaké nedostatky. Eliminovaním niektorých týchto nedostatkov je možné použiť kombináciu niektorých protokolov, napr. autentizáciu Kerberom u SSH, alebo zamedzenie replay attacku použitím IPsecu pri autentizácii Kerberom atď.

Najlepšie výsledky boli zistené pri SSH, ktoré zaťažovalo systém len minimálne a boli dosiahnuté maximálne prenosové rýchlosti. Protokol SSL bol na tom horšie, pretože pre svoj chod vyžaduje veľké množstvo potvrdzovacích správ a od toho sa odvíja aj zaťaženie systému a prenosová rýchlosť. Najhoršie dopadol IPsec, ktorý zaťažil systém najviac a boli dosiahnuté najhoršie prenosové rýchlosti. Kerberos slúžil len ako nástroj pre autentizáciu, preto nebolo možné zistiť jeho náročnosť na systémové zdroje.

Tab. 8.1: Komplexný prehľad vlastností protokolov

Protokol	Vrstva OSI	Autentizácia	Nebezpečenstvá	Zaťaženie
SSH	aplikačná	heslom, RSA kľúčom	replay attack, MITM, dictionary attack, brute force	nízke
SSL	prezentačná, transportná	certifikátom	replay attack, MITM, session renegotiation, DoS	vysoké
Kerberos	vyššie vrstvy	heslom, lístkom	replay attack	nezistené
IPsec	sieťová alebo žiadna	tajné kľúče	útok na ESP, IV attack	vysoké

9 ZÁVER

Témou tejto práce sú kryptografické protokoly, ich bezpečnosť a spôsob autentizácie a autorizácie v prostredí Open Source Software.

Prvá kapitola pojednáva o základoch kryptografie, kde vznikla a čo je jej úlohou.

Obsahom druhej kapitoly sú autentizácia, jej viaceré druhy, ktoré sú bližšie popísané, a autorizácia.

Samotné kryptografické protokoly sú obsiahnuté vo zvyšných kapitolách. Každý protokol má svoj vlastný rozbor, ktorý zahŕňa pôvod protokolu, jeho funkčnosť a bezpečnosť a praktické overenie.

Prvým protokolom v mojej práci je Secure Shell, ktorý slúži predovšetkým na vzdialené ovládanie iných počítačov. Je to veľmi rýchly a nenáročný protokol, čo som si overil aj v praxi.

Ďalším protokolom je komplexný protokol SSL. Jeho využitie sa nájde predovšetkým vo webových aplikáciách a samotnom webe. V tomto prípade sa nedá hovoriť o rýchlom protokole, pretože svojím množstvom správ značne zaťažuje systém, a od toho sa odvíja aj rýchlosť práce.

Zaujimavejším protokolom je protokol IPsec, ktorý je obsiahnutý v predposlednej kapitole. Na rozdiel od všetkých ostatných protokolov zabezpečuje celú komunikáciu a nie len nejakú aplikáciu alebo vrstvu referenčného modelu ISO/OSI. Jeho nevýhodou je náročnosť na systémové zdroje.

Posledná kapitola s kryptografickými protokolmi zahŕňa autentizačný protokol Kerberos. V tomto prípade ide iba o overenie identity užívateľa pomocou KDC. Súčasťou Kerbera je protokol Needham-Schroeder, ktorý ochraňuje jeho časové lístky proti replay attack útoku.

Súčasťou praktického overenia bolo sfunkčnenie systému na overenie každého protokolu, nasleduje postup, akým som sa k výsledkom dopracoval a samotné výsledky merania.

Na záver som uviedol stručný prehľad protokolov, vrstvy ISO/OSI modelu, súčasťou ktorých sú, ďalej je to spôsob autentizácie, bezpečnostné riziká a výkonové parametre.

LITERATÚRA

[1] BARRETT, Daniel, E. SILVERMAN, Richard. *SSH Kompletní průvodce. 1. vyd. Brno : Computer Press, 2003. 576 s. ISBN 80-7226-852-X.*

[2] DOSTÁLEK, Libor. *Velký průvodce protokoly TCP/IP: Bezpečnost. 2. aktualiz. vyd. Brno: Computer Press, a.s., 2003. 572 s. ISBN 80-7226-849-X.*

[3] KMEŤ, Peter. *KERBEROS: Čo? Prečo? Ako?* [online]. 2006 [cit. 2010-01-19]. Dostupný z WWW: `http://www.ms.mff.cuni.cz/~kmetp3am/kerb.html`

[4] PUŽMANOVÁ, Rita. *TCP/IP v kostce. 2. rozš. vyd. České Budějovice : Kopp, 2009. 620 s. ISBN 978-80-7232-388-3.*

[5] ASLAM, Mohamed. *How to fix Apache – "Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName" Error on Ubuntu* [online]. 17. 4. 2009 [cit. 2010-04-22]. How to fix Apache – "Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName" Error on Ubuntu. Dostupné z WWW: `http://www.linkon.cz/gm22a`

[6] CHAPPLE, Mike. *SearchSecurity.com* [online]. 2. 9. 2005 [cit. 2010-05-06]. An introduction to SSH2. Dostupné z WWW: `http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1052647,00.html`

[7] Hajný, Jan. *Úvod do kryptografie* [online]. [cit. 2010-04-17]. Dostupný z WWW: `http://www.pojdnatechniku.cz/images/documents/uvod_do_kryptografie.pdf`

[8] OCELIC, Davor. *SPINLOCKSOLUTIONS TECHPUB* [online]. 2006, 22. 3. 2010 [cit. 2010-05-09]. Debian GNU: Setting up MIT Kerberos 5. Dostupné z WWW: `http://techpubs.spinlocksolutions.com/dklar/kerberos.html#intro`

[9] PANAGIOTIS, Christias. *Unixhelp.ed.ac.uk* [online]. 1994 [cit. 2010-05-09]. UNIX man pages : ssh (1. Dostupné z WWW: `http://unixhelp.ed.ac.uk/CGI/man-cgi?ssh+1`

[10] RESCORLA, Eric. *SSH, SSL, and IPsec: wtf?* [online]. 6. 11. 2008 [cit. 2010-04-19]. Dostupné z WWW: `http://cseweb.ucsd.edu/classes/fa08/cse127/rescorla-comsec.pdf`

[11] ROSENKOETTER, Gabriel. *Re: [PLUG] ssh vs. ssh2* [online]. 2002 [cit. 2010-01-18]. Dostupný z WWW: http://lists.netisland.net/archives/plug/plug-2002-03/msg00425.html

[12] Visolve SSH Team. *Visolve* [online]. 28-01-02, 30-06-06 [cit. 2010-05-19]. OpenSSH & Kerberos. Dostupné z WWW: http://www.visolve.com/security/ssh_kerberos.php

[13] WHEELER, Evan. *Sans* [online]. 18. 9. 2008 [cit. 2010-04-18]. Replay Attacks. Dostupné z WWW: http://www.sans.org/security-resources/paper.php?cat=security_plus&id=replay_attack_sp08

[14] MIKE. *SSL on Ubuntu 8.10 Apache2* [online]. 3. 1. 2009 [cit. 2010-04-22]. SSL on Ubuntu Apache2 Server. Dostupné z WWW: http://beginlinux.com/blog/2009/01/ssl-on-ubuntu-810-apache2

[15] *Bezpečnostné protokoly IP (IPSec)* [online]. 2005 [cit. 2010-01-23]. Dostupný z WWW: http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzaja/rzajaahheader.htm

[16] *Employees.org* [online]. 9. 3. 2001 [cit. 2010-05-06]. Secure Shell FAQ Section 1: About Secure Shell:. Dostupné z WWW: http://www.employees.org/~satch/ssh/faq/ssh-faq-1.html

[17] *HMAC* [online]. 2009 , 1. 1. 2009 [cit. 2010-01-23]. Dostupný z WWW: http://sk.wikipedia.org/wiki/HMAC

[18] *ISAKMP, Internet Security Association and Key Management Protocol* [online]. 2009 [cit. 2010-01-24]. Dostupný z WWW: http://www.networksorcery.com/enp/protocol/isakmp.htm

[19] *Kerberos 5 Release 1.7* [online]. [cit. 2010-01-18]. Dostupný z WWW: http://web.mit.edu/Kerberos/krb5-1.7/

[20] *Kryptografie* [online]. 2009 , 8. 12. 2009 [cit. 2010-01-07]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Kryptografie

[21] *OpenSSH* [online]. 2009 [cit. 2010-03-22]. Dostupné z WWW: http://openssh.org/cs/index.html

[22] *PCmag* [online]. 2010 [cit. 2010-04-18]. Replay attack Definition from PC Magazine Encyclopedia. Dostupné z WWW: http://www.pcmag.com/encyclopedia_term/0,2542,t=replay+attack&i=50439,00.asp

[23] *Spamlaws* [online]. 2009 [cit. 2010-04-18]. What Is IP Spoofing and How Does It Work?. Dostupné z WWW: http://www.spamlaws.com/how-IP-spoofing-works.html

[24] *The H Security* [online]. 9. 3. 2010 [cit. 2010-03-21]. OpenSSH 5.4 couples standard local input with server ports - The H Security: News and Features. Dostupné z WWW: http://www.h-online.com/security/news/item/OpenSSH-5-4-couples-standard-local-input-with-server-ports-949963.html

[25] *TopBits.com* [online]. 2010 [cit. 2010-05-09]. Dictionary Attack. Dostupné z WWW: http://www.topbits.com/dictionary-attack.html

[26] *TopBits.com* [online]. 2010 [cit. 2010-05-09]. Brute Force Attack. Dostupné z WWW: http://www.topbits.com/brute-force-attack.html

[27] *TopBits.com* [online]. 2010 [cit. 2010-05-09]. Denial of Service (DoS) Attacks. Dostupné z WWW: http://www.topbits.com/denial-of-service-dos-attacks.html

[28] Ubuntu.com. *SSH - Ubuntu Česko* [online]. 2009 [cit. 2010-05-09]. SSH. Dostupné z WWW: http://wiki.ubuntu.cz/SSH

[29] *What Is IPSec?: Security Policy; Security Services* [online]. 2003 [cit. 2010-01-23]. Dostupný z WWW: http://technet.microsoft.com/en-us/library/cc776369%28WS.10%29.aspx

ZOZNAM SKRATIEK

3DES Triple DES

AES Advanced Encryption Standard

AH Authentication Header

CPU Central Processing Unit – procesor

DES Data Encryption Standard

DNS Domain Name System

DoS Denial of Service

DSA Digital Signature Algorithm

ESP Encapsulating Security Payload

GSM Global System for Mobile Communications

HMAC Keyed-hash Message Authentication Code

HTTPS Hypertext Transfer Protocol Secure

IKE Internet Key Exchange

IP Internet Protocol

IPsec secure Internet Protocol

ISAKMP Internet Security Association And Key Management Protocol

KDC Key Distribution Center

LAN Local Area Network

MITM Man-in-the-middle attack

MD Message-Digest

OSI Open Systems Interconnection

PC Personal Computer

PIN Personal Identification Number

RAM Random-Access Memory

RSA Rivest, Shamir, Adleman

SA Security Association

SSH Secure Shell

SSL Secure Socket Layer

TCP Transmission Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

VPN Virtual Private Network

ZOZNAM PRÍLOH

A SSL certifikát a jeho detaily **46**

B Kerberos **50**

A SSL CERTIFIKÁT A JEHO DETAILY

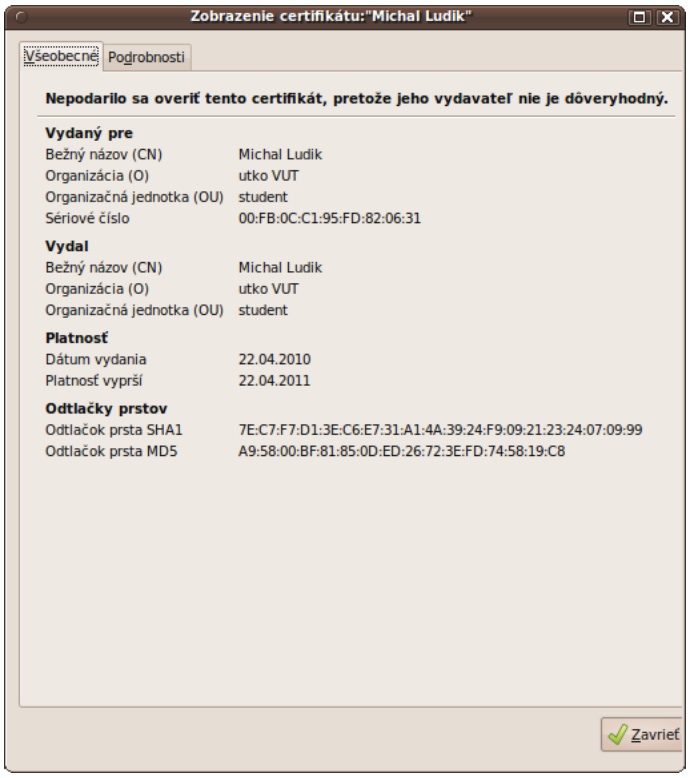
Textový výpis certifikátu na obr. A.1 a A.2 je nasledovný:

```
-Michal Ludik
   -Certifikát
      Verzia
        Verzia 1
      Sériové číslo
        00:FB:0C:C1:95:FD:82:06:31
      Algoritmus podpisu certifikátu
        PKCS #1 SHA-1 so šifrovaním RSA
      Vydavateľ
        E-mail (E) = xludik01@stud.feec.vutbr.cz
        Bežné meno (CN) = Michal Ludik
        Organizačná jednotka (OU) = student
        Organizácia (O) = utko VUT
        Miesto (L) = Brno
        Štát (ST) = Czech Republic
        Krajina (C) = CZ
      -Platnosť
        Neplatný pred
          22.04.2010 17:01:10
          22.04.2010 15:01:10 GMT
        Neplatný po
          22.04.2011 17:01:10
          22.04.2011 15:01:10 GMT
      Subjekt
        E-mail (E) = xludik01@stud.feec.vutbr.cz
        Bežné meno (CN) = Michal Ludik
        Organizačná jednotka (OU) = student
        Organizácia (O) = utko VUT
        Miesto (L) = Brno
        Štát (ST) = Czech Republic
        Krajina (C) = CZ
      -Informácie o verejnom kľúči subjektu
        Algoritmus verejného kľúča subjektu
          PKCS #1 šifrovanie RSA
```

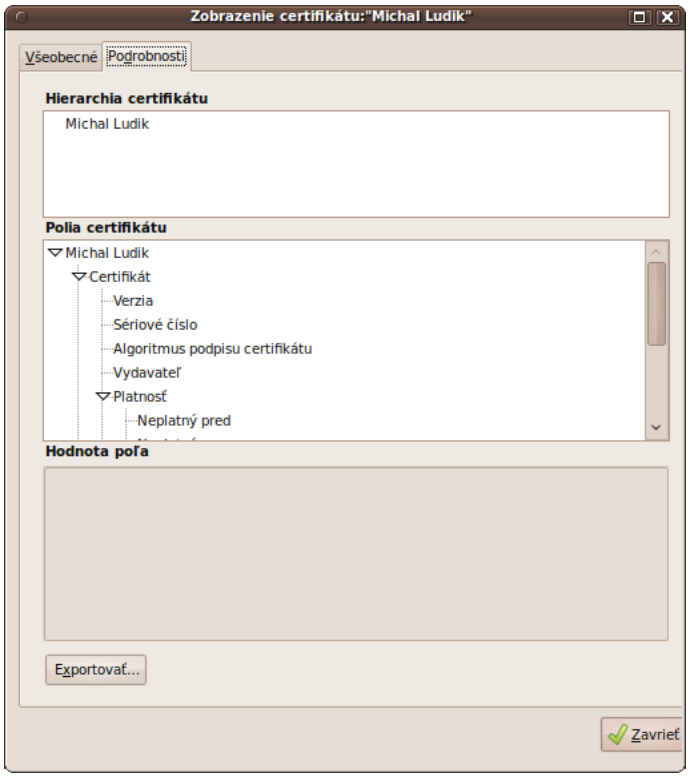
```
  Verejný kľúč subjektu
    Modulus (1024 bitov):
    af 2d 27 c0 fb a4 8a 4b 47 96 2e a4 52 6f d8 e2
    1a d8 16 12 9f ec e9 6c 74 b6 ef 53 83 b2 c9 98
    1f 9a 83 0b 7d d3 34 0e 17 14 60 70 0f 9a d0 b1
    ba 58 7d 8e 5a 6d fe d8 92 83 2f b7 e0 8c 8e 95
    92 43 8e 6a 89 69 f7 5b a5 6f b7 fd 17 6a 25 15
    d6 0b b5 5e c7 16 ce 27 5f 47 6a ca 10 b5 d4 e0
    37 ba d9 ca 5d 08 42 93 b1 b6 60 69 dc 6c c5 a6
    54 01 29 f1 77 31 cc 4f 93 db 07 ba b7 e4 7e 55

    Exponent (24 bitov):
    65537
Algoritmus podpisu certifikátu
  PKCS #1 SHA-1 so šifrovaním RSA
Hodnota podpisu certifikátu
  Veľkosť: 128 bajtov / 1024 bitov
  67 d6 d0 d2 35 28 7b 71 09 c4 75 1a 31 32 1a 3f
  08 5b ab 13 66 bc 07 eb c8 06 8c d9 09 a4 bd c9
  e7 82 60 97 6e b3 8b fb 59 d7 4b 26 bc d3 13 9c
  9f 38 e2 d4 8f 58 2d 6a fd 04 f8 05 3b 43 36 0f
  4a ce f4 49 23 9f 20 74 87 07 a1 bc 7c b1 61 0a
  99 7e 40 c0 07 50 5a 27 a1 30 f7 29 18 fe 90 45
  45 1e 66 f3 5f e7 30 bb 51 cf f5 26 1f cf 1c 4c
  45 f0 cf b7 cb 0d 1a f9 7b 41 42 cc a0 d3 33 17
```

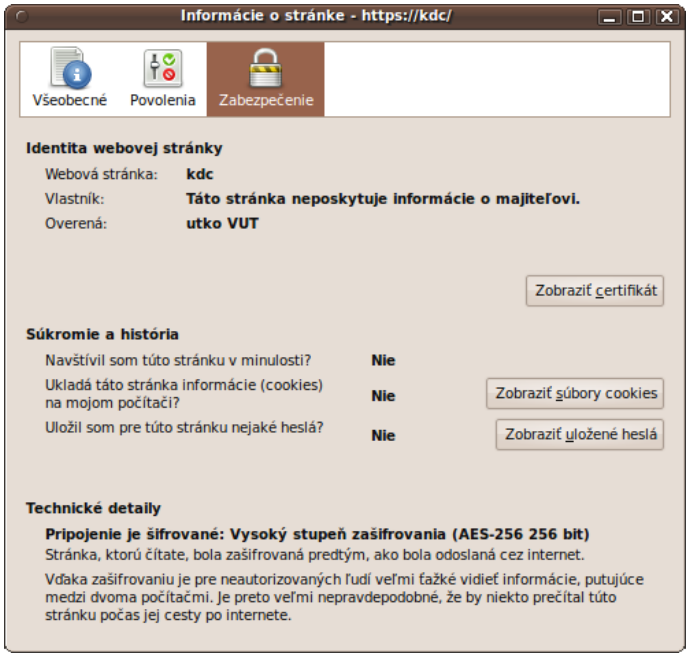
Na obr. A.4 sú zobrazené pakety zachytené Wiresharkom pri komunikácii.



Obr. A.1: Certifikát SSL (1)



Obr. A.2: Certifikát SSL (2)



Obr. A.3: Informácie o zabezpečenej stránke

No.	Time	Source	Destination	Protocol	Info
205	5.822976	147.229.196.145	147.229.148.151	TCP	56245 > https [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=1376262 TSER=0 WS=6
206	5.823319	147.229.148.151	147.229.196.145	TCP	https > 56245 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=26855444 TSER
207	5.823371	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=1376262 TSER=26855444
213	5.876995	147.229.196.145	147.229.148.151	SSL	Client Hello
214	5.877369	147.229.148.151	147.229.196.145	TCP	https > 56245 [ACK] Seq=1 Ack=163 Win=6912 Len=0 TSV=26855457 TSER=1376276
216	5.892405	147.229.148.151	147.229.196.145	TLSv1	Server Hello, Certificate, Server Key Exchange, Server Hello Done
217	5.892452	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=163 Ack=1179 Win=8768 Len=0 TSV=1376280 TSER=26855461
219	5.900358	147.229.196.145	147.229.148.151	TLSv1	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
223	5.911780	147.229.148.151	147.229.196.145	TLSv1	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message
226	5.951627	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=361 Ack=1445 Win=11136 Len=0 TSV=1376295 TSER=26855466
319	8.184606	147.229.196.145	147.229.148.151	TLSv1	Application Data
321	8.186203	147.229.148.151	147.229.196.145	TLSv1	Application Data, Application Data, Application Data, Application Data
322	8.186256	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=702 Ack=2057 Win=13504 Len=0 TSV=1376853 TSER=26856035
330	8.280533	147.229.196.145	147.229.148.151	TLSv1	Application Data
331	8.281758	147.229.148.151	147.229.196.145	TLSv1	Application Data, Application Data, Application Data, Application Data
332	8.281901	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=755 Ack=2493 Win=15808 Len=0 TSV=1376877 TSER=26856058
406	10.836235	147.229.196.145	147.229.148.151	TLSv1	Application Data
407	10.837435	147.229.148.151	147.229.196.145	TLSv1	Application Data, Application Data, Application Data, Application Data
408	10.837501	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=840 Ack=2865 Win=18176 Len=0 TSV=1377516 TSER=26856697
409	10.872951	147.229.196.145	147.229.148.151	TLSv1	Application Data
410	10.873871	147.229.148.151	147.229.196.145	TLSv1	Application Data, Application Data, Application Data, Application Data
413	10.910635	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=893 Ack=3029 Win=20544 Len=0 TSV=1377535 TSER=26856706
496	13.118728	147.229.196.145	147.229.148.151	TLSv1	Application Data
498	13.123258	147.229.148.151	147.229.196.145	TLSv1	Application Data,
499	13.123340	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=962 Ack=4477 Win=23424 Len=0 TSV=1378088 TSER=26857269
500	13.123377	147.229.148.151	147.229.196.145	TCP	[TCP segment of a reassembled PDU]
501	13.123393	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=962 Ack=5925 Win=26368 Len=0 TSV=1378088 TSER=26857269
502	13.123499	147.229.148.151	147.229.196.145	TCP	[TCP segment of a reassembled PDU]
503	13.123527	147.229.196.145	147.229.148.151	TCP	56245 > https [ACK] Seq=962 Ack=7373 Win=29248 Len=0 TSV=1378088 TSER=26857269

Obr. A.4: Prehľad paketov SSL zachytených wiresharkom

B KERBEROS

Na obr. B.1 je zobrazená komunikácia počítačov s autentizáciou pomocou Kerbera.

No.	Time	Source	Destination	Protocol	Info
465	11.89961	147.229.196.145	147.229.148.151	NTP	NTP client
466	11.89998	147.229.148.151	147.229.196.145	NTP	NTP server
3449	48.13037	147.229.196.145	147.229.148.151	KRB5	AS-REQ
3450	48.13145	147.229.148.151	147.229.196.145	KRB5	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
4313	52.05391	147.229.196.145	147.229.148.151	KRB5	AS-REQ
4315	52.11541	147.229.148.151	147.229.196.145	KRB5	AS-REP
4848	64.60564	147.229.196.145	147.229.148.151	TCP	56797 > ftp [SYN] Seq=0 Win=5840 Len=0 MSS=1460 T
4849	64.60599	147.229.148.151	147.229.196.145	TCP	ftp > 56797 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
4850	64.60604	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=
4853	64.67351	147.229.148.151	147.229.196.145	FTP	Response: 220 PA-274 FTP server (Version 5.60) re
4854	64.67368	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=1 Ack=46 Win=5888 Len=0 TSV
4855	64.67388	147.229.196.145	147.229.148.151	FTP	Request: AUTH GSSAPI
4856	64.67422	147.229.148.151	147.229.196.145	TCP	ftp > 56797 [ACK] Seq=46 Ack=14 Win=5824 Len=0 TS
4857	64.67427	147.229.148.151	147.229.196.145	FTP	Response: 334 Using authentication type GSSAPI; A
4858	64.67835	147.229.196.145	147.229.148.151	KRB5	TGS-REQ
4861	64.71109	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=14 Ack=102 Win=5888 Len=0 T
4864	64.75122	147.229.148.151	147.229.196.145	KRB5	TGS-REP
4867	64.75345	147.229.196.145	147.229.148.151	FTP	Request: ADAT YIICbQYJKoZIhvcSAQICAQBuggJcMIICWKA
4870	64.79215	147.229.148.151	147.229.196.145	TCP	ftp > 56797 [ACK] Seq=102 Ack=857 Win=7488 Len=0
4875	64.82795	147.229.148.151	147.229.196.145	FTP	Response: 235 ADAT=YIGZBgkqhkiG9xIBAgICAG+BiTCBhq
4876	64.82801	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=857 Ack=321 Win=6912 Len=0
5236	76.89961	147.229.196.145	147.229.148.151	NTP	NTP client
5237	76.90001	147.229.148.151	147.229.196.145	NTP	NTP server
15384	125.4113	147.229.196.145	147.229.148.151	FTP	Request: MIC BQQE/wAMAAAAAAAAD2mtw1VTRVIgbWljaGFs
15385	125.4117	147.229.148.151	147.229.196.145	TCP	ftp > 56797 [ACK] Seq=321 Ack=919 Win=7488 Len=0
15386	125.4316	147.229.148.151	147.229.196.145	FTP	Response: 631 BQQF/wAMAAAAAAAANMzi2jIzMiBHU1NBUEk
15387	125.4317	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=919 Ack=443 Win=6912 Len=0
15388	125.4320	147.229.196.145	147.229.148.151	FTP	Request: MIC BQQE/wAMAAAAAAAAD2mtxFBXRAAIbHithMQP
15389	125.4324	147.229.148.151	147.229.196.145	FTP	Response: 631 BQQF/wAMAAAAAAAANMzi2zI1NyAiL2hvbWU
15390	125.4325	147.229.196.145	147.229.148.151	FTP	Request: MIC BQQE/wAMAAAAAAAAD2mtxVNZU1QA0nvSLv/v
15391	125.4329	147.229.148.151	147.229.196.145	FTP	Response: 631 BQQF/wAMAAAAAAAANMzi3DIxNSBVTklYIFR
15393	125.4711	147.229.196.145	147.229.148.151	TCP	56797 > ftp [ACK] Seq=1019 Ack=607 Win=6912 Len=0
19936	140.8996	147.229.196.145	147.229.148.151	NTP	NTP client
19937	140.8999	147.229.148.151	147.229.196.145	NTP	NTP server

Obr. B.1: Prehľad paketov zachytených wiresharkom pri autentizácii Kerberom