**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**
**MATERIÁLOVOTECHNOLOGICKÁ FAKULTA V TRNAVE**
**Katedra Aplikovanej Informatiky a Automatizácie**

# NÁVRH, REALIZÁCIA A ZABEZPEČENIE FIREWALLOVÉHO SYSTÉMU

DIPLOMOVÁ PRÁCA
(web release)

**Bc. JAROSLAV IMRICH**

**TRNAVA 2006**

---

# ABSTRAKT

**Kľúčové slová: počítačová sieť, firewallový systém, GNU/Linux, bezpečnosť**

Študijné zameranie: Aplikovaná informatika a automatizácia v priemysle
Autor: Bc. Jaroslav Imrich
Diplomová práca: Návrh, realizácia a zabezpečenie firewallového systému
Vedúci diplomovej práce: Ing. Pavel Beňo
Odborný konzultant: Ing. Andrej Eliáš
Počet strán: 68

Úvodná časť práce obsahuje krátke oboznámenie s problematikou bezpečnosti počítačových sietí. Hneď za stručným prehľadom vybraných nebezpečenstiev, ktoré v nich hrozia nasleduje popis jednotlivých typov firewallových systémov. Osobitnú pozornosť v tejto kapitole venujem virtuálnym privátnym sieťam, pretože ich význam neustále rastie a začínajú prenikať aj do života bežných ľudí. Druhá kapitola je zhrnutím prevažnej časti mojich poznatkov o bezpečnosti unixových systémov a konkrétne sa zameriavam na popis systému GNU/Linux. Otázky súvisiace s bezpečnosťou tohto systému som rozdelil na tri skupiny. V prvej sa nachádzajú témy súvisiace s fyzickou bezpečnosťou systému. Vytvoril som konkrétny modelový útok, ktorému sa následne pokúšam zabrániť zavádzaním adekvátnych opatrení. Systémová bezpečnosť v mojom ponímaní zahŕňa pravidlá pre tvorbu silných hesiel, politiku pre inštaláciu nového a aktualizáciu starého softvéru a správnu konfiguráciu služieb. Je tu vysvetlené aj prečo je dobré, ak jednotlivé služby bežia iba s právami bežného neprivilegovaného používateľa. Do tretej skupiny partia otázky týkajúce sa sieťovej bezpečnosti. Vyzdvihujem najmä obmedzenie počtu spustených sieťových služieb, správne nastavenie hostového firewallu ale aj pokročilé techniky ako je napríklad port knocking alebo vytváranie honey potov. Tretia kapitola práce je zameraná na popis monitorovacích nástrojov pre Linux/UNIX všeobecne. Popísal som postupne viaceré oblasti, ktoré by určite mali byť monitorované nielen na firewallových systémoch ale na serverových systémoch všeobecne., či už sa jedná o monitorovanie hardvérových súčastí systému, sieťových aktivít alebo ďalších dôležitých oblastí. Štvrtá kapitola je akýmsi návrhom konkrétnych produktov, ktoré som vybral pre pripravovaný firewallový systém. V záverečných kapitolách opisujem vytvorený inštalačný program, ktorého hlavnou úlohou je skrátiť čas potrebný na implementáciu firewallového systému. Posledná kapitola je venovaná opisu dvoch konkrétnych sietí, v ktorých som firewallový systém pomocou môjho inštalátora implementoval. V závere sa zamýšľam nad možnými vylepšeniami a stanovujem ďalšie ciele pre vývoj jednotlivých nástrojov, ktoré vytvorený firewallový systém obsahuje.

# ABSTRACT

**Keywords: computer network, firewall system, GNU/Linux, security**

First part of this work contains short introduction to the computer network security. Description of different types of firewall systems follows right after the short overview of chosen dangers that occurs in networks. Special attention is paid to virtual private networks because of their growing significance in everyday life. Second chapter summarizes my knowledge about the security of unix systems and I focus on the GNU/Linux operating system. I divided security issues into three groups. In the first of them I analyze physical security of computer systems. Implementation of many restrictions should prevent the model attack. System security identifies rules for creation of strong passwords, rules for installation and update of software products and the right configuration of services. I try to explain why it is good to run services without the root privileges. The third group is all about network security, about reasons for decreasing amount of running services, right settings of host firewall rules and also advanced techniques as port knocking or creation of honey pots. The third chapter is focused on description of tools that are usually used to monitor GNU/Linux systems. I described many different areas that should be monitored not only on firewall systems but also on server systems in general. These are monitoring of hardware health, network resources and other important activities. Fourth chapter is proposition of concrete products that I have chosen for the firewall system. Last part of the work contains description of installation program I have created and the main goal of which is saving of time needed for implementation of firewall system. Last chapter is dedicated to description of two networks equipped with firewall system created by my installation program. Finally I think about possible improvements that could be made in the future development.

**OBSAH**

## ZOZNAM SKRATIEK

| | |
|---|---|
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol over SSL |
| ARP | Address Resolution Protocol |
| SSH | Secure Shell |
| VPN | Virtual Private Network |
| ISP | Internet Service Provider |
| DoS | Denial of Service |
| DDoS | Distributed Denial of Service |
| ISO/OSI | International Standard Organization's Open System Interconnect |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| EPROM | Erasable Programmable Read-Only Memory |
| NF | NetFilter |
| IP | Internet Protocol |
| PF | PacketFilter |
| CARP | Common Address Redundancy Protocol |
| MAC | Media Access Control |
| LAN | Local Area Network |
| GPRS | General Packet Radio Service |
| NAT | Network Address Translation |
| DMZ | Demilitarized Zone |
| IPsec | IP Security |

| | |
|---|---|
| SSL | Secure Sockets Layer |
| PPTP | Point-to-Point Tunneling Protocol |
| CD | Compact Disc |
| USB | Universal Serial Bus |
| CMOS | Complementary Metal-Oxide Semiconductor |
| SMS | Short Message Service |
| ICMP | Internet Control Message Protocol |
| ASCII | American Standard Code for Information Interchange |
| HTML | HyperText Markup Language |
| PGP | Pretty Good Privacy |
| GnuPG | Gnu Privacy Guard |
| IDS | Intrusion Detection System |
| SNMP | Simple Network Management Protocol |
| MIB | Management Information Base |
| I2C | Inter-Integrated Circuit |
| SMBus | System Management Bus |
| MD5 | Message-Digest Algorithm 5 |
| SHA-1 | Secure Hash Algorithm |
| NTP | Network Time Protocol |
| GPS | Global Positioning System |
| HIDS | Host-based Intrusion Detection System |
| NIDS | Network Intrusion Detection System |
| DNS | Domain Name Systom |
| DHCP | Dynamic Host Configuration Protocol |

| | |
|---|---|
| ISC | Internet Systems Consortium |
| SSL/TLS | Secure Sockets Layer / Transport Layer Security |
| FTP | File Transfer Protocol |
| VRRP | Virtual Router Redundancy Protocol |
| HSRP | Hot Standby Router Protocol |
| GSM | Global System for Mobile Communications |
| MRTG | Multi Router Traffic Grapher |
| BASE | Basic Analysis and Security Engine |
| SRBD | Systém Riadenia Bázy Dát |

# ÚVOD

Sieťový firewallový systém je neodmysliteľnou súčasťou každej modernej počítačovej siete. Vytvorenie takéhoto systému si vyžaduje nielen množstvo času, ale aj dostatok finančných prostriedkov a skúseností.

V teoretickej časti tejto práce sa zameriam primárne na popis bezpečnosti unixových systémov. S touto témou sú úzko späté nástroje na monitorovanie sieťovej prevádzky, log súborov, integrity súborového systému, systémy na detekciu prienikov a rôzne pokročilé techniky ako je napríklad otvorenie tzv. hluchých sieťových portov alebo vytváranie adaptívnych firewallov. Samozrejme sa nevyhnem ani opisu vybraných nebezpečenstiev číhajúcich v počítačových sieťach na akýkoľvek operačný systém.

Hlavný cieľ mojej diplomovej práce je zostaviť a v produkčnom prostredí nasadiť sieťový firewallový systém založený na distribúcii Slackware systému GNU/Linux. V tomto systéme však musí ostať zachovaná v čo najväčšej možnej miere kompatibilita s pôvodnou distribúciou, aby ho bolo možné aktualizovať bežným spôsobom pomocou pôvodných distribučných balíkov a aby bol jednoducho prispôsobiteľný budúcim požiadavkám.

Očakávam, že sa mi podarí vytvoriť inštalačný program, ktorý by zjednodušil proces inštalácie takéhoto systému a tým by výrazne skrátil čas, ktorý je na túto činnosť potrebný. Tento program by mal ponúknuť používateľovi možnosť výberu typu použitého firewallu, vygenerovať „hrubé“ skripty potrebné pre jeho zavedenie a pripraviť systém na vykonávanie všetkých činností, ktoré sa od neho očakávajú. Medzi podporovanými typmi firewallových systémov by určite nemala chýbať brána vykonávajúca preklad zdrojových adries, transparentný sieťový bridge a proxy server. Inštalačný program musí byť tiež schopný do systému nainštalovať vybrané nástroje používané na monitorovanie takýchto systémov, nástroje na vytváranie VPN sietí a nástroje na detekciu útokov a prienikov, ktoré prevezme z webovej stránky projektu alebo z ľubovoľného pripojeného pamäťového média. Samozrejme používateľ musí mať možnosť vybrať, ktoré zo spomínaných produktov budú nainštalované. Je dôležité uvedomiť si, že tento inštalačný sprievodca neumožní implementovať všetky spomínané technológie používateľovi, ktorý nemá potrebné teoretické znalosti. Skúsenému

používateľovi však niekoľkonásobne skráti čas potrebný na vytvorenie robustného firewallového systému.

Proces zabezpečenia tohto systému bude pozostávať najmä z vykonania správnej konfigurácie jednotlivých komponentov a tiež z implementácie nástrojov schopných detekovať pokusy o narušenie bezpečnosti samotného systému, ale aj siete ktorú chráni.

V systéme budú použité len nástroje a programy voľne dostupné prevažne pod licenciou GNU/GPL. Narozdiel od podobných komerčných ale aj open source produktov bude možné k navrhnutému firewallovému systému veľmi ľahko vytvoriť záložný systém, ktorý v prípade výpadku primárneho začne okamžite plniť jeho úlohy.

# 1 ZÁKLADNÉ PRINCÍPY ZABEZPEČOVANIA POČÍTAČOVÝCH SIETÍ

Pre niekoho je počítačová sieť nepostrádateľným pomocníkom v práci, pre niekoho zas len kopou káblov. Väčšina z nás si však neuvedomuje, že v nej môžu číhať mnohé nebezpečenstvá a že v počítačových sieťach neexistuje súkromie. Rovnako, ako sú niektorí jedinci schopní prepadnúť banku a odniesť z nej obrovské finančné čiastky, tak sú na svete ľudia schopní napadnúť počítačovú sieť a „odniesť“ z nej napríklad dokumenty obsahujúce duševné vlastníctvo iných ľudí. Každá banka má špecializovanú bezpečnostnú službu, ktorej úlohou je zabrániť prepadom, no nie všetky počítačové siete majú bezpečnostné prvky, ktorých cieľ je podobný.

Bohužiaľ častá chyba, ktorej sa mnohé organizácie pri správe počítačových sietí dopúšťajú sú chybne stanovené priority a s tým súvisiace následné nedostatočné prostriedky na rozvoj a skvalitnenie siete. Myslím si, že je to spôsobené najmä pasivitou správcov siete, ktorých povinnosťou by malo byť informovať vedenie o bezpečnostných rizikách a vypracovať sieťovú bezpečnostnú politiku organizácie. Je však nutné zabezpečiť, aby táto politika bola všetkými zamestnancami striktne dodržiavaná. To sa dá dosiahnuť jedine tak, že ju schváli vedenie organizácie a stanoví sankcie za jej nedodržiavanie.

## 1.1 Nebezpečenstvá v počítačových sieťach

Jedným z hlavných cieľov ochrany počítačovej siete je ochrana dát uložených na systémoch, ktoré sa v nej nachádzajú. Dáta je nutné chrániť pred znehodnotením, ktoré môže nastať v dôsledku ich úniku, pozmenenia alebo úplnej straty. No spolu s dátami je nutné chrániť aj samotné systémy vo vnútri siete, pretože môžu byť zneužité. Zneužiť ich môžu napríklad nespokojní súčasní i bývalí zamestnanci, konkurenti a samozrejme aj počítačový experti špecializujúci sa na túto činnosť tzv. crackeri.

Podobne ako v iných oblastiach aj v počítačovej bezpečnosti platí, že prevencia je lepšia než následná náprava škôd. Osobne zastávam názor, že v každej väčšej organizácii by mal pôsobiť expert na počítačovú a sieťovú bezpečnosť resp. aspoň človek, ktorý sa v tejto problematike dobre orientuje.

Zvýšenie bezpečnosti, či už počítačovej siete alebo akéhokoľvek iného systému,

nie je možné dosiahnuť použitím jedného všemocného prvku, ale neustálym zdokonaľovaním viacerých prvkov. Je známa teória budovania bezpečnostných bariér, ktorá pokladá každý jeden prvok zabezpečenia – bariéru – za prekonateľný. Ak chceme zvýšiť bezpečnosť, je treba vytvoriť čo najviac bezpečnostných bariér. Potenciálnemu útočníkovi sa môže podariť prekonať päť bariér a zastaviť ho môže šiesta. Neprekonateľná bariéra neexistuje, no na prekonanie všetkých vytvorených bariér bude útočník potrebovať stráviť v nepriateľskom tábore určitý čas. Tento čas nahráva človeku zodpovednému za bezpečnosť, pretože má väčšiu šancu spozorovať nepovolené aktivity a následne vykonať potrebné opatrenia.

Nutným predpokladom pre zabezpečovanie počítačovej siete je znalosť a presné definovanie hrozieb, ktorým môže byť vystavená. Zoznam hrozieb však nikdy nemôže byť úplný, preto je nutné sieť monitorovať a dohliadať na jej prevádzku. Podľa miesta odkiaľ hrozba pochádza som sa rozhodol niektoré známe hrozby rozdeliť na vnútorné a vonkajšie.

### 1.1.1 Vybrané vnútorné hrozby

Bezpečnostné riziká je nutné brať do úvahy už pri prvotnom návrhu počítačovej siete. Ak je to možné, všetky prístupové body k sieti by mali byť chránené firewallovými systémami. To znamená, že na sieti nesmie byť vytvorený prístupový bod k internetu, alebo akejkoľvek inej vonkajšej sieti, o ktorom by správca siete nevedel. Súčasťou bezpečnostnej politiky musí byť teda prísny zákaz zmien v sieti pre všetkých používateľov s výnimkou systémových administrátorov. To znamená najmä pre používateľov pracovných staníc zákaz pripájania modemu k akémukoľvek systému, ktorý je súčasťou siete. Jediný nekontrolovaný prístupový bod k lokálnej sieti môže narušiť celú bezpečnostnú politiku organizácie. [1]

Podobne prísne opatrenia by sa mali dodržiavať aj pre kabeláž v interiéri alebo exteriéri budovy. Existujúce voľne prístupné sieťové prípojky alebo lištami nechránené káble na verejne prístupných chodbách znamenajú obrovské riziko. Tieto nedostatky pomáha odstraňovať presná a stále aktualizovaná mapa siete a bezpečnostné audity.

Vo vnútornej sieti existuje aj riziko, že zamestnanci budú odpočúvať jednotlivé sieťové prenosy svojich kolegov v snahe získať väčšie oprávnenia alebo prístup ku kritickým firemným dátam. Obranou proti odpočúvaniu je používanie šifrovaných

aplikačných protokolov, teda nahradenie napríklad klasického HTTP *(HyperText Transfer Protocol)* protokolu šifrovanou alternatívou HTTPS *(HyperText Transfer Protocol over SSL)*, poprípade aj vytváranie bezpečnostných perimetrov.

Lokálne segmenty siete, v ktorých je prenos do značnej miery závislý na protokole ARP *(Address Resolution Protocol)* pracujúcom na linkovej vrstve, sú náchylné aj na ľahko vykonateľný útok tzv. ARP poisoning. Pri tomto útoku útočník predstiera falošnú identitu v snahe smerovať cez seba sieťové prenosy, ktoré by na switchovanej sieti k nemu za normálnych okolností nikdy nedorazili. Najjednoduchšou obranou voči tomu útoku sú najmä správne nakonfigurované manažovateľné switche, poprípade aj na údržbu náročnejšie statické ARP záznamy na kritických systémoch. Osobne zastávam názor, že každý systém v lokálnej sieti by mal obsahovať statický ARP záznam na bránu siete. V sieťach, kde sú všetky systémy pod správou špecializovanej skupiny technikov je implementácia jednoduchá. V sieťach, kde sa o svoj systém stará každý používateľ sám, je nutné organizovať odborné semináre, na ktorých sú používatelia oboznámení s možnými dopadmi tohto útoku a formou ochrany pred ním.

Za najväčšiu hrozbu v lokálnych sieťach považujem tzv. trójske kone a iné formy zlomyseľného kódu. Miera ich škodlivosti je často podceňovaná vďaka falošnému pocitu bezpečia, ktorý vytvára prítomnosť sieťového firewallového systému. Vhodne navrhnutý trójsky kôň však dokáže vytvoriť smerom z lokálnej siete von korektne nadviazané spojenie, ktoré akýkoľvek stavový firewall prepustí. Čo viac potrebuje externý útočník? Ak využije spôsob tunelovania podobný protokolu SSH *(Secure Shell)* alebo VPN *(Virtual Private Network)* sieťam, môže sa prostredníctvom z lokálnej siete iniciovaného spojenia pripojiť na systém, z ktorého spojenie pochádza. Ak vezmeme do úvahy, že trójske kone prevládajú na systémoch Windows, ktorých bezpečnostný model je oslabený vďaka tomu, že bežní používatelia väčšinou pracujú s privilégiami správcu systému, tak dopad jedného trójskeho koňa môže byť zdrvujúci pre celú lokálnu sieť. Preto je viac než nutné, na týchto systémoch nasadiť antivírovú ochranu a personálne firewally, ktoré informujú o pokusoch jednotlivých programov pristupovať k sieťovým prostriedkom. Ďalšou bezpečnostnou bariérou môže byť napríklad nasadenie proxy servera s accountingom, s použitím ktorého nemusí návrh trójskeho koňa rátať. Bezpečnostné perimetre, v ktorých sú oddelené kriticky dôležité systémy sú taktiež vhodným spôsobom posilnenia obrany.

### 1.1.2 Vybrané vonkajšie hrozby

Podobne ako pri vnútorných hrozbách aj pri vonkajších má odpočúvanie spojení svoje čestné miesto. Odpočúvanie môže prebiehať buď priamo u ISP *(Internet Service Provider)* alebo aj na vzdialenejších miestach. Je nutné si uvedomiť, že každá komunikácia má dve strany a ak je jedna z nich vystavená určitému riziku, toto riziko má dopad aj na druhú stranu. Ak teda ktorýkoľvek z bodov, ktorými naša komunikácia prechádza, považujeme za nebezpečný, je nutné prijať dodatočné bezpečnostné opatrenia resp. vyhnúť sa prenášaniu kritických dát cez tieto miesta. Podobne ako pri lokálnych sieťach aj v externej komunikácii je vhodné uprednostňovať šifrované aplikačné protokoly.

Jednou zo známejších aj keď často nesprávne chápaných hrozieb sú otvorené sieťové porty. Aby mohol byť sieťový port otvorený, musí na hostiteľskom systéme bežať nejaká sieťová služba, ktorá ho používa. Ak by boli všetky systémy v lokálnej sieti správne nakonfigurované a teda by nemali vďaka nepotrebným sieťovým službám zbytočne otvorené sieťové porty, kleslo by riziko s nimi spojené na minimum. Keďže skutočnosť je iná, je nutné obmedzovať prístup z externých sietí len na vybrané služby na vybraných systémoch. Túto filtráciu prichádzajúcich sieťových spojení najčastejšie vykonávajú rôzne sieťové firewallové systémy, ktoré však okrem toho zabezpečujú aj množstvo iných funkcií. Za spomenutie určite stojí aspoň kontrola stavov spojení, monitorovanie prenášaných dát či rôzna miera detekcie útokov.

Menej častou, no nie zanedbateľnou formou útoku z externých sietí je DoS *(Denial of Service)* útok, ktorého cieľom je „zahltiť“ dostupnú kapacitu linky a tak sieti alebo konkrétnemu systému odoprieť prístup k sieťovým prostriedkom. Ak tento útok prichádza z jedného konkrétneho externého systému, dá sa mu zamedziť napríklad oznámením tejto skutočnosti svojmu ISP, ktorý všetku komunikáciu prichádzajúcu z inkriminovaného systému zablokuje. Ak sa však jedná o DDoS *(Distributed Denial of Service)* útok a nežiadúca komunikácia prichádza z množstva externých systémov z rôznych externých sietí, býva to často nerovný boj. Bojovať proti DoS útokom sa dá napríklad na bráne lokálnej siete, čo je ale väčšinou málo platné, keďže nevyžiadaná sieťová komunikácia vyťažuje linku medzi ISP a bránou lokálnej siete. Preto je nutné kontaktovať ISP a požiadať ho o pomoc. Tiež je vhodné o prebiehajúcom útoku informovať správcov sietí odkiaľ jednotlivé časti útoku prichádzajú. Ak sú však v útoku

zapojené desiatky alebo stovky sietí, je to beh na dlhú trať.

## 1.2 Firewallový systém

Najčastejším postupom pri zabezpečovaní siete je nasadenie sieťového firewallového systému, ktorý dokáže znížiť mnohé riziká na únosnú mieru, no určite nie je univerzálnym všeliekom.

Pod slovom firewall si každý používateľ výpočtovej techniky predstaví niečo iné. Niekto si toto slovo asociuje s personálnym firewallom pre systém Microsoft Windows, niekto si predstaví počítač chrániaci celú sieť a niekto len malú krabičku podobnú na switch, ktorá oddeľuje vnútornú sieť od vonkajšej. Pre tých menej znalých je firewall akýmsi mýtom, ktorý poskytuje absolútnu ochranu a spájajú si ho s obrázkom múriku postaveného z červených tehličiek.

Zostaviť presnú definíciu je však neľahká úloha. Vo všeobecnosti môžeme povedať, že firewall je metóda ochrany počítačov a počítačových sietí, spojených s inými počítačmi a sieťami, proti útokom zvonku a zvnútra. Pojem firewall je tiež možné použiť na opis rôznych sieťových konfigurácií zostavených pre tento účel. Firewall, je ale aj označenie pre tzv. „paketové filtre“, ktoré sú umiestnené medzi dvoma alebo viacerými počítačmi, alebo celými počítačovými sieťami, a filtrujú pakety podľa súboru pravidiel zostaveného osobami zodpovednými za sieťovú bezpečnosť.

V tejto práci budem pojmom firewallový systém označovať počítač s operačným systémom GNU/Linux, ktorý obsahuje viacero sieťových rozhraní a kontroluje a filtruje dáta prechádzajúce medzi nimi.

Hlavnou úlohou firewallového systému je chrániť vnútornú sieť pred nežiadúcimi spojeniami prichádzajúcimi z vonkajšej siete, ale aj chrániť vonkajšiu sieť pred útokmi z vnútornej. Bežné firewallové systémy sú schopné filtrovať sieťovú prevádzku na tretej a štvrtej vrstve modelu ISO/OSI *(International Standard Organization's Open System Interconnect)*, čiže na základe zdrojových a cieľových adries a podľa zdrojových a cieľových portov transportných protokolov TCP *(Transmission Control Protocol)* a UDP *(User Datagram Protocol)*. Niektoré implementácie firewallov dokážu však nahradiť aj funkciu manažovateľných switchov a filtrovať aj na druhej vrstve ISO/OSI.

### 1.2.1 Softvérový a hardvérový firewall

V dostupnej literatúre sa často firewally rozdeľujú na hardvérové a softvérové. Pojmom hardvérový firewall by podľa týchto publikácií malo byť označované zariadenie podobné klasickému sieťovému switchu, ktoré umožňuje filtrovať sieťové pakety na základe definovaných pravidiel. Pojmom softvérový firewall zas autori označujú implementáciu filtrovacieho systému v rôznych operačných systémoch. Ja som sa s týmto rozdelením nikdy nedokázal stotožniť a moje pochybnosti o správnosti tohto rozdelenia potvrdil vo svojej knihe poľský autor Jacek Artymiak, ktorý napísal, že nič také, ako softvérový a hardvérový firewall neexistuje. Všetky tieto zariadenia sú totiž podľa neho len paketové filtre pracujúce vďaka hardvéru a softvéru. To, či je softvér uložený v pamäti EPROM *(Erasable Programmable Read-Only Memory)* uzavretej v peknej krabičke z umelej hmoty, alebo na pevnom disku počítača triedy PC, je úplne jedno. Hardvér samotný by nikdy nedokázal plniť úlohu firewallu, keby nebolo softvéru, ktorý ho ovláda. [2]

Osobne považujem za vhodnejšie používať ako firewallový systém osobný alebo serverový počítač pracujúci so systémom GNU/Linux alebo OpenBSD, pretože toto riešenie poskytuje omnoho väčšie možnosti, než ktorékoľvek mne známe špecializované zariadenie.

### 1.2.2 Typy firewallov

Firewallový systém môže pracovať ako:

- Nestavový paketový filter;
- Stavový paketový filter;
- Proxy server.

Nestavový paketový filter je najjednoduchším typom firewallu a je schopný filtrovať sieťovú prevádzku len na základe zdrojových a cieľových adries či portov. Nie je schopný rozlíšiť, či prichádzajúce pakety patria k regulérne začatým spojeniam, pretože nijako nezohľadňuje stavy spojení. Takýto firewall sa väčšinou nezaobíde bez toho, aby pre každé pravidlo existovalo aj tzv. protipravidlo. V praxi to znamená, že ak napríklad z vnútornej časti siete povoľujeme spojenie na cieľový port 80/TCP, musíme napísať aj protipravidlo, ktoré povolí spojenie do vnútornej siete zo zdrojového portu

80/TCP. Ak by sa teda podarilo vzdialenému útočníkovi zasielať ľubovoľné pakety zo zdrojového portu 80/TCP, firewall by ich teoreticky mohol prepustiť.

Túto „zlú vlastnosť“ odstraňuje druhý typ firewallového systému tzv. stavový paketový filter, ktorý dokáže kontrolovať či prichádzajúce pakety patria k niektorému z regulérnych spojení. Aby to mohol vykonávať, musí si udržiavať stále aktuálne informácie o prebiehajúcich spojeniach. Zjednodušene môžeme povedať, že používa tabuľku, do ktorej zapisuje informácie o stave každého jedného sieťového spojenia. Teda odpadá možnosť, že by prepustil pakety zo zdrojového portu 80/TCP bez toho, aby naň bolo tesne predtým nadviazané spojenie. Výhoda takéhoto postupu spočíva v tom, že nemusí pre každé pravidlo existovať protipravidlo, ale stačí jediné pravidlo hovoriace o akceptovaní paketov patriacich k nadviazaným spojeniam. Udržiavanie informácií o prebiehajúcich spojeniach sa však oproti nestavovým paketovým filtrom výraznou mierou odzrkadľuje na nárokoch na operačnú pamäť systému. Ak jej nebude dostatok, môže sa stať, že firewallový systém prestane obsluhovať nové spojenia do doby, kým sa nejaká pamäť neuvoľní. Na stanovenie potrebnej veľkosti operačnej pamäte však neexistuje nijaký vzorec, preto je nutné systém monitorovať a určiť ju na základe empirických poznatkov.

Proxy server je špeciálnym druhom firewallového systému, ktorý poskytuje klientom nepriamy prístup k vybraným sieťovým službám. Klientský systém sa pripája na proxy server a žiada o zdroje z iného servera. Proxy server mu ich sprístupní buď tak, že sa pripojí na určený server ako klient a získa ich z neho, alebo ich poskytne z vyrovnávacej pamäte (angl. cache). Klientský systém teda nemá priamy prístup k vonkajším zdrojom, ale poskytuje mu ich proxy server, ktorý je pre vonkajšie systémy v úlohe klienta. Toto umiestnenie proxy serveru medzi klienta a server, rovnako ako aj skutočnosť, že dokáže pracovať na aplikačnej vrstve, ho priamo predurčujú na to, aby našiel uplatnenie v sieťach, ktoré vyžadujú najvyšší stupeň zabezpečenia. Daňou za možnosť filtrovať na aplikačnej vrstve je však v porovnaní s predchádzajúcimi typmi firewallových systémov vyššia hardvérová náročnosť.

### 1.2.3 Filtrovacie systémy v jednotlivých operačných systémoch

Systémy pracujúce s jadrom Linux využívajú na filtrovanie paketov jeho časť zvanú netfilter (ďalej len NF). Konfigurácia NF sa vykonáva pomocou obslužných

programov ipchains alebo iptables. Ipchains sú k dispozícii pre jadrá vetvy 2.2 a pre jadrá 2.4 a 2.6 je možné použiť aj iptables. Nie je ich však možné kombinovať, čo znamená, že buď vytvárame firewall iba s iptables alebo iba s ipchains. Syntax oboch programov je veľmi podobná, avšak významný rozdiel medzi nimi je v tom, že iptables spolu s jadrami 2.4 a 2.6 poskytujú možnosti stavového firewallu. NF poskytuje vymoženosti, ktoré by sme v iných filtrovacích systémoch hľadali márne. Spomenúť si určite zaslúži aspoň možnosť vytvárať pravidlá pokrývajúce viacero vrstiev ISO/OSI naraz, limitovanie počtu paketov v čase a dynamické vytváranie zoznamu IP *(Internet Protocol)* adries a porovnávanie IP adries z prichádzajúcich spojení voči nemu, čo využíva napríklad tzv. technika port knockingu. Ďalšou nespornou výhodou NF je, že je súčasťou jadra Linux a teda v prípade objavenia chyby je oprava v krátkom čase dostupná vo forme patchu pre jadro. Pre NF existujú aj mnohé zatiaľ však neoficiálne moduly, ktoré mu dokonca umožňujú filtráciu na aplikačnej vrstve. Vďaka tomu je možné efektívne a jednoducho filtrovať napríklad spojenia peer-to-peer sietí.

V systémoch BSD je v súčasnosti najpoužívanejší filtrovací systém PacketFilter (ďalej len PF), ktorý bol vyvinutý ako súčasť systému OpenBSD a následne portovaný do systémov FreeBSD a NetBSD. Tento filtrovací systém ponúka také vymoženosti ako synproxy, logovanie v binárnom formáte čitateľnom utilitou tcpdump, autentifikáciu pre gateway s využitím SSH, alebo jednoduchú správu a rozdeľovanie šírky pásma. Takmer všetko, čo ponúka PF, dokáže skúsený používateľ implementovať aj s linuxovým NF. Správa pravidiel s PF je však oveľa prehľadnejšia, pretože pravidlá, ktoré sa na linuxe definujú s použitím cyklov, sa v syntaxy PF definujú jednoducho s využitím zložených zátvoriek. To, čím sa tento filtrovací systém výrazne odlišuje od ostatných, je natívna podpora protokolu CARP *(Common Address Redundancy Protocol)* a technológia synchronizácie stavových tabuliek medzi hlavným a redundantným firewallom zvaná pfsync. Proti PF však v určitých prípadoch hovorí nemožnosť vytvárať pravidlá pokrývajúce viacero vrstiev ISO/OSI naraz. Ak napríklad chceme porovnávať zdrojovú MAC *(Media Access Control)* adresu so zdrojovou IP adresou, nie je na to možné použiť jedno pravidlo ako u linuxového NF. PF totiž nie je schopný pracovať na inej než sieťovej a transportnej vrstve protokolu TCP/IP. Tento sa problém sa dá riešiť pomocou tzv. tagovania paketov, keď na linkovej vrstve pracujúcou utilitou brconfig nastavíme tag paketu. PF dokáže tento tag prečítať a podľa neho pakety identifikovať. Tento postup však zdvojnásobuje hardvérovú náročnosť filtrovania.

Operačné systémy od firmy Microsoft v súčasnosti nie sú vybavené filtrovacím systémom, ktorý by bol použiteľný na vytvorenie sieťového firewallového systému. Existujú však špecializované produkty, ktoré sa tento nedostatok snažia odstrániť. Medzi ne patrí napríklad Kerio WinRoute Firewall. Z vlastnej skúsenosti však viem, že akákoľvek implementácia stavového firewallu na systéme Windows sa výraznou mierou odráža na výkonnosti sieťového rozhrania. Proti použitiu týchto produktov hovorí aj ich cena. Ročná licencia Kerio WinRoute Firewall 6 schopného chrániť 10 používateľov (10 hostov) stojí 399 USD a upgrady a patche na ďalší rok 99 USD. Licencia pre ďalších 100 používateľov stojí 1399 USD a upgrady na ďalší rok 399 USD. Ak uvažujeme o ochrane siete s 200 počítačmi, tak cena licencie na tri roky prekročí 100.000 Sk. Okrem toho, systém Windows sa na plnenie funkcie sieťového firewallu absolútne nehodí, pretože čas od nájdenia chyby po vydanie záplaty je oproti open source systémom abnormálne dlhý.

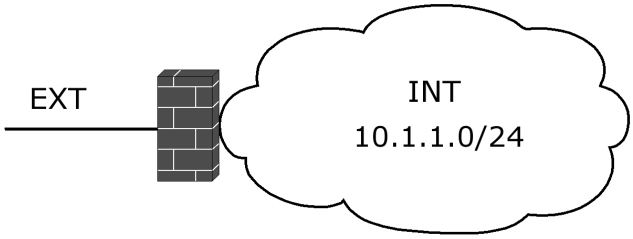
## 1.3 Firewallový systém ako súčasť štruktúry siete

Ani ten najlepší firewallový systém nedokáže správne plniť svoju obrannú funkciu, ak mu nie je náležite prispôsobená štruktúra siete. Firewallový systém je účinný, len ak má pod kontrolou všetky možné prístupové body k vonkajším sieťam resp. ak sú všetky prístupové body k LAN *(Local Area Network)* chránené firewallom. Ak si niektorý z účastníkov chránenej siete nainštaluje na nesprávne nakonfigurovaný systém pripojenie k internetu napríklad prostredníctvom GPRS *(General Packet Radio Service)*, vystavuje tým nebezpečenstvu celú LAN. V popise jednotlivých možností umiestnenia firewallového systému v štruktúre siete zámerne neuvádzam personálne alebo hostové firewally. Ich používanie totiž považujem za úplnú samozrejmosť.

### 1.3.1 Jednoduchá brána

Vhodné miesto pre nasadenie firewallového systému je prístupový bod siete. Príklad je znázornený na obrázku č.1. Takto umiestnený firewall je schopný zabezpečovať kontrolu komunikácie pre každý jeden prvok v sieti, či už sa jedná o komunikáciu z lokálnej siete smerom von, alebo naopak. Počítač s operačným systémom GNU/Linux alebo BSD vybavený vhodnými sieťovými rozhraniami je priam predurčený, aby plnil úlohu jednoduchej brány siete (angl. gateway), zvanej tiež

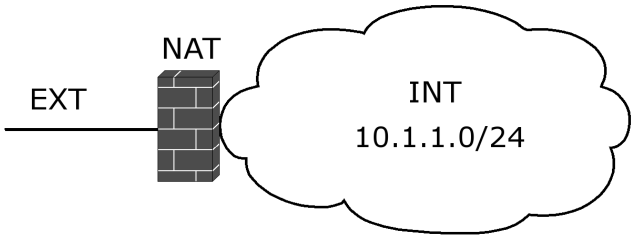
smerovač (angl. router). Minimálna konfigurácia tohto systému pozostáva z vhodne nakonfigurovaných sieťových rozhraní, povoleného preposielania IP paketov medzi nimi a zavedení pravidiel pre filtrovací systém tohto počítača. Samozrejme tu môžu bežať systémy na detekciu prieniku alebo rôzne iné monitorovacie nástroje.




*Obr.1: Firewallový systém ako jednoduchá brána*

### 1.3.2 Brána s NAT

NAT (Network Address Translation) je anglickým názvom pre činnosť, pri ktorej sa prepisujú zdrojové alebo cieľové adresy v IP paketoch pri ich prechode routerom alebo firewallovým systémom, ktorého umiestnenie je znázornené na obrázku č. 2. Väčšina systémov používa NAT preto, aby umožnili viacerým počítačom alebo celej privátnej sieti pristupovať k prostriedkom internetu prostredníctvom jedinej verejnej IP adresy. Táto technológia sa však dá použiť aj na jednoduché rozkladanie záťaže (angl. load balancing) medzi viacero serverov alebo na zabezpečenie dostupnosti služby, kedy router s NAT overuje dostupnosť vnútorného servera a v prípade jeho výpadku smeruje požiadavky na záložný server.



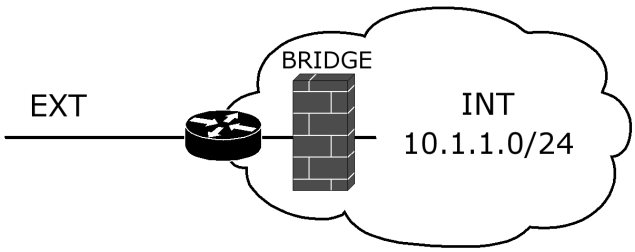

*Obr.2: Firewallový systém ako brána s NAT*

Ak pomocou NAT meníme v paketoch zdrojovú adresu, hovoríme o preklade zdrojových adries (angl. source NAT). Ak meníme cieľovú adresu jedná sa o preklad cieľových adries (angl. destination NAT) alebo tiež presmerovanie portov (angl. port forwarding). Pri preklade zdrojových adries sa v paketoch smerujúcich z lokálnej siete von mení zdrojová adresa na adresu vonkajšieho rozhrania firewallového systému, čím

sa vykonáva tzv. maškaráda siete (angl. masquerading). Preklad cieľových adries je možné použiť, ak chceme sieťové služby niektorého počítača z lokálnej siete sprístupniť vonkajším systémom. Vonkajšie systémy smerujú požiadavky na IP adresu vonkajšieho rozhrania firewallového systému a ten mení cieľovú IP adresu na adresu vybratého vnútorného systému. Toto je jediná možnosť ako sprístupniť služby z vnútornej siete vonkajším systémom. NAT teda aktívne zvyšuje bezpečnosť vnútorných systémov, pretože vonkajším systémom znemožňuje pripájanie na vnútorné systémy. To niektorých správcov sietí vedie k rozhodnutiu nasadiť NAT aj v prípade, že majú k dispozícii dostatok verejných IP adries. Vytvárajú tak konfiguráciu, v ktorej každej vnútornej adrese prislúcha jedna externá adresa a teda maskujú vnútornú sieť fiktívnou vonkajšou sieťou.

### 1.3.3 Ethernet bridge

Ethernet Bridge sa využíva na spájanie dvoch alebo viacerých segmentov lokálnej siete na linkovej vrstve, a preto je možné použiť ho iba v rámci LAN. Funguje však úplne transparentne a dokonca pre svoju činnosť nepotrebuje ani IP adresu, preto je často vhodným riešením pri vytváraní bezpečnostných perimetrov v lokálnej sieti. Často býva umiestnený aj priamo za smerovačom ako to znázorňuje obrázok č. 3.




*Obr.3: Firewallový systém ako ethernet bridge*

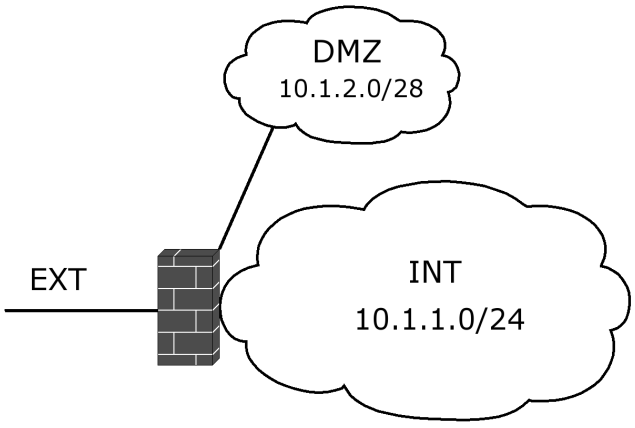
Ethernet bridge je taktiež ideálnym miestom pre nasadenie monitorovacích nástrojov. Linux kernel vo verzii 2.4 sa vyznačuje tým, že NetFilter bez aplikovania neoficiálneho patchu s názvom bridge-nf nie je schopný filtrovať sieťovú prevádzku prechádzajúcu cez sieťový bridge. Netfilter z jadra verzie 2.6 už túto vlastnosť obsahuje a navyše obsahuje aj nový physdev modul, ktorý výrazne uľahčuje vytváranie filtrovacích pravidiel, pretože umožní špecifikovať vstupné a výstupné sieťové rozhrania, ktorými sieťové pakety prechádzajú.

V prípade výpadku firewallového systému nasadeného ako ethernet bridge nie je nutné vykonať nijaké zmeny v nastaveniach klientských systémov, ale stačí len jednotlivé segmenty prepojiť pomocou bežného switchu. Samozrejme bežný switch nie je schopný filtrovať sieťovú prevádzku.

### 1.3.4 Demilitarizovaná zóna

Demilitarizovaná zóna (ďalej len DMZ) je oddelená časť vnútornej siete, na ktorú sa zvyčajne vzťahujú špeciálne filtrovacie pravidlá. Spojenia z lokálnej a vonkajšej siete sú do DMZ povolené, ale systémy z DMZ môžu zahajovať spojenia len do vonkajšej siete. Preto v DMZ bývajú oddelené najmä systémy poskytujúce sieťové služby, ktoré musia byť dostupné z externých sietí. Ak útočník získa kontrolu nad niektorým systémom v DMZ, nepripojí sa z neho na žiadny systém z lokálnej siete. Takéto rozdelenie siete znamená zníženie ničivého dopadu v prípade úspešného útoku voči niektorému zo systémov, ktoré poskytujú služby externým systémom. Jediným terčom pre útočníka ostávajú ostatné systémy v DMZ, ktoré sú ale za bežných okolností chránené hostovým firewallom. DMZ aktívne zvyšuje bezpečnosť systémov z lokálnej siete a je osvedčeným prvkom pri budovaní bezpečnostných bariér.

DMZ môže byť realizovaná napríklad pridaním ďalšieho sieťového rozhrania do firewallového systému (Obr.4), alebo správnym sériovým umiestnením dvoch firewallových systémov. Druhá spomínaná možnosť sa najčastejšie realizuje tak, že za prvým firewallovým systémom sú umiestnené systémy patriace do DMZ a až za nimi je zapojený druhý firewallový systém a lokálna sieť.




*Obr.4: Firewallový systém s demilitarizovanou zónou*

### 1.3.5 Virtuálne privátne siete

Virtuálna privátna sieť (ďalej len VPN) je súkromná sieť zvyčajne využívaná medzi jednotlivými pobočkami väčšej spoločnosti, alebo medzi viacerými spoločnosťami a organizáciami na komunikáciu prostredníctvom verejných sietí. Mnohé implementácie VPN vďaka autentifikácii používateľov, kontrole integrity dát a použitiu protokolov schopných vytvárať šifrované tunely v mnohých prípadoch poskytujú dostatočný stupeň ochrany prenášaných dát. Dobre navrhnutá VPN sieť môže byť použitá na prenášanie dôveryhodných dát verejnými sieťami.

Vytvorenie VPN siete môžeme zveriť svojmu ISP, čo znamená stále mesačné poplatky za údržbu siete. Taktiež vykonanie bezpečnostného auditu na zariadeniach používaných ISP sa stáva prakticky nemožné. Nemôžeme si teda byť celkom istí kvalitou poskytovanej siete a musíme sa spoliehať na svojho ISP. Existuje však aj druhá možnosť, ktorá znamená odbornú implementáciu niektorej z dostupných technológií na našich zariadeniach.

VPN je možné vytvoriť napríklad pomocou technológií:

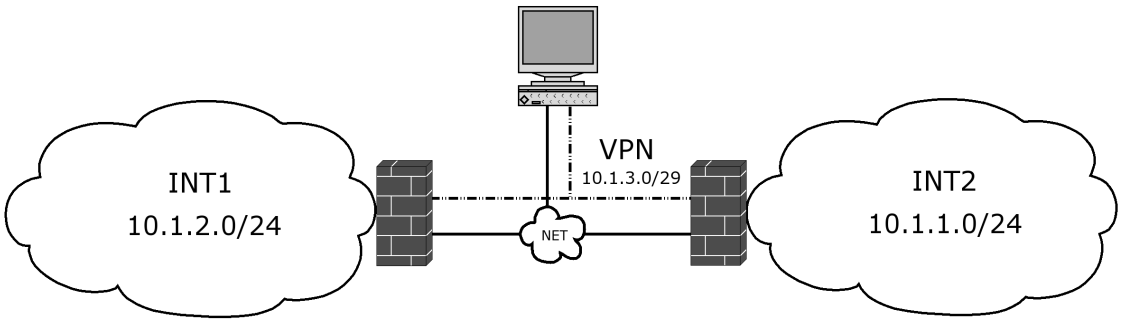
- IPsec (IP Security);
- SSL (Secure Sockets Layer);
- PPTP (Point-to-Point Tunneling Protocol).

Sada protokolov IPsec je súčasťou štandardu IPv6 a jej hlavným poslaním je vytvoriť bezpečnú alternatívu k v súčasnosti používanému sieťovému protokolu IP. Implementácia VPN sietí s protokolmi IPsec si vyžaduje, aby systém využívajúci tieto protokoly obsahoval ich podporu priamo vo svojom jadre a zatiaľ je to náročná úloha, ktorú by mal vykonávať odborník. Komunikácia s protokolmi IPsec ale môže znamenať pre niektoré sieťové prvky problém a preto s ich nasadením môžu byť spojené ďalšie investície.

Naproti tomu implementácia technológií využívajúcich algoritmy knižníc SSL je oveľa jednoduchšia. Patrí medzi ne napríklad open source riešenie s názvom OpenVPN, ktoré šifruje sieťové pakety určené na prenos VPN sieťou, opäť ich obalí sieťovou hlavičkou a až potom ich pošle cez verejnú sieť, k druhému koncu VPN siete, ktorý tento postup zopakuje obrátene. VPN siete založené na SSL sú preto relatívne menej výkonné.

Protokol PPTP bol prvýkrát použitý firmou Cisco, ktorá neskôr poskytla licenciu firme Microsoft. Je veľmi jednoducho konfigurovateľný, a preto ho používajú všetky implementácie VPN sietí v operačných systémoch firmy Microsoft. Kvôli známym obmedzeniam tohto protokolu sa očakáva logický prechod Microsoftu k dokonalejšej technológii. Možnou alternatívou je aj už spomínaný IPsec, proti ktorému však hrá nutnosť podpory v jadre operačného systému. Staršie systémy ako Windows 98 a ME ho totiž nepodporujú.[3]

Brána do VPN siete býva často umiestnená priamo na firewallovom systéme organizácie. Firewallové systémy jednotlivých pobočiek sú nakonfigurované tak, aby medzi sebou navzájom komunikovali prostredníctvom bezpečnej siete VPN (Obr.5). Takáto konfigurácia je pre klientské systémy úplne transparentná a zamestnanci vnímajú druhú geograficky vzdialenú pobočku ako súčasť lokálnej siete. VPN brána však môže poskytovať prístup k lokálnej sieti aj zamestnancom, ktorí sú nútení pracovať v teréne. Znamená to ale značné bezpečnostné riziko, a preto musia títo zamestnanci podstúpiť školenie o bezpečnosti VPN siete a o jej správnom používaní. Taktiež musia striktne dodržiavať pravidlá bezpečnostnej politiky organizácie, pretože strata alebo odcudzenie notebooku môžu znamenať neoprávnený vstup do firemnej siete.




*Obr.5: Firewallové systémy s virtuálnou privátnou sieťou*

# 2 BEZPEČNOSŤ LINUX/UNIX SYSTÉMOV

Úroveň zabezpečenia operačných systémov nainštalovaných na jednotlivých staniciach v sieti výrazne ovplyvňuje bezpečnosť celej počítačovej siete. O operačných systémoch s jadrom Linux kolujú mýty, že sú bezpečnejšie než operačné systémy od firmy Microsoft. Osobne považujem nesprávne nainštalovaný systém GNU/Linux za omnoho nebezpečnejší než systém MS Windows. Ak osoba, ktorá inštaláciu vykonáva, nepozná význam jednotlivých komponentov systému, je výsledkom inštalácie systém, na ktorom beží množstvo nepotrebných a nesprávne nakonfigurovaných služieb. Vďaka skutočnosti, že distribúcie systému GNU/Linux obsahujú aj množstvo sieťových démonov, ktoré by sme v systémoch rodiny Windows hľadali márne, hrozí riziko, že ich používateľ v svojej vlastnej nevedomosti nainštaluje a spustí pod dojmom, že sú nevyhnutné pre beh systému. Typickým príkladom takéhoto démona je napríklad démon inetd, ktorý je skvelým nástrojom napríklad pre tvorbu adaptívnych firewallov, no vo svojej defaultnej konfigurácii spúšťa zbytočné sieťové služby. Neznalého používateľa samozrejme zmätie popis balíku s týmto démonom, ktorý v množstve distribúcií obsahuje slovné spojenia ako napríklad „internet superserver deamon“.

Bezpečnosť operačného systému GNU/Linux je veľmi komplexná problematika. Vždy keď v mojom podvedomí začne prevládať pocit, že som v potrebnej miere oboznámený so všetkými hrozbami, ktoré na mnou spravovaný linuxový systém číhajú, objaví sa nejaká nová hrozba, s ktorou som v bezpečnostnej politike systému vôbec nerátal. Myslím si, že dobrý správca systému nesmie postrádať potrebnú dávku paranoje a vďaka nej každým dňom posilňovať bezpečnostné bariéry, ktoré systém chránia.

Bezpečnosť systémov Linux/UNIX je vo všeobecnosti možné rozdeliť na:

- fyzickú;
- systémovú;
- sieťovú.

## 2.1 Fyzická bezpečnosť

Na fyzickú bezpečnosť počítačových systémov má priamy vplyv bezpečnostná

politika organizácie. V dnešnom svete sú takmer všetky budovy, v ktorých je vykonávaná podnikateľská činnosť, chránené rôznymi bezpečnostnými systémami. S ohľadom na bezpečnosť systémov Linux/UNIX je nutné poznamenať, že ak útočník získa fyzický prístup k počítaču, na ktorom takýto systém beží, môže za ideálnych podmienok jednoduchým zásahom behom dvoch až troch minút získať konto správcu systému – roota. Preto je viac než potrebné zamedziť prístup neautorizovaných osôb do miestností, kde sa sa nachádzajú kriticky dôležité systémy.

### 2.1.1 Modelový útok

Aby sme mohli zabrániť fyzickému útoku na linuxové systémy, je nutné vedieť, ako takýto útok prebieha. V prvom rade sa útočník musí dostať k počítaču, v ktorom chce získať používateľský účet správcu. V ďalšom kroku sa pravdepodobne pokúsi z nejakého pamäťového média zaviesť na počítači iný systém, v ktorom na tzv. „root partícii“ v súbore */etc/passwd* zmaže záznam o existencii hesla pre účet používateľa root. To však nezmaže heslo samotné, ktoré je väčšinou uložené v súbore */etc/shadow*, a tak bude schopný pred skončením infiltrácie vrátiť systém do pôvodného stavu s aktívnym pôvodným heslom správcu. Po následnom reštarte sú všetky funkcie pôvodného systému obnovené, útočník sa môže prihlásiť ako používateľ root bez hesla a v systéme vykonať zmeny resp. do systému zaviesť trójskeho koňa alebo iný zlomyseľný kód. Nakoniec obnoví platnosť hesla pre používateľa root a odstráni záznamy o svojich aktivitách z log súborov systému. Inteligentný útočník tiež upraví uptime systému, aby sťažil jednoduchú možnosť detekcie reštartu systému.

### 2.1.2 Ochranné opatrenia

Základom ochrany proti fyzickému útoku je prísne kontrolovaný prístup do miestností, v ktorých sa nachádzajú kritické systémy. Bežne sa stretávame s tým, že pri vstupe do firemných budov musí každá osoba na vrátnici preukázať svoju totožnosť a uviesť dôvod návštevy. Menej bežné však je vidieť vo vnútorných priestoroch organizácie dôkladne uzamknuté a monitorované serverové miestnosti. Prístup do týchto priestorov by mal byť obmedzený len na dôveryhodné osoby, ktoré sú si vedomé bezpečnostných rizík spojených s prístupom ku kritickým systémom. V ideálnom prípade by dvere mali byť zabezpečené osobitným uzamykacím systémom a pod

neustálym dohľadom priemyselných kamier. Nemali by tu mať prístup dokonca ani pracovníci úseku údržby a pri prípadnom vykonávaní nutných činností týchto osôb by mal byť prítomný systémový administrátor. Tieto kroky vedú k aktívnemu zvýšeniu bezpečnosti systémov, no na rozdiel od ďalších možností, ktoré uvediem, nie sú realizovateľné bez podpory vedenia organizácie.

V súlade s teóriou bezpečnostných bariér musíme počítať s alternatívou, že sa útočníkovi podarí získať fyzický prístup k systému. Preto je nutné podniknúť kroky, ktoré mu znemožnia zaviesť cudzí operačný systém, z ktorého by sa pokúsil pripájať diskové oddiely, kde je nainštalovaný produkčný systém. Správca systému by mal teda z počítača odstrániť všetky mechaniky podporujúce externé pamäťové médiá, čím zabráni zavedeniu systému z optických diskov CD *(Compact Disc)* alebo klasických diskiet. Špeciálnu pozornosť treba venovať aj USB *(Universal Serial Bus)* portom, ku ktorým je možné pripojiť rôzne veľkokapacitné pamäťové médiá. Všetky porty, ktoré nie sú pre činnosť produkčného systému potrebné by mali byť vypnuté na úrovni BIOSu *(Basic Input/Output System)*. Taktiež je treba dať pozor na nastavenie tzv. boot sekvencie, aby sa počítač pri štarte nepokúšal zavádzať systém z iných zariadení než z pevného disku. Prístup k utilite SETUP, ktorou sa konfigurujú parametre BIOSu treba samozrejme zabezpečiť silným heslom. Toto heslo však neznamená veľkú prekážku ani pre priemerného používateľa osobných počítačov. Jeho prínos je najmä v tom, že útočníka zdrží a zvyšuje možnosť, že bude prichytený pri čine. Pre zrušenie tohto hesla je nutné vymazať pamäť CMOS *(Complementary Metal-Oxide Semiconductor)*, čo v praxi znamená rozobratie počítačovej skrine a odpojenie záložnej batérie na matičnej doske. Ak by chcel útočník vykonať túto operáciu na počítači s uzamknutou skriňou alebo na počítači umiestnenom v racku, pravdepodobne by sa vzdal už pri pomyslení na obtiažnosť a časovú náročnosť tejto práce.

Vhodným posilnením bezpečnosti je aj zavedenie monitorovacích prvkov, ktoré oznamujú nedostupnosť jednotlivých systémov. Takýmto prvkom môže byť napríklad zaslanie SMS *(Short Message Service)* správy systémovému administrátorovi pri vypínaní a reštartovaní serveru. Pridať túto akciu do skriptov spúšťaných pri vypínaní a reštarte systému je aj pre menej skúseného správcu otázkou pár minút. Nepredpokladám však, že by sa útočníkovi podarilo vypnúť systém bezpečným spôsobom, a preto je vhodné zaviesť aj nejaký druh sieťového monitorovania dostupnosti hostu. Takáto aplikácia môže bežať napríklad na pracovnej stanici správcu a v prípade, že server

neodpovie na ICMP *(Internet Control Message Protocol)* požiadavku, bude spustený zvukový signál.

S nutnou dávkou paranoje sa môžeme zamyslieť aj nad prípadom, že by sa útočníkovi podarilo zaviesť na našom počítači cudzí operačný systém. V takom prípade mu môže spôsobiť problémy už len šifrovaná root partícia. Ak však z nejakých príčin šifrovanie dát na root partícii nepoužívame, je vhodné vykonávať pravidelnú kontrolu integrity súborového systému. O tejto technike sa zmienim pri opise problematiky systémovej bezpečnosti.

Posledný krok útoku, keď útočník vymaže záznamy o svojej činnosti z log súborov je možné čiastočne obmedziť používaním špecializovaného sieťového logovacieho servera. Útočník síce zmaže záznamy na lokálnom disku, no ak by chcel odstrániť aj záznamy z logovacieho servera, znamenalo by to pre neho napadnutie ďalšieho systému a s tým spojené plytvanie drahocenným časom.

## 2.2 Systémová bezpečnosť

Do kategórie systémovej bezpečnosti patria najmä techniky útoku a obrany súvisiace so zneužitím v systéme existujúceho lokálneho účtu. Zahrnúť tu možno aj časti bezpečnostnej politiky pojednávajúce o správnom používaní a inštalovaní softvéru a o bezpečnosti jadra systému.

### 2.2.1 Postupy zvyšujúce systémovú bezpečnosť

Základným stavebným kameňom systémovej bezpečnosti sú silné heslá pre jednotlivé používateľské účty. Správne heslo by malo byť dlhé minimálne 8 znakov, malo by obsahovať malé aj veľké písmena abecedy, čísla a nejaké špeciálne ASCII *(American Standard Code for Information Interchange)* znaky ako napríklad pomlčku alebo bodku. Heslo zostavené podľa týchto pravidiel je schopné dlhú dobu odolávať útoku hrubou silou, pri ktorom útočník automaticky postupne skúša všetky možné kombinácie znakov. Keď je heslo dlhšie ako 8 znakov, môže trvať útok z jedného zdroja niekoľko dní ba aj týždňov a za túto dobu proti nemu systémový administrátor určite podnikne potrebné opatrenia.

Inštalácia a aktualizácia jednotlivých programových komponentov systému je tiež

záležitosťou, ktorú musí usmerňovať bezpečnostná politika organizácie. Na kritických systémoch nesmú byť inštalované nástroje umožňujúce vývoj alebo zostavenie aplikácií, aby používatelia alebo prípadní útočníci nemohli zostaviť nijaký vlastný program, ktorý by mohli použiť na kompromitovanie systému. Do produkčného systému taktiež nesmie byť inštalovaný softvér z neoficiálnych zdrojov. Do neoficiálneho zdroja môže pridávať balíky ktokoľvek a nie všetci ľudia majú dobré úmysly. Keďže inštaláciu nových balíkov vykonáva používateľ root, môžu byť následky podvrhnutého resp. mierne upraveného programu katastrofické. Videl som webové servery, na ktorých vykonal správca aktualizáciu systému z neoficiálnych mirrorov a zaviedol tak do systému zlomyseľný kód, ktorý prepísal všetky HTML *(HyperText Markup Language)* dokumenty na pripojených diskových oddieloch. Na kritických produkčných systémoch, ktoré vyžadujú vysokú bezpečnosť, by sa nemali inštalovať dokonca ani aktualizácie pochádzajúce z originálnych distribučných zdrojov bez toho, aby bola ich funkčnosť najskôr overená. Tiež je veľmi dôležité nikdy nezabúdať overovať signatúry inštalovaných balíkov resp. sťahovaných archívov so zdrojovými kódmi. Terčom útoku sa totiž môžu stať aj oficiálne mirrory a útočník môže získať prístup k mnohým systémom tým, že na nich sprístupní podvrhnuté balíky. Takýto podvod je možné odhaliť len pozorným kontrolovaním signatúr, ktoré sa väčšinou vytvárajú programom PGP *(Pretty Good Privacy)* alebo jeho open source alternatívou GnuPG *(Gnu Privacy Guard)*.

Jeden z hlavných princípov unixových systémov hovorí o tom, že všetko je súbor. Tento princíp je samozrejme zachovaný aj v systémoch s jadrom Linux, a preto sa v adresári */dev* nachádzajú špeciálne súbory reprezentujúce hardvér počítača. V praxi to znamená výhodu najmä v tom, že prístupové práva k hardvérovým prostriedkom môžeme nastavovať rovnakým spôsobom ako prístupové práva k bežným súborom. Je teda možné vytvoriť vysoko automatizované skripty, ktoré kontrolujú nastavenie prístupových práv k spomínaným zariadeniam a oznamujú prípadné zmeny správcovi. Taktiež je pre zvýšenie systémovej bezpečnosti nutné správne nastaviť prístupové práva, pre všetky súbory na pripojených diskových oddieloch. Zvýšenú pozornosť treba venovať najmä adresárom */etc*, */dev*, */lib* a */home*. Na systémoch, kde musia pracovať s príkazovým interpretom aj nedôveryhodní používatelia je dobré pre prípojné body */home* a */tmp* vytvoriť samostatné diskové oddiely a pripojiť ich s odopretým právom spúšťať akékoľvek súbory, ktoré sa na nich nachádzajú.

Existuje viacero projektov, ktoré sa venujú zvyšovaniu bezpečnosti linuxového kernelu. Medzi najznámejšie patrí SELinux, GRsecurity a Openwall. Popri aplikovaní týchto patchov je však vhodné pre serverové systémy vytvoriť aj kernel, ktorý nepodporuje dynamické pridávanie modulov. V prípade, že jadro podporuje pridávanie modulov, existuje teoretická možnosť, že útočník podvrhne falošný modul, ktorého zavedenie pri nasledujúcom reštarte mu zabezpečí získanie vyšších oprávnení. Nevýhodou takéhoto jadra je, že ak pridáme do počítača nejakú časť hardvéru, pre ktorú nemáme v jadre zakompilovanú podporu, budeme musieť znovu celé jadro rekompilovať.

So systémovou bezpečnosťou sú úzko späté aj techniky kontroly integrity súborového systému a analýzy log súborov. Túto problematiku podrobnejšie rozoberiem v ďalších kapitolách pri opise monitorovacích nástrojov a IDS *(Intrusion Detection System)*.

Častým prehreškom voči systémovej bezpečnosti sú služby spustené s právami používateľa root aj keď jeho privilégia pre ich činnosť vôbec nie sú potrebné. Najhoršie, čo sa môže v takomto prípade stať je, že sa lokálnemu alebo vzdialenému útočníkovi podarí odhaliť v tejto službe slabé miesto a prinúti ju k spúšťaniu ľubovoľného kódu. Kód je samozrejme spúšťaný s právami používateľa, pod ktorého účtom bola služba spustená. V prípade, že to bezpečnostný model aplikácie umožňuje, je dobré sa po vykonaní všetkých privilegovaných akcií zbaviť identity roota a zmeniť vlastníka procesu na neprivilegovaného používateľa. Takto pracujú napríklad mnohé sieťové služby, ktoré potrebujú pri svojom štarte otvoriť privilegovaný TCP port. Pri konfigurácii takýchto služieb je však potrebné dbať na to, aby každá z nich využívala iné používateľské konto. Ak sú totiž v systéme dva démony spustené pod používateľom nobody, znamená narušenie jedného z nich automaticky aj narušenie druhého.

## 2.3 Sieťová bezpečnosť

Najdôležitejším krokom v procese zvyšovania sieťovej bezpečnosti akéhokoľvek systému je vypnutie všetkých nepotrebných sieťových služieb. Snažiť sa odolať útokom zo siete na systéme, ktorý očakáva požiadavky od klientov na veľkom množstve portov je podobné ako snažiť sa obrániť pred nepriateľskou armádou tábor na otvorenej pláni. Ak však okolo tábora postavíme múr a ponecháme v ňom iba dve brány, bude stačiť ak

sústredíme sily práve k nim. Situácia so sieťovými službami a portami na ktorých počúvajú je obdobná. Ak ponecháme spustené iba tie sieťové služby, ktoré sú nevyhnutné na to, aby systém mohol plniť svoju funkciu, môžeme sústrediť prevažnú časť úsilia na zlepšovanie ich konfigurácie. Prínosom takéhoto postupu je úspora času a samozrejme zvýšenie miery zabezpečenia systému.

Existujú však aj správcovia, ktorí tvrdia, že na systéme môžu mať spustenú akúkoľvek nepotrebnú službu, ak prístup k nej ošetria pravidlami firewallu. Rozhodne však takýto postup nemôžem doporučiť. Snáď žiadny triezvo uvažujúci útočník z lokálnej siete sa neuspokojí s jednoduchým oskenovaním otvorených portov. Ja osobne by som sa napríklad v čase reštartu pracovnej stanice správcu serveru pokúsil prebrať jeho identitu a zistiť, aké služby sú prístupné pre jeho IP adresu. Je to triviálna úloha, ktorá sa dá plne automatizovať, no znemožniť by ju mohli napríklad starostlivo nakonfigurované switche. Z uvedeného teda jasne vyplýva, že sieťová bezpečnosť unixových systémov je do značnej miery závislá na bezpečnosti lokálnej siete a aktívnych prvkov v nej použitých. Je dobré zvyknúť si pre každý unixový systém vypracovať zoznam potrebných sieťových služieb aj s uvedením dôvodov, prečo sú jednotlivé služby pre systém potrebné a pri bezpečnostných auditoch overovať, či sú v systéme spustené naozaj iba tieto služby. Aj napriek tomu, že konfigurácii týchto vybratých služieb sa venuje maximálna pozornosť, je dobré použiť pravidlá firewallu alebo TCP wrappers na obmedzenie prístupu k nim čo najužšej skupine používateľov.

V prvej kapitole som medzi hrozbami v lokálnych sieťach popisoval techniku ARP poisoningu. Ako jeden z čiastočných spôsobov obrany som tiež uviedol nutnosť existencie statického ARP záznamu pre bránu siete v každom systéme, ktorý je súčasťou LAN. Pre zvýšenie sieťovej bezpečnosti je ale potrebné udržiavať aktuálne statické ARP záznamy okrem brány aj pre všetky systémy, ktoré sú pre činnosť serveru kritické. Nie je to síce neprekonateľná bezpečnostná bariéra, napokon ako každá, ale za istých podmienok môže zabrániť napríklad odpočúvaniu prenosov alebo aspoň útočníkovi sťažiť prácu.

### 2.3.1 Vytvorenie „fiktívnej reality“

Úlohou všetkých doteraz opísaných techník zvyšovania bezpečnosti bolo len odoprieť prístup útočníkom a osobám, ktoré naň nemajú nárok. V kapitolách o fyzickej

bezpečnosti je hlavný cieľ zabrániť prístupu do serverovej miestnosti, pri snahe o zvýšenie systémovej bezpečnosti sa snažíme napríklad nastaviť prístupové práva k hardvérovým prostriedkom počítača tak, aby k nim útočník nemal prístup, pri sieťovej bezpečnosti zas vytvárame pravidlá firewallu, ktoré zabránia pokusom o neautorizované využívanie sieťových služieb. Existujú však aj odlišné prístupy k zvyšovaniu bezpečnosti, či už jednotlivých systémov, alebo celých počítačových sietí.

Pokusy o útok vždy boli a vždy budú. To je skutočnosť, ktorá je vďaka ľudskej podstate nemenná. Mnohí experti zastávajú názor, že ak profesionál zameria svoju pozornosť na hľadanie chyby, tak aj v dôkladne nakonfigurovanej službe ju raz nájde a využije ju na prienik do systému. Určite aj vďaka tomuto názoru vznikli techniky, pri ktorých sa snažíme vytvoriť niečo ako „fiktívnu realitu“ a tú predostrieť útočníkovi. Cieľom je odlákať jeho pozornosť od reálne zneužiteľných systémov alebo ich častí a podsunúť mu tzv. medový hrniec (angl. honeypot), ktorý ho zabaví kým jeho prítomnosť nebude odhalená. Tieto systémy tiež často pracujú ako adaptívne firewally. Znamená to, že v momente keď sa útočník pokúsi napríklad pripojiť na falošný port, bude jeho IP adrese automaticky odopretý prístup k celej sieti.

Keď teda skenujeme sieťové porty nejakého systému a zistíme, že ich má otvorených päť, nemusí to byť pravda. Z toho teda názov „fiktívna realita“. V skutočnosti môže spomínaný systém prijímať požiadavky od klientov iba na jednom z týchto portov a akýkoľvek pokus o pripojenie na zvyšné štyri môže vyústiť k úplnému odopretiu prístupu. Alebo tiež môžu byť všetky porty falošné. Alebo môže na všetkých naozaj počúvať nejaká služba. Alebo.. Alebo.. Isté však je, že jediný chybný krok môže viesť k úplnému odopretiu prístupu k celej sieti, na ktorej sa systém nachádza. Nezvyčajné ale nie sú ani prípady, keď systémoví administrátori vytvoria systém, ktorý je jednoduchým sústom pre priemerne zručného útočníka. Ten tak pokračuje v útoku v utkvelej predstave, že získal vytúžený prístup ku kritickému systému a v skutočnosti svojim konaním zhromažďuje proti sebe dôkazy na úplne nepodstatnom systéme – honey pote.

Pri opise pokročilých postupov ochrany systémov však musím spomenúť aj techniku tzv. port knockingu. Je to metóda, ktorá umožňuje z externého hostu iniciovať otvorenie portu. Server monitoruje pokusy o pripojenie na množine preddefinovaných uzavretých portov. Keď je z externého hostu vykonaná správna sekvencia pokusov o

pripojenie na tieto porty, uvedú sa v platnosť nejaké rozšírené pravidlá firewallu. Napríklad môže byť externému hostu povolené pripojenie na špecifické porty. Implementácií tejto techniky existuje mnoho. Najjednoduchšia s akou som sa stretol, je vytvorenie špeciálnych pravidiel firewallu pre linuxový NF s využitím modulu recent. Klopanie sa v tomto prípade vykonáva pomocou notoricky známeho programu telnet. Medzi náročnejšie implementácie patria najmä tie na systémoch BSD, ktoré potrebujú spustený samostatný monitorovací démon a tiež na klopanie je potrebná špeciálna utilita.

Predstavme si, že v lokálnej sieti máme webový server, ktorého správu vykonávame prostredníctvom SSH. Ak aj obmedzíme prístup k portu 22/TCP len na IP adresu pracovnej stanice správcu, môže útočník za špecifických podmienok získať jeho identitu a dostane tak šancu pokúsiť sa o útok na SSH démona. Ak však použijeme na kontrolu prístupu k tomuto portu techniku port knockingu, musel by útočník poznať aj správnu kombináciu na klopanie. To by sa ale najskôr musel dovtípiť, že na spomínanom serveri bezpečný shell SSH vôbec beží.

Kombináciou techník port knockingu a falošných portov s adaptívnym firewallom môžeme dosiahnuť veľmi vysokú úroveň sieťovej bezpečnosti unixového systému. Navyše na platformách s jadrom Linux je implementácia tohto riešenia možná s využitím štandardných komponentov systému ako inetd, iptables a TCP wrappers.

# 3 MONITOROVACIE NÁSTROJE PRE LINUX/UNIX

Pre neustále zlepšovanie a optimalizovanie zabezpečenia systému je nutné vykonávať komplexné monitorovanie aktivít a javov, či už sieťových, systémových alebo hardvérových. Dodržiavanie základných princípov teórie bezpečnostných bariér síce pomáha pri zvyšovaní miery zabezpečenia systému, no intenzívny pokus o útok môžeme včas zastaviť jedine v prípade, že o ňom vieme. Záber pojmu monitorovanie systému je však veľmi široký. Sledovať možno akúkoľvek časť systému a v každom systéme sú požiadavky individuálne. V rámci tejto práce som sa pokúsil obsah pojmu monitorovanie počítačového systému rozdeliť a bližšie špecifikovať jeho jednotlivé časti.

**Monitorovanie počítačového systému:**

- **monitorovanie hardvérových súčastí systému**
  - monitorovanie vyťaženia CPU
  - monitorovanie záťaže sieťových rozhraní
  - monitorovanie využitia pamäte RAM
  - monitorovanie využitia kapacity pevných diskov
  - monitorovanie teplôt jednotlivých hardvérových súčastí
- **monitorovanie softvérových súčastí systému**
  - monitorovanie sieťových aktivít systému
  - monitorovanie integrity súborového systému
  - monitorovanie spustených procesov
  - monitorovanie systémových log súborov
  - monitorovanie útokov a pokusov o prienik

Toto rozdelenie však určite nie je úplné a hranice medzi jeho jednotlivými časťami nie sú vždy jasné. Už v úvode som spomínal, že každý systém je individuálny, a tak k nemu treba pristupovať aj v oblasti monitorovania. Na niektorých počítačoch je smerodajné vyťaženie CPU, na iných zas využitie kapacít sieťových rozhraní a dokonca sa nájdu aj systémy, na ktorých je nutné monitorovať „exotickejšie“ prvky, ako napríklad počet prístupov používateľov za určité časové obdobie. V unixových

systémoch je implementácia aj takýchto nevšedných požiadaviek možná a ľahko realizovateľná, pretože sú pre ne voľne dostupné kvalitné a vysoko konfigurovateľné nástroje.

## 3.1 Monitorovanie hardvérových súčastí systému

Pri monitorovaní hardvérových súčastí systému sa často využíva aplikačný protokol SNMP *(Simple Network Management Protocol)*, ktorý je primárne určený na monitorovanie stavu zariadení pripojených k počítačovej sieti. Podstata komunikácie prostredníctvom tohto protokolu je veľmi jednoduchá. Klient (subagent) si vyžiada od servera (master agent) informácie o nejakom konkrétnom objekte, ktorý server pozná a vie o ňom poskytnúť informácie (napríklad vyťaženie CPU). Server zašle klientovi v odpovedi aktuálnu hodnotu prislúchajúcu danému objektu. Už pri návrhu tohto protokolu sa počítalo s nutnosťou jeho budúceho rozširovania o podporu nových objektov, a preto je v ňom implementovaný špeciálny typ databázy zvanej MIB *(Management Information Base)*. Ak chce výrobca hardvéru pre svoj produkt zaviesť podporu do rôznych agentov alebo subagentov, stačí ak vydá rozšírenie MIB. Napriek tomu, že protokol SNMP už nie je práve najmladší, používa sa vďaka kvalitnému návrhu dodnes. Výraznými zmenami prešiel akurát spôsob autentifikácie používateľov, následkom čoho sú dnes známe tri verzie tohto protokolu SNMPv1, SNMPv2c a SNMPv3. Protokol SNMP sa dá využiť nielen na získavanie informácií o jednotlivých definovaných objektoch, ale aj na nastavovanie konkrétnych hodnôt pre tieto objekty. Tak môže administrátor napríklad vzdialene spravovať smerovaciu tabuľku alebo ARP záznamy na aktívnych sieťových prvkoch. Výhodou oproti webovým a telnetovým správcovským rozhraniam je možnosť správu protokolom SNMP skriptovať a teda kompletne automatizovať. [4]

K protokolu SNMP existujú aj alternatívne technológie, ktoré sa snažia vytvoriť podobné rozhranie pre získavanie informácií o systéme. Sú to však väčšinou len pokusy postrádajúce sieťový model, a preto dokážu pracovať iba na konkrétnom hoste. To znemožňuje ich nasadenie vo väčších sieťach, kde sa na monitorovanie systémov vyhradzuje osobitný počítačový systém získavajúci údaje prostredníctvom siete.

Pri popise možností monitorovania hardvérových súčastí systému však nesmie byť vynechaný projekt lm_sensors, ktorý zavádza do linuxového jadra podporu pre

zariadenia umiestnené na zberniciach I2C *(Inter-Integrated Circuit)* a SMBus *(System Management Bus)* slúžiacich v osobných počítačoch najmä na sprostredkovanie informácií z diagnostických senzorov ako je napríklad snímač teploty CPU. Pre serverové systémy je prítomnosť takéhoto softvéru nevyhnutnosťou, pretože znalosť aktuálneho „zdravotného stavu“ počítača pomáha ľahšie identifikovať aktuálne resp. predpovedať budúce problémy.

## 3.2 Monitorovanie sieťových aktivít systému

Samotný proces monitorovania sieťových aktivít systému by sa dal rozdeliť na monitorovanie využitia sieťových kapacít a na monitorovanie aktuálnych spojení a otvorených sieťových portov.

Na monitorovanie využitia sieťových kapacít alebo inak povedané vyťaženia dostupnej linky sa používajú dva druhy nástrojov, ktoré majú mierne odlišný princíp činnosti. Nástroje z prvej skupiny využívajú na získanie aktuálnych hodnôt protokol SNMP. Zozbierané dáta následne do grafickej podoby spracováva nejaký vizualizačný softvér. Nástroje z druhej skupiny sú väčšinou založené na knižnici libpcap, ktorá je hojne využívaná napríklad analyzátormi sieťových protokolov. Tieto produkty prepnú sieťové rozhranie do promiskuitného režimu a analyzujú všetky prichádzajúce pakety. Ich výhodou oproti riešeniam založeným na protokole SNMP je, že ponúkajú množstvo nastavení. Dajú sa s nimi monitorovať napríklad iba určité rozsahy adries alebo vybrané protokoly a väčšinou obsahujú aj modul pre vizualizáciu dát.

Monitorovanie aktívnych spojení a otvorených sieťových portov plní v procese zabezpečenia a monitorovania operačného systému veľmi dôležitú úlohu. Ak by bol systém kompromitovaný a bol by na ňom spustený nejaký backdoor, alebo ak by sa útočník pripájal pomocou tunelovaných spojení, tak monitorovanie aktívnych spojení a otvorených sieťových portov je prakticky jediná možnosť ako ho odhaliť. Na kritických systémoch je teda okrem reštriktívnych nastavení firewallu nutné aj vykonať niekoľko krát za deň monitorovanie spomínaných záležitostí. Bez problémov sa na to dajú použiť štandardné nástroje ako netstat alebo nmap, ktoré sú dostupné vo väčšine linuxových distribúcií.

## 3.3 Monitorovanie integrity súborového systému

Neodmysliteľnou súčasťou detekcie prienikov v operačných systémov je sledovanie zmien dôležitých systémových súborov. Nie je nič nezvyčajné, keď v snahe vyťažiť maximum z úspešného prieniku, zamení útočník niektoré systémové programy za svoje upravené verzie. Ak zamení napríklad SSH klienta, môže získať prístupové údaje k iným systémom, na ktoré sa používatelia napadnutého systému vzdialene prihlasujú. Preto sa používajú techniky kontrolujúce integritu jednotlivých súborov pomocou hašovacích algoritmov ako napríklad MD5 *(Message-Digest Algorithm 5)* alebo SHA-1 *(Secure Hash Algorithm)*. Pri prvom spustení nástroja pre kontrolu integrity súborového systému sa vytvorí databáza kontrolných súčtov definovaných súborov a je potrebné ju uložiť na médium chránené proti zápisu, aby ju útočník nemohol pozmeniť. Tato databáza je používaná ako etalón a všetky neskôr vypočítané kontrolné súčty sú proti nej porovnávané. Samozrejme v prípade aktualizácie dôležitých komponent systému je treba referenčnú databázu aktualizovať.

## 3.4 Monitorovanie spustených procesov

Je dobré, ak bezpečnostná politika definuje pre každý systém zoznam procesov, ktoré na ňom môžu bežať. Procesy bežiace na systéme by sa mali nepretržite a v nepravidelných intervaloch porovnávať s týmto zoznamom a o každom výskyte neznámeho procesu by mal byť informovaný správca systému.

Neustále monitorovanie bežiacich procesov však nemá charakter iba bezpečnostný, ale zabezpečuje aj opätovné spustenie dôležitých procesov v prípade ich pádu. Softvér vykonávajúci túto úlohu musí byť schopný o všetkých vykonaných akciách informovať správcu systému poprípade ich zaznamenávať do log súborov operačného systému.

## 3.5 Monitorovanie systémových log súborov

V svete unixových systémov takmer všetky procesy zapisujú informácie o svojej činnosti do log súborov. V skutočnosti ich tam nezapisujú priamo, ale prostredníctvom špecializovaného démona zvaného syslog. Tieto súbory sa zvyčajne nachádzajú v

adresári */var/log*, ktorý by mal byť čitateľný iba pre správcu systému, pretože môže obsahovať citlivé informácie o systéme alebo jeho používateľoch. Ak počas činnosti niektorého z procesov nastane chyba, nezobrazí sa priamo na monitore, ale odovzdá sa jej popis démonovi syslog, ktorý ho zapíše do log súborov. Správca systému z nich následne môže v prípade problémov zistiť čo bolo príčinou chýb. Tieto súbory je však potrebné kontrolovať pravidelne, pretože môžu obsahovať dôležité informácie o stave hardvéru, konfiguračných chybách, pokusoch o prienik a mnohých ďalších nemenej dôležitých udalostiach. Snáď každému bežnému človeku sa po pár pokusoch o manuálne kontrolovanie týchto súborov zrodí v hlave myšlienka, že táto činnosť by mala byť automatizovaná, pretože je časovo náročná a príliš monotónna.

Existuje viacero nástrojov, ktoré slúžia na automatizovanú kontrolu log súborov. Ich činnosť je v podstate rovnaká. Vyberú z log súborov iba riadky obsahujúce vybrané slová alebo riadky zodpovedajúce preddefinovaným regulárnym výrazom a odošlú ich správcovi systému.

V organizáciách, ktoré vlastnia viac ako dva unixové servery je vhodné zvážiť nasadenie dedikovaného logovacieho servera. Log súbory na ostatných serveroch sa tak môžu nechať rotovať nástrojom ako je napríklad logrotate a ich archiváciu stačí vykonávať iba na logovacom serveri. Ak by sa útočníkovi podarilo napadnúť niektorý zo systémov, môže síce zmazať záznamy na lokálnom disku, no nezmaže ich z logovacieho servera. Pri takomto riešení však treba vyriešiť problém synchronizácie času medzi jednotlivými systémami, pretože časový posun medzi nimi môže znemožniť spätnú analýzu prienikov alebo iných mimoriadnych situácií. Na synchronizáciu času jednotlivých počítačových systémov sa zvyčajne používa protokol NTP (Network Time Protocol), ktorý je implementovaný aj v mnohých hardvérových produktoch synchronizovaných pomocou GPS *(Global Positioning System)*.

### 3.6 Monitorovanie útokov a pokusov o prienik

Systémy na detekciu pokusov o prienik (ďalej len IDS) sú schopné zachytiť mnohé druhy zlomyseľných aktivít na sieti i v samotnom operačnom systéme, ktoré nemôžu byť zachytené bežnými firewallovými systémami. Okrem bežných sieťových útokov, ktoré môžu byť čiastočne blokované aj firewallom, dokáže IDS vďaka operovaní na aplikačnej vrstve zachytiť aj prechádzajúce vírusy, trójske kone či

internetové červy.

IDS môžu pracovať s bázou znalostí o prienikoch a porovnávať s ňou vyskytujúce sa udalosti, alebo môžu detekovať odchýlky od definovaného normálu. Ak sa ich činnosť opiera o bázu znalostí, dokážu rozoznávať len známe útoky, ktoré sú v databáze resp. podobné útoky, ktoré sú svojimi hlavnými prvkami podobné už známym útokom. Kvalita týchto IDS závisí od kvality bázy vzoriek. Naproti tomu, IDS založené na vyhľadávaní odchýlok od definovaného normálu sa dokážu učiť a tak zlepšovať svoj výkon.

Podľa poľa pôsobenia môžeme IDS rozdeliť na HIDS *(Host-based Intrusion Detection System)* teda hostové systémy na detekciu prieniku a NIDS *(Network Intrusion Detection System)* čiže sieťové systémy na detekciu prieniku. Úlohou HIDS je monitorovať systémové volania, log súbory, modifikácie súborového systému a iné aktivity v konkrétnom operačnom systéme. NIDS sú zase často inštalované na prístupové body lokálnej siete, kde analyzujú sieťovú prevádzku celej siete.

Úlohou pasívnych IDS je pokusy o prienik len vyhľadávať a informovať o nich správcu systému. Existujú však aj tzv. aktívne IDS, ktoré fungujú ako adaptívne firewally a dokážu podozrivým IP adresám zabrániť v prístupe ku konkrétnemu systému alebo dokonca k celej lokálnej sieti. [5]

# 4 ZOSTAVENIE FIREWALLOVÉHO SYSTÉMU

Pri tvorbe firewallového systému si treba najmä uvedomiť, že pravdepodobne bude oddeľovať viacero sietí a tak akýkoľvek jeho výpadok bude znamenať obmedzenie konektivity medzi nimi. Neuvážený výber hardvéru, operačného systému alebo ostatných softvérových komponentov sa môže neskôr ukázať pre organizáciu veľmi nákladný. Nikdy nie je na škodu veci ak firewallový systém tvoria dva redundantné počítačové systémy. Na prvý pohľad to síce môže vyzerať ako mrhanie finančnými prostriedkami, no určite každý, kto zažil výpadok prístupového bodu siete vie, že to prináša mnohé komplikácie, ktoré sa negatívne prejavia na pracovnom výkone celej firmy. Práve preto sa pri tvorbe firewallových systémov všeobecne preferujú iba osvedčené hardvérové i softvérové produkty. Ak zadávateľ vyslovene vyžaduje nasadenie produktu, s ktorým správca nesúhlasí, je nutné spísať o tom dohodu, v ktorej sa zadávateľ zaviaže, že preberá na seba zodpovednosť za možné následky vyplývajúce z tohto rozhodnutia.

## 4.1 Operačný systém – Slackware Linux

Pri výbere vhodného operačného systému pre túto prácu som dlhšiu dobu váhal, či mám použiť Slackware Linux alebo OpenBSD. Použitie OpenBSD by znamenalo značné uľahčenie mojej práce a úsporu času, pretože väčšina softvérových produktov, ktoré sa chystám na vytvorenie firewallového systému použiť, je dostupná priamo vo forme distribučných binárnych balíkov. To, čo nakoniec prekvapivo rozhodlo o nepoužití OpenBSD bola jeho najmocnejšia súčasť, a síce PacketFilter, ktorý nie je schopný pracovať na inej než sieťovej a transportnej vrstve protokolu TCP/IP. Pri vytváraní pravidiel firewallu je často nutné použiť aj MAC adresu, ktorá je identifikátorom na linkovej vrstve, no PF to nedovoľuje. Niekto môže namietať, že stačí vytvoriť pre jednotlivé IP adresy statické ARP záznamy, no v skutočnosti ich firewall zostavený ako ethernet bridge ignoruje. Na tento problém existuje riešenie a v jednej z predchádzajúcich kapitol som ho popisoval, no výrazne zvyšuje nároky na výkon počítačového systému.

Vybral som teda Slackware linux, distribúciu systému GNU/Linux s viac ako

desaťročnou tradíciou. Vytvoril a udržiava ju prevažne jediný človek – Patrick Volkerding. Slackware je známy najmä svojou stabilitou, výkonnosťou a podobou BSD systémom. Neobsahuje zbytočne veľa grafických konfiguračných nástrojov, a preto panuje názor, že nie je vhodný pre začínajúcich používateľov. Prepracovaná inštalácia, ktorá dáva používateľovi absolútnu kontrolu nad tým, čo bude výsledný systém obsahovať, ho priam predurčuje na to, aby bol nasadený v serverových systémoch. Nevýhodou oproti iným distribúciám systému GNU/Linux ako aj OpenBSD je snáď iba skutočnosť, že obsahuje relatívne málo binárnych distribučných balíkov. Slackware je distribúciou veľmi konzervatívnou a dodnes používa kernel rady 2.4. Dôvodom pre zmenu na radu 2.6 môže byť potreba podpory nových zariadení, alebo potreba niektorých nových modulov pre NetFilter, ako je napríklad modul physdev uľahčujúci filrovanie sieťovej prevádzky na firewallových systémoch pracujúcich ako ethernet bridge.

## 4.2 Vybrané softvérové produkty

V tejto kapitole opisujem softvérové produkty, s ktorými mám vlastné skúsenosti a považujem za vhodné nasadiť ich vo firewallovom systéme. Inštalačný program, ktorý popri tvorbe tejto práce vytvorím, bude schopný nainštalovať každý jeden z nich a pre niektoré dokáže vykonať resp. odporučiť i základnú konfiguráciu.

### 4.2.1 Dynamické prideľovanie IP adries – DHCPd

Aby mohol počítač pracovať v lokálnej sieti a mať prípadne prístup na internet, je nutné nakonfigurovať v jeho operačnom systéme niektoré parametre dôležité pre protokol TCP/IP. Zväčša sú to IP adresy jednotlivých sieťových rozhraní, masky siete, predvolená brána a IP adresa jedného alebo viacerých DNS *(Domain Name System)* serverov. Tieto parametre je možné zadať na každom systéme v sieti manuálne a v prípade reinštalácie operačného systému túto akciu opakovať, alebo je možné prideľovať ich dynamicky s využitím protokolu DHCP *(Dynamic Host Configuration Protocol)*. Osobne preferujem druhú možnosť, pretože výrazne uľahčuje prácu najmä v prípade zmeny rozsahu používaných IP adries alebo ktoréhokoľvek iného parametra. V Slackware linuxe sa priamo v distribučných balíkoch nachádza DHCP démon od ISC *(Internet Systems Consortium)*. Tento démon je možné nastaviť tak, aby konkrétnej

MAC adrese vždy pridelil rovnakú IP adresu, čo zaručí, že IP adresy jednotlivých systémov sa nebudú meniť, ale budú prideľované dynamicky pri štarte operačného systému. Takéto riešenie dáva správcovi siete jednoduchú možnosť ovplyvniť nastavenie protokolu TCP/IP na jednotlivých systémoch v LAN bez nutnosti fyzického zásahu. Ďalšou výhodou, ktorú použitie tohto démona prináša je možnosť dynamického pridelenia IP adresy z predom definovaného rozsahu pre neznáme systémy pripojené k lokálnej sieti. Prejaví sa to ako výhodné najmä v prípadoch, keď napríklad obchodný partner s notebookom potrebuje prístup k internetu. Stačí, ak sa fyzicky pripojí k lokálnej sieti a nastaví vo svojom operačnom systéme automatickú konfiguráciu parametrov protokolu TCP/IP. Takéto nastavenie DHCP démona je však treba veľmi dobre zvážiť, pretože ak sa podarí útočníkovi dostať do priestorov budovy a pripojiť k lokálnej sieti, tak nemusí strácať čas zisťovaním používaného rozsahu IP adries. Tu si opäť dovolím zdôrazniť, že všetky sieťové káble v interiéri je treba viesť v uzatvorených lištách a zásuvky vyvádzať len do uzamykateľných priestorov.

### 4.2.2 Vzdialená správa serveru – OpenSSH a port knocking

Požiadavky na vzdialený prístup k počítačovým systémom sú vo všeobecnosti rôzne a menia sa prípad od prípadu. Ak je napríklad server umiestnený v racku a nie je k nemu permanentne pripojený monitor, dá sa vzdialená správa považovať za nutnosť. Ak je server umiestnený v dosahu správcu a nič mu nebráni vo fyzickom prístupe k nemu, je inštalácia vzdialenej správy zbytočná. Treba si uvedomiť, že prítomnosť akejkoľvek formy vzdialenej správy zvyšuje riziko, že systém bude terčom útokov, a preto je nutné tieto služby konfigurovať s najväčšou opatrnosťou, čo najviac k nim obmedziť prístup napríklad pravidlami firewallu a zo zásady používať iba služby podporujúce šifrované spojenie. Takmer ideálnym kandidátom na túto úlohu je démon OpenSSH, ktorý je vyvíjaný autorským tímom OpenBSD a jeho binárny balík je súčasťou distribúcie Slackware. Pre zvýšenie bezpečnosti vždy upravujem jeho konfiguráciu tak, aby využíval iba protokol SSH verzie 2 a aby neumožňoval priame prihlásenie používateľa root. Je dobré ak SSH démon akceptuje prihlásenie iba jediného neprivilegovaného používateľa, ktorý môže získať oprávnenia správcu pomocou príkazu su až po úspešnom prihlásení. Navyše, ak je to čo i len trochu možné, maskujem port využívaný bežiacim démonom OpenSSH technikou port knockingu, ktorú som popísal v kapitole o vytváraní fiktívnej reality. To zabezpečí, že táto služba ostane pre nepovolané osoby

skrytá.

### 4.2.3 Synchronizácia času – OpenNTPd

Je veľmi dôležité, aby serverové systémy pracovali s presným časom. Každý jeden záznam v log súboroch sa totiž začína údajom o čase, kedy bol vytvorený. Jedine v prípade, že sú tieto časové údaje správne, môžeme spätne zrekonštruovať nejakú situáciu, identifikovať systémový, či konfiguračný problém, alebo použiť log súbory pri prípadnom súdnom spore s prichyteným útočníkom.

Na synchronizáciu času som pre mnou navrhovaný systém zvolil produkt OpenNTPd, ktorý bol taktiež vyvinutý ako súčasť systému OpenBSD. Dokáže synchronizovať systémový čas s definovanými NTP servermi a zároveň môže pôsobiť sám ako NTP server a umožňovať synchronizáciu času iným systémom.

### 4.2.4 Implementácia VPN siete – OpenVPN

OpenVPN implementuje bezpečnostné rozšírenie sieťovej vrstvy modelu OSI. Používa na to štandardné protokoly SSL/TLS *(Secure Sockets Layer / Transport Layer Security)* a podporuje viaceré metódy autentifikácie klientov. Umožňuje tiež vytvárať osobitné pravidlá firewallu pre spojenia pochádzajúce z VPN vďaka tomu, že pracuje s virtuálnymi sieťovými rozhraniami TUN/TAP, ktoré softvérovo simulujú sieťové zariadenia a musia mať podporu v jadre operačného systému. Kým bežné sieťové rozhrania ako eth0 priamo zastupujú hardvér, čo môže byť napríklad sieťová karta v PCI slote, tak pakety prechádzajúce rozhraniami TUN/TAP sú preposielané používateľským programom, čo je v tomto prípade OpenVPN. Rozhranie TUN simuluje point-to-point zariadenie a TAP klasické zariadenie typu ethernet. OpenVPN dokáže tunelovať sieťovú komunikáciu cez jediný TCP alebo UDP port. Tunelovanie cez TCP port sa však prejavuje mierne zvýšenými nárokmi na réžiu, pretože protokol TCP overuje integritu preposielaných dát.

Mnou navrhnutý firewallový systém bude obsahovať konfiguráciu VPN siete len so statickými kľúčmi a s použitím prekladu zdrojových adries pre odchádzajúce spojenia, pretože komplexnejšiu konfiguráciu je aj tak vždy nutné prispôsobiť konkrétnym podmienkam. Vytvorený systém bude teda v predvolenej konfigurácii

použiteľný len na bezpečné prepojenie dvoch od seba vzdialených sietí alebo na umožnenie prístupu k lokálnej sieti jednému vzdialenému systému.

### 4.2.5 Sieťové sprístupnenie informácií o systéme – Net-SNMP

Operačný systém Slackware Linux je schopný komunikovať prostredníctvom protokolu SNMP len v prípade, že je v ňom spustený démon, ktorý dokáže na SNMP požiadavky odpovedať. Výber konkrétnej implementácie takéhoto démona bol dosť jednoznačnou záležitosťou, pretože v súčasnosti na tomto poli dominuje jediný projekt Net-SNMP. Je stále aktívne vyvíjaný a takmer po dvoch rokoch používania som v ňom neobjavil nič, čo by ma donútilo za neho hľadať nejakú náhradu.

### 4.2.6 Transparentný proxy server – SQUID

V sieťach, kde je nutné umožňovať používateľom prístup k prostriedkom internetu len po overení používateľského mena a hesla, alebo kde je nutné vykonávať filtráciu prenášaných dát v závislosti na obsahu, je vhodným riešením zavedenie proxy serveru. Za normálnych podmienok musí každá aplikácia, ktorá potrebuje pristupovať na internet podporovať nastavenie prístupu cez proxy. Všetky požiadavky zasiela tomuto serveru a ten sa potom ako klient pripája na externé systémy. Po získaní požadovaných dát ich poskytne klientovi v lokálnej sieti, pre ktorého sa javí ako server. Samozrejme medzi tým môže vykonať úpravy týchto dát alebo ich aj uložiť do vyrovnávacej pamäte (angl. cache) a z nej neskôr poskytnúť ďalším klientom, čím šetrí aj prenosové kapacity siete. Je dôležité poznamenať, že proxy server pracuje na aplikačnej vrstve, a preto musí každý aplikačný protokol, ktorý obsluhuje poznať. Ďalšou bežnou možnosťou nasadenia proxy serveru je tzv. transparentný proxy server. Ak je ako brána siete použitý tento typ serveru, nepotrebujú klientské stanice žiadnu špeciálnu konfiguráciu. Na port proxy serveru bežiaceho na bráne siete sú vďaka prekladu cieľových adries presmerované všetky spojenia smerujúce na port 80/TCP. Proxy server tieto požiadavky vybaví s externými systémami a napokon klientskej stanici z lokálnej siete zašle odpoveď. Klient tak ani nezistí, že jeho požiadavky mohli prejsť filtrovaním obsahu.

Na unixových systémoch sa veľkej popularite teší proxy server Squid, ktorý dokáže pracovať s protokolmi HTTP, FTP *(File Transfer Protocol)* ale aj HTTPS a

mnohými ďalšími. Mnou navrhnutý firewallový systém ho bude schopný používať v oboch opisovaných módoch.

### 4.2.7 Vytvorenie redundantného systému – UCARP

Protokol CARP je bezpečnou a voľne dostupnou alternatívou k protokolom VRRP *(Virtual Router Redundancy Protocol)* a HSRP *(Hot Standby Router Protocol)*. Dovoľuje viacerým systémom z tzv. „redundantnej skupiny“ zdielať v lokálnej sieti rovnakú IP adresu. Jeden systém z tejto skupiny je považovaný za „hlavný“ (angl. master) a ostatné za „zálohy“ (angl. backups). Za normálnych okolností používa zdieľanú IP adresu iba hlavný systém a vybavuje všetky požiadavky, ktoré sú na ňu smerované. Ak by však tento systém z ľubovoľných príčin prestal reagovať, začne zdieľanú IP adresu používať jeden zo záložných systémov v závislosti od jeho priority. [6]

Tento protokol je taktiež jednou z mnohých skvelých technológií, ktoré dala svetu komunita vývojárov zoskupená okolo systému OpenBSD. Jeho port pre ostatné systémy, medzi ktoré patrí samozrejme aj GNU/Linux sa volá UCARP a je k dispozícii na webovej stránke www.ucarp.org. Zahŕňam ho ako súčasť firewallového systému aj keď nepredpokladám jeho časté využitie. Myslím si však, že kvalitný firewallový systém musí poskytovať možnosť vytvorenia redundantného systému. Riešenie s protokolom CARP dáva možnosť vytvorenia redundantnej skupiny firewallov aj s odlišnými operačnými systémami. Situácia, keď sa útočníkovi podarí prelomiť bezpečnostné bariéry linuxového firewallového systému, tak môže byť veľmi jednoducho vyriešená jeho vypnutím, ak jeho funkcie okamžite na seba preberie firewall s operačným systémom OpenBSD.

### 4.2.8 Firewallový systém ako ethernet bridge – Bridge-utils

Súčasťou distribúcie Slackware nie sú nástroje umožňujúce zoskupovanie sieťových rozhraní tak, aby vytvorili ethernet bridge. Preto pridávam do firewallového systému balík nástrojov menom Bridge-utils, ktorý je šírený pod licenciou GNU/GPL a je stiahnuteľný zo stránky bridge.sourceforge.net.

## 4.3 Monitorovacie nástroje

Pre systém GNU/Linux je dostupných oveľa viac monitorovacích nástrojov než pre iné operačné systémy a sú schopné plniť aj tie najzložitejšie funkcie. Nenadarmo sa medzi „zasvätenými“ zvykne hovoriť *„If Linux doesn't have solution, you have the wrong problem.“*, čo vo voľnom preklade znamená, *„Ak Linux nemá riešenie, tak máte zlý problém.“*. Navyše mnohé z týchto nástrojov sú nezávislé na platforme a dokážu vzdialene monitorovať aj počítačové systémy s inými operačnými systémami, či už je to nejaká odroda BSD unixu, Solaris alebo Windows.

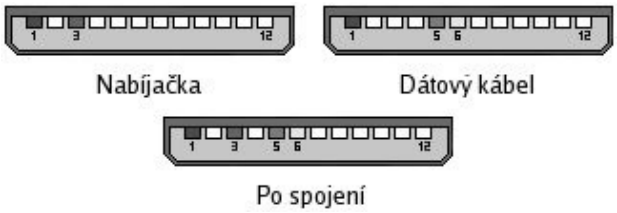
### 4.3.1 Monitoring teploty a napätí – lm_sensors

Medzi štandardné súčasti navrhovaného systému som zaradil už spomínaný program lm_sensors, ktorý umožňuje získavať informácie o jednotlivých napätiach, ktoré systému sprostredkúva napájací zdroj a o teplotách hardvérových prvkov počítača. Binárny balík, ktorý obsahuje okrem obslužných programov aj ovládače pre i2c subsystém a jednotlivé podporované senzory som kompiloval pre jadro 2.4.31, ktoré je základným jadrom systému Slackware Linux 10.2. Pripravil som však aj balík pre systémy s jadrom rady 2.6, ktorý obsahuje len obslužné programy, pretože ovládače pre jednotlivé senzory ako aj zbernice i2c a SMBus sú štandardnou súčasťou tejto verzie jadra.

### 4.3.2 Oznamovanie udalostí prostredníctvom siete GSM - SCMxx

Na svete nič nie je ideálne a výnimkou nie sú ani počítačové siete. Nie je neobvyklé, keď dôjde k prerušeniu spojenia lokálnej siete s okolitým svetom kvôli výpadku jedného z aktívnych sieťových prvkov. Aby nás firewallový systém o tomto stave dokázal informovať aj v prípadoch, keď nemá prístup k sieťovým prostriedkom, je nutné použiť iné technológie pre prenos informácií. Jednou z lacnejších a ľahko dostupných možností je pripojiť k počítaču mobilný telefón pre sieť GSM *(Global System for Mobile Communications)*. Z vlastnej skúsenosti odporúčam použitie mobilných telefónov značky Siemens, pretože je pre ne dostupné množstvo programov a existuje k nim podrobná dokumentácia. Problém, ktorý však treba pri neustálom pripojení telefónu značky Siemens k počítaču vyriešiť je jeho napájanie, pretože tieto

telefóny používajú jediný konektor pre napájanie aj pre dátové prenosy. Preto je nutné rozobrať koncovku na dátovom kábli i na nabíjačke a integrovať ich do jednej ako je to znázornené na obrázku č. 6.




*Obr.6: Náčrt konektorov pred úpravou a po nej*

Pre samotnú komunikáciu s telefónom a jeho ovládanie som do firewallového systému zaradil program SCMxx od nemeckého autora Hendrika Sattlera, ktorý podporuje väčšinu modelov mobilných telefónov od firmy Siemens a pracuje v textovom režime, vďaka čomu je ľahko použiteľný v skriptoch. Dá sa použiť napríklad na oznamovanie o výpadkoch linky alebo na pravidelné informovanie o vyťažení systému.

### 4.3.3 Monitoring sieťových rozhraní – MRTG, Bandwidthd a IFplugd

Ak je v systéme prítomný SNMP démon, tak je možné na monitorovanie využitia sieťových kapacít systému použiť pod licenciou GNU/GPL šírený produkt MRTG *(Multi Router Traffic Grapher)*. Ten je schopný vďaka spolupráci s grafickými knižnicami GD a libpng vytvárať zo zozbieraných dát prehľadné grafy. Nie je však obmedzený len na monitorovanie vyťaženia dostupnej linky, ale môže tvoriť grafy pre ľubovoľný objekt dostupný prostredníctvom protokolu SNMP. S obľubou ho preto používam aj na vytváranie grafov pre vyťaženie procesora, pamäte RAM, SWAP partície a teplôt jednotlivých hardvérových komponentov. MRTG okrem obrázkov grafov vytvára aj jednoduché HTML stránky s ich popisom. Tie sa dajú prezerať buď lokálne alebo je možné sprístupniť ich pomocou webového servera Apache, ktorý je súčasťou distribúcie Slackware.

Ak je nutné na firewallovom systéme monitorovať aj sieťovú aktivitu pre jednotlivé IP adresy, je veľmi jednoduché a v mnohých prípadoch postačujúce nasadiť démona bandwidthd. Tento démon po spustení prepne definované sieťové rozhranie do promiskuitného režimu a pre vybrané rozsahy IP adries vypracováva podrobné

štatistiky. Prezentuje ich podobne ako MRTG vo forme HTML stránok.

Keďže firewallové systémy bývajú umiestnené na rozhraní dvoch sietí, znamená výpadok linky na takomto systéme obmedzenie konektivity medzi týmito sieťami. Preto je nutné v prípade odpojenia sieťového kábla okamžite informovať správcu. Túto úlohu dostačujúco plní démon Ifplugd, ktorý môže v prípade akéhokoľvek výpadku napríklad zaslať správcovi SMS prostredníctvom pripojeného mobilného telefónu.

### 4.3.4 IDS a vytvorenie fiktívnej reality – Snort, BASE a inetd

Do navrhovaného firewallového systému som sa rozhodol zaradiť aj známy systém na detekciu prienikov zvaný SNORT, ktorý kombinuje výhody viacerých typov IDS najmä vďaka tomu, že používa vlastný jazyk na definovanie pravidiel. Aktuálne pravidlá sú dostupné pre registrovaných používateľov zadarmo priamo z domovskej stránky tohto projektu. Na uľahčenie jeho správy a prehliadania ním detekovaných udalostí som zaradil do systému aj program BASE *(Basic Analysis and Security Engine)*, ktorý s používateľom komunikuje prostredníctvom webového rozhrania. Aby bolo možné používať BASE, musí SNORT generované udalosti a výstrahy zapisovať do databázy, ktorú v tomto prípade sprostredkúva SRBD *(Systém Riadenia Bázy Dát)* MySQL, ktorý je štandardnou súčasťou distribúcie Slackware.

V prípade, že by z nejakého dôvodu nebolo na firewallovom systéme možné zaviesť SNORT, je dobré otvoriť aspoň pár tzv. hluchých TCP portov a využívať ich na získavanie dát pre adaptívny firewall. Toho je možné docieliť vhodnou konfiguráciou démona inetd, ktorý dokáže otvoriť ľubovoľný port, očakávať na ňom požiadavky a následne ich odovzdať na ďalšiu kontrolu démonovi TCPd, ktorý je súčasťou TCP wrappers. Ak niektorý z parametrov prichádzajúceho spojenia, či už je to (hluchý) cieľový port alebo zdrojová IP adresa, zapríčiní, že spojenie nebude prepustené na ďalšie spracovanie do systému, môže TCPd vykonať ľubovoľný príkaz definovaný direktívou spawn v súbore */etc/hosts.deny*. Ak je týmto príkazom pravidlo firewallu vytvorené pomocou iptables, môžeme hovoriť o adaptívnom firewalle. Bob Toxen v knihe *„Bezpečnost v Linuxu: Prevence a odvracení napadení systému“* uvádza, že táto technika v sieťach jeho klientov spoľahlivo dokázala, že je schopná naozaj veľmi efektívne odraziť útok crackerov, ale zároveň nijako nebráni prístupu „poctivých“ používateľov k ponúkaným službám. Pri správnej konfigurácii odrazí mnohé pokusy

crackerov o prístup k podporovanej službe z iného systému, než z ktorého majú klienti povolené pripojenie. Jeden ním spravovaný systém zhromaždil vo svojom firewallovom subsystéme (s procesorom Pentium III na frekvencii 800 Mhz) 4000 pravidiel pre netfilter, ale právoplatné pakety stále prepúšťal za menej než jednu milisekundu. [1]

### 4.3.5 Monitorovanie stavu služieb – slackkeeper

Už dlhšiu dobu som sa intenzívne pokúšal nájsť nástroj, ktorý by bol schopný monitorovať a v prípade pádu štartovať jednotlivé služby v systéme GNU/Linux. Objavil som síce sadu programov s názvom daemontools, ktoré by sa na tento účel dali použiť, no po vzhliadnutí zdrojového kódu som sa ich radšej ani nepokúšal nasadiť. Sú dielom matematika menom D.J. Bernstein, ktorý by sa dal podľa môjho skromného názoru označiť minimálne ako extrémista ak nie až ako anarchista, pretože neuznáva GNU autoconf ani balíčkovacie systémy a tomu prispôsobuje aj zdrojové kódy svojich projektov.

Po nedobrovoľnom akceptovaní reálneho stavu som sa rozhodol, že na vykonávanie tejto dôležitej činnosti vytvorím skript pre bourne shell a nazvem ho slackkeeper. Po spustení vyhľadá všetky súbory s príponou *.watch* v adresári */etc/slackkeeper* a načíta z nich informácie o jednotlivých procesoch, ktoré má monitorovať. Každý „watch“ súbor obsahuje minimálne dve premenné *process* a *start*. Obsahom prvej premennej je text, obyčajne názov procesu, ktorý má slackkeeper hľadať vo výpise procesov. Ak sa tento text vyskytuje aspoň na jednom riadku vo výpise získanom pomocou utility ps, tak sa služba považuje za spustenú. Ak sa tento text nenájde, tak slackkeeper spustí službu príkazom, ktorý je obsahom premennej *start*. Slackkeeper je potrebné periodicky spúšťať pomocou démona cron. Definičné „watch“ súbory môžu obsahovať aj celočíselné premenné *time* a *attempts*. Počet pokusov, ktoré môže slackkeeper využiť na spustenie procesu je definovaný premennou *attempts* a jednotlivé pokusy nasledujú po čase definovanom premennou *time*. Každý neúspešný pokus o spustenie služby tento program zapisuje do systémových log súborov. Ak sa mu činnosť služby nepodarí obnoviť po definovanom počte pokusov, vzdá sa a spustí posledný príkaz definovaný premennou *panic* v hlavnom konfiguračnom súbore */etc/slackkeeper/slackkeeper.conf*.

Slackkepper ponúka aj voliteľnú možnosť oznamovania o behu procesov, ktorých

názvy nie sú uvedené v konfiguračnom súbore */etc/slackkeeper/processes.conf*. Takto je možné monitorovať, či v systéme nebeží nejaký neznámy proces, ktorým môže byť napríklad trójsky kôň. Samozrejme, že cracker môže úpravou jadra alebo utility ps docieliť, aby jeho procesy neboli zobrazované, no takáto úprava by mala byť okamžite spozorovaná počas kontroly integrity súborového systému.

### 4.3.6 Analýza log súborov – logiq

Súčasťou mnohých distribúcií systému GNU/Linux je program na analýzu log súborov zvaný logcheck/logsentry. Tento program je schopný vyhľadávať v log súboroch riadky, v ktorých sa nachádza jedno alebo viaceré z množiny definovaných „zlých“ slov. Mne osobne však prekáža jeho obmedzená konfigurovateľnosť, a preto som sa opäť vybral vlastnou cestou a vytvoril som nástroj logiq. Logiq je podobne ako logcheck schopný z log súborov systému generovať reporty obsahujúce len riadky, ktoré spĺňajú kritéria zaujímavosti. Jednotlivé vyhľadávané slová sú uvedené v súbore */etc/logiq/words.alert*. Vo výnimočných situáciách sa stáva, že sa tieto slová vyskytujú aj v bezvýznamných riadkoch. Preto súbor */etc/logiq/words.ignore* obsahuje zoznam slov, ktoré sa hľadajú v „zlých“ riadkoch a ak sa v niektorom z nich nájde aspoň jedno slovo z tohto súboru, bude riadok z finálneho reportu vypustený. Logiq tiež necháva administrátorovi plnú kontrolu nad tým, ktoré súbory nechá programu kontrolovať. Spôsob konfigurácie tejto možnosti je takmer identický s konfiguráciou programu slackkeeper. Pre každý jeden kontrolovaný log súbor je nutné vytvoriť konfiguračný súbor s príponou .logiq v adresári */etc/logiq*. Program logiq však ide ešte ďalej a dovoľuje pre každý log súbor definovať zoznam „zlých“ a „ignorovaných“ slov. Logiq jednotlivé súbory logov dokáže aj zálohovať a k zálohe pribaliť aj vygenerovaný report s vybratými riadkami. Tieto reporty si samozrejme správca môže nechať zasielať aj elektronickou poštou.

### 4.3.7 Kontrola integrity súborového systému – md5deep

Snáď najznámejším nástrojom na kontrolu integrity súborového systému je komerčný produkt tripwire od rovnomennej firmy Tripwire, Inc. Jeho zdrojové kódy sú síce voľne dostupné, no mnoho ľudí ho považuje za používateľsky neprívetivý. To viedlo k vzniku viacerých projektov, ktorých cieľom je k nemu vyvinúť rovnako silnú

alternatívu. Ja osobne preferujem nástroj md5deep, ktorý pomocou rôznych hašovacích funkcií generuje kontrolné súčty pre definované súbory alebo rekurzívne pre celé adresáre. Je dobré hneď po inštalácii firewallového systému vytvoriť referenčnú databázu kontrolných súčtov tzv. etalón, ktorý je treba umiestniť na pamäťové médium, z ktorého sa dá iba čítať. Systém tak môže automaticky každú noc porovnať kontrolné súčty jednotlivých súborov a prípadné zmeny oznámiť správcovi.

# 5 INŠTALAČNÝ PROGRAM FIRESLACK

Kapitola bola z verzie web release vypustená.

# 6 IMPLEMENTÁCIA NAVRHNUTÉHO FIREWALLOVÉHO SYSTÉMU

Kapitola bola z verzie web release vypustená.

# ZÁVER

Som zástancom kvalitných firewallových systémov a myslím si, že firewallový systém, či už sieťový alebo hostový, by mal chrániť každý počítač pripojený k sieti. No zdá sa mi, že až príliš často počúvam vetu „Máme firewall, sme v bezpečí!“. Mnohí ľudia si totiž pod pojmom firewall asi predstavujú niečo, čo vytvorí v ich sieti alebo na ich počítačovom systéme absolútnu bezpečnosť. A sú ochotní sa so mnou neuveriteľne zanietene hádať, keď sa im pokúšam vysvetliť, že firewall môže znížiť iba niektoré riziká, ktoré v sieti číhajú. Stretol som sa dokonca aj s takými, ktorí na operačnom systéme Windows používali dva rôzne personálne firewally v predstavách, že ich to lepšie ochráni. Som presvedčený, že tieto utkvelé predstavy sú pôvodom nevedomosti. Jedine v prípade, že nemám ani tušenia ako funguje stavový firewall si môžem myslieť, že ma ochráni pred všetkými útokmi a nemôže sa mi nič stať. S obľubou pozorujem reakcie jednotlivcov, keď im názorne vysvetľujem, že kde niet cesty „zvonku dovnútra“, tam môže existovať cesta „zvnútra von“ a tá je väčšinou voľne priechodná. Ak sa mi nejakým spôsobom podarí dostať môj program do systému obete, existuje len minimálne riziko, že mu bude zabránené iniciovať spojenie na môj počítačový systém. Ak navyše bude schopný obsluhovať tunelované spojenia a zároveň spustí na počítači obete akýkoľvek druh vzdialenej správy, tak mi už nikto a nič nezabráni v tom, aby som nad ním prebral plnú kontrolu. Možno je namieste otázka, že načo je firewallový systém dobrý. Určite minimálne na kontrolu stavu spojení, zastavenie priamych pokusov o pripojenie prichádzajúcich z vonkajších sietí a na vykonávanie rôznych druhov monitoringu. Tiež môže byť veľmi nápomocný pri rozširovaní alebo diferencovaní siete.

Firewallový systém treba chápať ako nutnú súčasť rozsiahlej bezpečnostnej politiky organizácie, ktorá musí definovať presné postupy a pravidlá pre rôzne činnosti a udalosti. Je to len ďalšia bezpečnostná bariéra, ktorá pomáha chrániť systémy v organizácii. Je takou istou bariérou ako prísny vrátnik, ktorý odmietne vpustiť do budovy neznáme osoby, ako mreže na oknách, ktoré zabránia vniknutiu zlodejov, ako kvalitný alarm, ktorý upozorní bezpečnostnú službu na nepovolený pohyb alebo bezpečnostné kamery, ktoré zaznamenajú akýkoľvek prístup do chránených miestností.

Sieťová bezpečnosť nesmie byť podceňovaná a rovnako tak ani fyzická alebo

systémová. Je síce možné pomerne presne určiť, ktoré prvky a činnosti spadajú do sieťovej bezpečnosti a ktoré do systémovej, no nikdy ich nie je možné od seba oddeliť, pretože úzko spolu súvisia. Bez prísneho dodržiavania pravidiel systémovej bezpečnosti nemožno ani len uvažovať nad sieťovou bezpečnosťou. Radoví používatelia počítačov si myslia, že keď im správca odmietne zaviesť internet alebo nainštalovať nejaký program, že je lenivý a neochotný. Radi operujú s výrokom, že počítače majú slúžiť ľudom a nie ľudia počítačom. Keby však poznali všetky bezpečnostné problémy, ktoré môže pripojenie ich systému k internetu priniesť, pravdepodobne by si búchali hlavu o stenu. Preto by som spomínaný výrok rád doplnil a odpovedal, že počítače majú slúžiť ľudom, ktorých prioritnou úlohou je slúžiť zamestnávateľovi. Ak je prístup na internet alebo nejaký program pre plnenie pracovných úloh zamestnanca nevyhnutný, určite budú zvážené z toho vyplývajúce riziká, podniknuté opatrenia pre ich minimalizáciu a následne bude zamestnancovi umožnené ich využívať.

Aj keď som v celej práci postupne uvádzal mnoho pádnych dôvodov, prečo nepracujem so špecializovanými firewallovými distribúciami, musím spomenúť ešte jeden. Zastávam názor, že ak má človek k dispozícii tzv. „all-in-one“ riešenie, nebude sa snažiť objavovať podstatu vecí. Ako ale môže vykonávať dobre svoju prácu, ak nechápe základné princípy? Ako bude taký človek reagovať v kritických situáciách? Bude schopný vybrať najlepšie riešenie?

Serverové systémy používajúce operačný systém GNU/Linux už aktívne nasadzujem v produkčnom prostredí viac než dva roky. Za túto dobu som si všimol, že v porovnaní s inými systémami postráda linuxové jadro niektoré základné vlastnosti užitočné najmä v sieťovom prostredí. Napríklad nevygeneruje udalosť pre syslog pri výpadku sieťového kábla. Tento nedostatok však skvele odstraňuje v práci opisovaný démon IFplugd, ktorý je navyše pri tejto udalosti schopný spustiť akýkoľvek program. Dodnes sa mi však nepodarilo vyriešiť nedostatok spôsobujúci, že sa v systémových log súboroch pri konflikte IP adresy s iným systémom na sieti neobjaví ani len pol slova, ktoré by signalizovalo túto dôležitú udalosť. Nie je nič nezvyčajné, keď sa „vyspelejší“ používatelia pokúšajú získať IP adresu serverového systému. Ako však má správca proti tomu bojovať, keď o tom ani nebude vedieť? Tento problém som pôvodne plánoval vyriešiť pri realizácii tejto diplomovej práce, no objavil som iba neoficiálny patch pre linuxové jadro s názvom arppatch, ktorý napísal Marc Merlin – dlhoročný správca linuxových systémov z USA. Autor sa už viackrát pokúšal presadiť jeho zahrnutie do

oficiálnych zdrojových kódov jadra, no vždy bol odmietnutý s odôvodnením, že o túto činnosť sa môže starať používateľská aplikácia. [7]

Kontaktoval som teda pána Merlina a žiadal som ho o radu, či vie o existencii takejto aplikácie. V priebehu pár minút som dostal mailom odpoveď, že bohužiaľ takú aplikáciu nepozná, ale jej napísanie by nemalo byť príliš komplikované. Rozhodol som sa teda, že ju vytvorím, no kvôli nedostatku času som to nebol schopný zrealizovať ešte pred odovzdaním tejto práce. Zaradenie tohto programu do navrhnutého firewallového systému považujem za nevyhnutné. Na detekciu tejto udalosti by som chcel použiť rovnakú techniku, ako používa arppatrch Marca Merlina, no aby som to bol schopný realizovať formou používateľskej aplikácie, budem pravdepodobne nútený prepnúť sieťové rozhranie do promiskuitného režimu.

Pri zostavovaní firewallového systému som vytvoril aj aplikáciu na monitorovanie stavu služieb s názvom Slackkeeper, ktorej opisu som sa podrobne venoval v kapitole 4.3.5. Názov tejto aplikácie síce napovedá, že má veľa spoločného s distribúciou Slackware, no opak je pravdou. Táto aplikácia je úplne nezávislá na použitej distribúcii a mala by po miernych úpravách fungovať aj na ďalších unixových systémoch. Určite aspoň na BSD systémoch a Solarise. Hneď ako to bude možné, začnem pracovať na jej zdokonalení a uvoľním ju pod licenciou GNU/GPL, pretože už v týchto ranných fázach prejavili dvaja moji známi o ňu záujem. Budem však nútený zmeniť jej meno, aby neodrádzalo používateľov iných distribúcií.

Som presvedčený, že firewallové systémy, ktorých inštaláciu som realizoval pomocou vytvoreného inštalačného nástroja potrebujú ešte jeden dôležitý prvok. Je ním centralizovaná správa MAC adries počítačov z lokálnej siete, ktorá by priamo ovplyvňovala nastavenie statických ARP záznamov systému, nastavenie DHCP démona a tiež pravidlá firewallu, ktoré by prepúšťali len pakety so správnou dvojicou IP a MAC adries. V súčasnom stave treba pri každej zmene aktualizovať údaje v systéme až na troch miestach. Bolo by teda vhodné vytvoriť nástroj, ktorý by používal napríklad webové rozhranie a pomocou neho by sa aktualizovali záznamy v relačnej databáze. Tá by mohla obsahovať aj ďalšie údaje ako napríklad meno osoby používajúcej systém a mnohé iné. Som však realista a myslím si, že na vytvorenie takéhoto nástroja nebudem mať dostatok voľného času. Zvažujem však možnosť, že jeho realizáciu navrhnem niektorému z mladších študentov ako tému bakalárskej práce a osobne dohliadnem na

jeho vývoj.

Hlavný prínos mojej diplomovej práce vidím v tom, že sa mi podarilo pomerne prehľadne zhrnúť do jedného dokumentu prevažnú časť mojich vedomostí o bezpečnosti unixových systémov a vytvoriť inštalačný program, ktorý mi podstatne skráti čas potrebný na vytvorenie ďalších komplexných firewallových systémov. Dodnes som ich s jeho pomocou vytvoril päť a dva z nich som podrobne opísal v práci. Text tejto práce bude slúžiť aj ako učebná pomôcka pre účastníkov mnou organizovaných dobrovoľných kurzov zameraných na využitie systémov GNU/Linux a OpenBSD v počítačových sieťach.

## ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

[1] TOXEN, B. Bezpečnost v Linuxu: Prevence a odvracení napadení systému, Brno: Computer Press, 2003, 849s., ISBN 80-226-716-7

[2] ARTYMIAK, J. Building firewalls with OpenBSD and PF, Warszawa: Sowa, 2003, 247s., ISBN 83-916651-4-3

[3] WIKIPEDIA CONTRIBUTORS. Virtual private network [online]. Wikipedia, The Free Encyclopedia; 18.04.2006, 11:52 UTC [Cit. 18.04.2006]. Dostupné na: <http://en.wikipedia.org/w/index.php?title=Virtual_private_network&oldid=48989312>

[4] WIKIPEDIA CONTRIBUTORS. Simple Network Management Protocol [online]. Wikipedia, The Free Encyclopedia; 18.04.2006, 18:35 UTC [Cit. 18.04.2006]. Dostupné na: <http://en.wikipedia.org/w/index.php?title=Simple_Network_Management_Protocol&oldid=49036184>

[5] WIKIPEDIA CONTRIBUTORS. Intrusion-detection system [online]. Wikipedia, The Free Encyclopedia; 12.04.2006, 16:32 UTC [Cit. 18.04.2006]. Dostupné na: <http://en.wikipedia.org/w/index.php?title=Intrusion-detection_system&oldid=48134444>

[6] OpenBSD. Firewall Redundancy with CARP and pfsync [online] OpenBSD; 11.11.2005, 02:22:50 UTC [Cit. 18.04.2006]. Dostupné na: <http://www.openbsd.org/faq/pf/carp.html>

[7] MERLIN, M. Layer 2, MAC addresses, ARP, and Duplicate IP Detection. [online] Merlins.org; 26.01.2006 19:08 UTC [Cit. 18.4.2006] Dostupné na: <http://marc.merlins.org/linux/talks/Layer2&ARP/talk-src/Layer2&ARP.pdf>