UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Centrální protokolovací systém

Bc. Michal Seifert

Diplomová práce
2011

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2010/2011**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Michal SEIFERT**
Osobní číslo:	**I08381**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Centrální protokolovací systém**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Zásady pro zpracování (cíl, obsah teoretické a implementační části): Bude proveden bezpečnostní audit dané sítě. Budou zhodnocena bezpečnostní rizika a rizika možných příčin ztráty dat. Bude provedena také analýza dostupnost provozovaných služeb z hlediska možných výpadků. Bodou navržena opatření směřující k odstranění nalezených slabin systému: * Bude zajištěno automatické zálohování potřebných dat, * bude navržen a zprovozněn systém sledování dostupnosti služeb provozovaných v dané síti, * bude navržen a zprovozněn centrální protokolovací systém, který bude shromažďovat a analyzovat data o službách v síti a který bude problémy sám řešit (např. restart služby) nebo upozorní správce. * Bude zajištěn bezpečný vzdálený přístup ke správě.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

*** VESELSKÝ, Jiří, et al. Linux – Dokumentační projekt [online]. 3. aktualiz. vyd. Brno: Computer Press, 2003. 1020 s. Dostupný z WWW: <http://knihy.cpress.cz/DataFiles/Book/00000675/Download/K0819.pdf>. * GRAHAM, Steven; SHAH, Steve. Administrace systému Linux. Překlad třetího vydání. Praha: Grada, 2007. 550 s. * SOBELL, Mark G. Mistrovství v Linuxu: Příkazový řádek, shell, programování. Vyd. 1. Brno: Computer Press, 2007. 517 s. * RANKIN, Kyle. Linux Knoppix na maximum. Vyd. 1. Brno: Computer Press, 2006. 298 s.**

Vedoucí diplomové práce: **Mgr. Tomáš Hudec**
Katedra informačních technologií

Datum zadání diplomové práce: **27. října 2010**

Termín odevzdání diplomové práce: **20. května 2011**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

doc. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 3. listopadu 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. května 2011 Michal Seifert

Poděkování

Tímto bych chtěl velice poděkovat panu Mgr. Tomáši Hudcovi za cenný čas strávený při konzultacích a odborné rady, které mi v průběhu studia poskytl.

Zvláštní poděkování patří také P. Šafránkové a S. Seifertové za vytrvalou podporu a asistenci při dokončování této práce.

Anotace

Práce se zabývá bezpečným uchováním multimediálních dat na GNU/Linux systémech. V analýze jsou odhaleny slabiny prověřovaného řešení, ke kterým jsou navrhnuta opatření vedoucí k odstranění problémů.

Praktická část implementuje zabezpečení běžících služeb, zálohování uživatelských dat a operačních systémů. Pro dohled nad celou sítí byly zavedeny monitorovací systémy, systém pro detekci vniknutí a centrální sběr protokolů. Vzdálená správa byla zajištěna s důrazem na bezpečnost a dostupnost připojení.

Klíčová slova

Unix, GNU/Linux, systém, protokol, sběr, montoring, zabezpečení, rsync, ssh, mon, zabbix

Title

Central Protocol System

Annotation

The thesis deals with secure multimedia data storage in GNU/Linux systems. The analysis reveals the drawbacks of the solution under verification, proposing measures to eliminate them.

The practical part implements the security of running services, the back-up of user data and operating systems. For the purpose of control over the network, monitoring systems, an intrusion detection system and central protocol collection were introduced. Remote administration was ensured with an emphasis on security and connection availability.

Keywords

Unix, GNU/Linux, system, protocol, collection, monitoring, security, rsync, ssh, mon, zabbix

Obsah

Úvod **12**
1. Teoretický rozbor **13**
1.1 Lokální organizace dat 13
1.2 Síťové souborové systémy 15
1.3 Zálohovací systémy 18
1.4 Monitoring systémů a sítí 21
1.5 Zabezpečení GNU/Linux 26
2. Analýza **31**
2.1 Charakteristika klienta a jeho potřeb 31
2.2 Struktura sítě 31
2.3 Porty 33
2.4 Bezpečnost provozovaných služeb 35
2.5 Dostupnost klíčových služeb 38
2.6 Bezpečnost systémů a uchování dat 38
3. Implementace **41**
3.1 Hromadná distribuce příkazů 41
3.2 Aktualizace systému 42
3.3 Zabezpečení služeb 43
3.4 Záloha dat 49
3.5 Dohledový systém 54
3.6 Systém pro detekci útoků 59
3.7 Sběr systémových protokolů 60
3.8 Vzdálená správa 61
Závěr **63**
Použitá literatura **64**

Seznam obrázků

Obr. 1: LVM....15
Obr. 2: Analyzovaná síť....32
Obr. 3: Návrh restrukturalizace sítě....40
Obr. 4: SSL Server Test - před úpravami....48
Obr. 5: SSL Server Test - po úpravách....48
Obr. 6: Dostupnost záložních serverů....50
Obr. 7: Dohledový systém mon....57
Obr. 8: Dohledový systém zabbix....58

Seznam tabulek

Tabulka 1: Otevřené porty.....34

Seznam zkratek

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
WWW	World Wide Web
SSL	Secure Socket Layer
HTML	HyperText Mark Language
GNU	GNU's Not Unix
md	Multiple Devices
LVM	Logical Volume Manager
FTP	File Transfer Protocol
S. M. A. R. T.	Self-Monitoring, Analysis, and Reporting Technology
NFS	Network File System
iSCSI	Internet Small Computer System Interface
LAN	Local Area Network
MAN	Metropolitan Area Network
WAN	Wide Area Network
RAID	Redundant Array of Independent Disks
NIDS	Network Intrusion Detection System
HIDS	Host Intrusion Detection System
IDS	Intrusion Detection System
TSL	Transport Layer Security
tar	Tape Archiver
CPU	Central Processing Unit
SATA	Serial ATA
AoE	ATA over Ethernet
SFTP	Secure File Transfer Protocol
LDAP	Lightweight Directory Access Protocol
CA	Certification Authority
DMZ	Demilitarized Zone
GPL	General Public License

Úvod

V běžném životě se na každém kroku využívají systémy, které produkují data, vyměňují si mezi sebou nejrůznější informace a nakonec vše někam zaznamenávají. Potřeba mít bezpečně uložené informace v digitální podobě čím dál víc narůstá a v budoucnu tomu nebude jinak. Bezpečnost uchovaných dat je závislá na celé skupině faktorů vycházejících z hardwarových vlastností médií, přes softwarové vybavení, až po práci koncového uživatele.

Společnost 24SNAiLS, a. s., která se pohybuje v oblasti data managementu bezmála deset let, se touto problematikou zabývá od svého počátku. Nadměrná produkce multimediálního obsahu a jeho uchování v nejvyšších kvalitách si žádá specifický přístup. Standardní řešení bývají nevyhovující především pro nedostatečnou kapacitu a nízkou spolehlivost. Naopak komerčně nabízené profesionální modely, která by vyhovovaly svou charakteristikou a výkonem, jsou finančně mimo možnosti menší až středně velké firmy.

Stále se zvyšující nároky na objem uchovávaných dat a jejich důležitost pro společnosti 24SNAiLS, a. s., daly téma pro tuto diplomovou práci. Základním předpokladem je navázat na nynější data management jedné z provozovaných sítí, která již nevyhovuje aktuálním požadavkům. Analyzovat bezpečnost uchování dat a navrhnout opatření, která povedou k odstranění nalezených slabin.

Souborové servery pracují se standardními disky v operačním systému GNU/Linux. Výsledné řešení by mělo nejen zvýšit bezpečnost uchování dat, ale také předcházet možným komplikacím jak na úrovni lidské chyby, tak v případě fyzického selhání hardware nebo hackerského útoku.

Zamezení ztrátě dat vinou hardwarového selhání bude zajištěno kontrolou kondice disků a stavu diskových polí. Chyby způsobené lidským faktorem budou eliminovány restriktivní politikou výchozích práv a pravidelným zálohováním. Vnitřní síť bude restrukturalizována tak, aby lépe odolala neoprávněnému vniknutí a upozornila pověřené osoby na podezřelé chování. Veřejně dostupné služby jako FTP, WWW a SSH budou zabezpečeny dle charakteru jejich použití. Dostupnost systémů a na nich provozovaných služeb bude monitorovat centrální dohledový systém, který dokáže upozornit správce na vzniklé problémy. Bude také zaveden centrální sběr systémových protokolů, který poskytne důležité informace pro hloubkovou analýzu systémů. Implementace se bude opírat o nástroje open source v kombinaci se skripty.

1. Teoretický rozbor

O uložená data je nutné se starat a tuto péči rozložit na všechny úrovně, které je mohou ovlivnit. Prvním krokem by mělo být ocenění dat a dostupnosti poskytovaných služeb. Podle těchto hodnot vybrat vhodný kompromis mezi cenou a použitou technologií. Celou problematiku je možné ilustrovat na vybudování bezpečnostní pyramidy:

- Základ by tvořil použitý hardware a ochrana před jeho selháním.
- Na úrovni systému jsou stavebními kameny:
  - vhodný souborový systém, od kterého se očekává, že bude nejen rychlý, ale i stabilní,
  - zálohovací mechanismus,
  - nastavené oprávnění pro uživatele.
- Z hlediska vzdáleného přístupu je nutná dostupnost nabízených služeb.
- Udržení stavu celé pyramidy by obstarávaly monitorovací nástroje, které v případě výskytu problémů zajistí potřebné úkony vedoucí k jejich odstranění.

Na takto postaveném modelu lze vybudovat stabilní systém u kterého bude možné garantovat stabilitu a dostupnost poskytovaných služeb. [1]

1.1 Lokální organizace dat

Standardně jsou data uložena na lokálním disku v pracovní stanici. Vzhledem k možnému selhání a nedostatečné kapacitě vznikl systém pro seskupování úložných zařízení do takzvaný polí RAID (Redundant Array of Independent Disks).

1.1.1 Hardwarový RAID

Hardwarový RAID je zajištěn v podobě přídavné karty, která má svůj vlastní procesor a paměť pro operace s polem. Takové řešení nabízí vysoký výkon a komfortní správu nezávislou na operačním systému. Naopak nepotěší jeho vysoká cena, možné problémy s obnovou pole při selhání samotného řadiče a omezené rozšíření o nové funkce a vlastnosti. [11]

1.1.2 Softwarový RAID

Softwarový RAID je tvořen na úrovni systému z obyčejně zapojených disků. V systému GNU/Linux se o tuto funkci stará ovladač *md* (Multiple Devices), jehož správu umožňuje program *mdadm*. Mezi kladné vlastnosti tohoto řešení patří přenositelnost vytvořeného pole na jiný hardware, podpora monitorovacích nástrojů na úrovni systému a žádné extra náklady na jeho zřízení a upgrade. Oproti hardwarovému řešení nedosahuje tak vysokých rychlostí a závisí na použitém operačním systému, který svou činností také zatěžuje. Pro větší počet disků tedy méně vhodné řešení. [8] [12]

1.1.3 Typy polí RAID

Při sestavení pole RAID je nutné zvolit jeho typ, který představuje kompromis mezi bezpečností, výkonem a užitnou kapacitou. V praxi se používají převážně pouze některé typy, například RAID 0, RAID 1, RAID 5, RAID 6 a RAID 10. Zbylé typy se využívají ojediněle. [9]

RAID 0

Pole v módu RAID 0 nepředstavuje žádný stupeň ochrany proti selhání disku a jeho výsledná velikost je součtem kapacit použitých disků. Vzhledem k tomu, že jsou data rozprostřena po všech discích, dosahuje vyšších rychlostí. V praxi se samotný RAID 0 objevuje jen zřídka, zejména kvůli riziku ztráty dat už při výpadku jediného disku.

RAID 1

RAID 1 naopak od pole RAID 0 slouží pro maximální redundanci dat. Disky v poli jsou navzájem zrcadleny. S rostoucím počtem disků stoupá jeho spolehlivost, kapacita zůstává stejná. Vhodné jako řešení pro dva disky nebo v kombinaci s RAID 0.

RAID 5

RAID 5 rozkládá paritní informace přes celé pole a tím umožňuje výpadek jednoho libovolného disku. Dopočítávání paritních informací při zápisu do pole má za následek jeho zpomalení. Naopak čtení je podobně rychlé jako RAID 0.

RAID 6

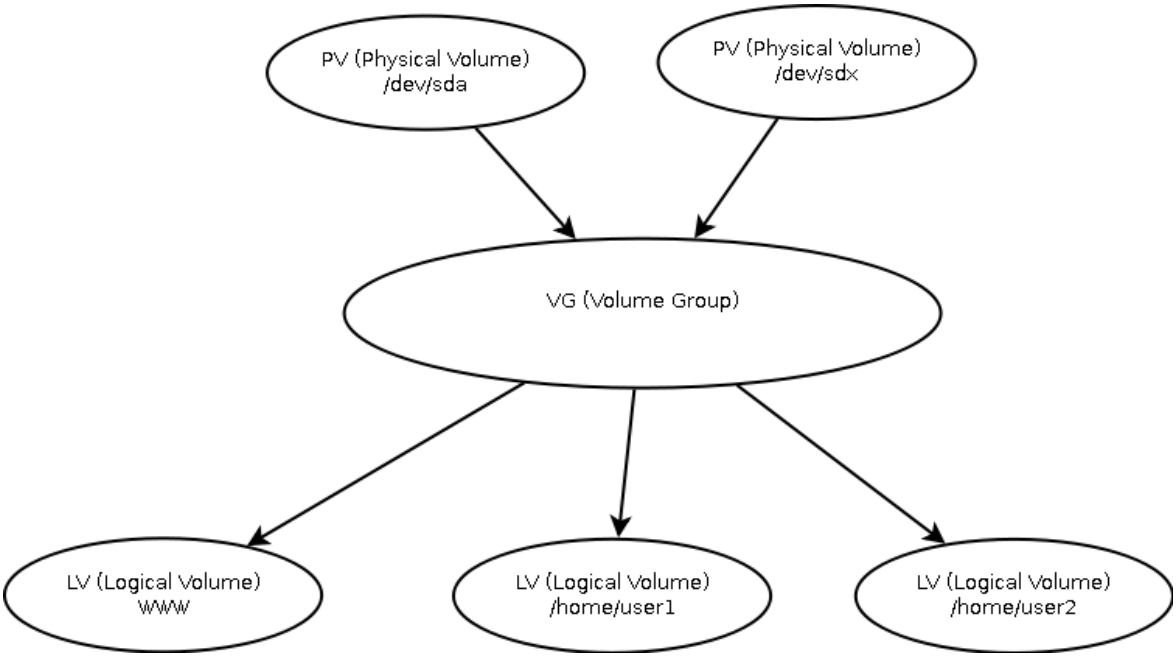
Obdobně jako RAID 5 rozprostírá paritní informace přes všechny disky pole. Navíc však vytváří ještě jednu nezávislou kopii, což má za následek možný výpadek až dvou libovolných disků.

RAID 10

Kombinuje vlastnosti RAID 0 a RAID 1 tak, že zrcadlí každý disk vytvořený v RAID 0. Tím se získá rychlost zápisu a čtení jako je u RAID 0 a redundance jako u RAID 1.

1.1.4 LVM

LVM (Logical Volume Manager) [10] nabízí oproti standardním nástrojům abstrakci nad úložnými zařízeními, které spojí do jednoho celku a ten opět dle potřeby přerozdělí. Díky této vlastnosti je možné za chodu libovolně měnit velikosti jednotlivých přípojných bodů a volit, přes jaká fyzická zařízení povede přidělený blok pamětí. U většiny implementací LVM je možné použít tzv snapshoty. Základní princip funkce LVM je znázorněn na obr. 1.



Obr. 1: LVM

LVM je implementováno na nejrůznějších serverových OS včetně GNU/Linux.

1.1.5 S. M. A. R. T.

S. M. A. R. T. [16] je zkratka pro monitorovací systém, který sleduje provozní hodnoty disku a porovnává je s přípustnými limity. V případě, že nějaký parametr překročí stanovený limit, je na to uživatel při testu upozorněn. Testy mohou probíhat buď plánovaně (BIOS, start systému) nebo neplánovaně, ručním spuštěním. Z roční analýzy dat [15] zveřejněné společností Google, patří mezi kritické parametry indikující vysoké riziko selhání disku:

- Snan Errors – chyby zjištěné při autodiagnostice,
- Realoced Sector Count – počet přealokovaných sektorů,
- Current Pending Sector Count – počet čekajících sektorů na přealokování.

1.2 Síťové souborové systémy

Pro přístup k datovému úložišti, umístěnému na lokální síti, je třeba připojení do dané sítě a síťový souborový systém, který zajistí samotnou výměnu dat. Oproti protokolu pro veřejné sdílení dat jako je FTP (File Transfer Protocol), jsou i zde kladeny požadavky na rychlost a co největší integraci do uživatelského operačního systému.

1.2.1 Centralizované souborové systémy

Centralizované systémy mají obvykle jeden server poskytující své služby na síti. Nevýhodou je, že takové řešení představuje kritické místo pro dostupnost služeb a škálovatelnost výkonu. Naopak vyniká svou jednoduchostí a širokou podporou u nejrůznějších operačních systémů.

SAMBA

Program `samba` [21] je svobodnou implementací protokolu SMB a CIFS. CIFS je novější verzí SMB, jehož úkolem je zajistit komunikaci a sdílení prostředků v sítích mezi systémy. Je podporován na systémech MS Windows i systémech unixových (GNU/Linux, MAC OS X, FreeBSD a další). Balík softwaru `samba` poskytuje několik základních služeb:

- autentizační nástroje,
- sdílení souborů,
- sdílení tiskáren,
- procházení sdílených zdrojů.

Nasazením `samby` jako primárního řadiče domény v kombinaci s OpenLDAP se vytvoří centrální autentizační systém nejen pro ostatní servery `samba` a klienty, ale mnoho dalších služeb.

NFS

NFS [22] představuje síťový souborový systém používaný převážně na unixových systémech. Vyniká svou jednoduchostí a rozšířeností. Umožňuje vzdálené připojování adresářů a disků tak, že pro uživatele se chovají stejně, jako by byly připojené lokálně. NFS se používá od roku 1989. Od té doby prošel mnoha vývojovými verzemi, které obsahovaly již v návrhu řadu bezpečnostních slabin. Poslední čtvrtá verze ale tyto nedostatky odstraňuje a přidává pokročilé funkce pro sdílení dat. Oproti `sambě` dosahuje větších přenosových rychlostí, ale chybí nativní podpora v MS Windows.

1.2.2 Distribuované souborové systémy

Distribuované souborové systémy [32] jsou tvořeny skupinou serverů (uzlů). Jejich úkolem je zprostředkovat vzdálený přístup k datům, která se uživateli tváří jako jeden celek uložený na lokálním disku. Jejich hlavní výhody oproti centralizovaným systémům jsou:

- rozložení zátěže mezi jednotlivé uzly,
- v případě výpadku jednoho nebo několika uzlů je datové úložiště stále dostupné.

Nevýhodou je potom složitost celého řešení, potřeba více serverů a tím i vyšší cena.

Fibre Channel

Fibre Channel [19] je standardizovaná technologie, která se používá pro vysokorychlostní přenos dat v síti. Nejčastěji je nasazována pro vytvoření SAN[1] (Storage Area Network). Vytvoření sdíleného úložiště na technologii Fibre Channel s sebou přináší benefity v podobě bezkonkurenčních přenosových rychlostí a vysoké dostupnosti. Naopak speciální hardwarové požadavky a nároky na obsluhu toto řešení natolik prodražují, že si jej mohou dovolit pouze velké podniky a instituce.

AFS

AFS [13] [17] [18] byl původně vyvíjen na Carnegie Mellon University a podporován firmou IBM Pittsburg Labs, která v roce 2000 uveřejnila zdrojové kódy v podobě Open AFS. AFS funguje na principu klient-server. Umožňuje ukládání stejných dat na různé souborové servery. Mezi nejpoužívanější implementace patří:

- Open AFS – často nasazován do ostrého provozu, stabilní, aplikace pro klienty a servery dostupné pro většinu operačních systémů včetně UNIX, GNU/Linux, Mac OS X a MS Windows. V aktuální verzi 1.5.78 stále chybí podpora pevných odkazů.
- CODA – nabízí spoustu vylepšení a inovativních funkcí, avšak na úkor stability. Zatím je využíván převážně na testovací účely.

iSCSI

Protokol iSCSI (Internet Small Computer System Interface) [14] je standardizovaný protokol SCSI upravený pro použití přes síť TCP/IP. Umožňuje spolehlivé doručení příkazů SCSI s využitím relativně nespolehlivé sítě LAN, MAN nebo WAN (Internet). Má podporu u většiny výrobců diskových polí a operačních systémů. Vzhledem k jeho komunikaci přes protokol TCP lze tento provoz snadno směrovat. Hlavní výhody představují:

- zajištění bezpečnosti přenášených dat
  - šifrování přes IPSec,
  - ochrana proti zfalšování identity CHAP protokolem,
- použitelnost na velké vzdálenosti,
- nevyžaduje žádný speciální hardware,
- nízké náklady oproti Fibre Channel.

Protokol iSCSI nachází uplatnění jako alternativa při konsolidaci dat pro malé a středně velké firmy, kde nejsou kladeny extrémní nároky na rychlosti datového úložiště. Jeho podpora u výrobců hardware pak umožní snadné propojování souborových serverů.

[1] SAN představuje oddělenou síť souborových serverů zajišťující přístup ke společně sdíleným datům.

AoE

AoE (ATA over Ethernet) [32] představuje nízkoúrovňový protokol, vyvinutý pro vysokorychlostní propojení SATA zařízení přes síť. Na rozdíl od iSCSI pracuje pod vrstvou TCP/IP, což snižuje zatížení CPU a znemožňuje jeho směrování. Zpřístupněné vzdálené blokové zařízení se tváří jako lokální disk, se kterým je možné dále pracovat v softwarovém RAID nebo LVM.

1.3 Zálohovací systémy

Zálohovací systémy [23] představují další bezpečnostní prvek pro ochranu dat. I přes použití redundantních polí RAID a nejlepších hardwarových řešení nelze zajistit, že uložená data budou v pořádku. Na vině může být chyba softwarová, fyzického zařízení a nebo selhání lidského faktoru. Archivovací systém tato rizika odstraňuje periodickým a automatizovaným vytvářením záloh.

1.3.1 Způsoby zálohování

Před samotným zálohovacím procesem [24] je důležité zvolit správnou strategii, která by měla odrážet požadavky na aktuálnost, dostupnost, bezpečnost, obnovitelnost a maximální délku archivace zálohovaných dat.

Úplná záloha

Úplná záloha spočívá [24] ve vytvoření identické a nezávislé kopie zálohovaných dat. Způsobů, jak toho docílit, existuje několik:

- Záloha na úrovni souborového systému a dat – šetří místo, protože se nemusí přenášet nevyužitý prostor na disku, ale pouze data. Výsledná podoba zálohy je buď identická kopie adresářů, souborů nebo celých diskových oddílů. V případě, že není třeba k záloze přistupovat okamžitě, je možné ji nasměrovat do jediného souboru a použitím komprese minimalizovat paměťovou náročnost. Výhodné řešení pro vytváření online i offline datových záloh.
- Nízkoúrovňová záloha – kopíruje oddíly, disk nebo celé diskové pole na úrovni paměťových bloků. To znemožňuje efektivní kompresi prázdného místa. Méně vhodná varianta na zálohy obyčejných dat. Naopak pro přenesení operačního systému z disku na disk nebo jeho obnovu, je to velice rychlý a bezproblémový způsob. Odpadají problémy s obnovou FAT tabulky nebo oprávněním.

Dvojitá a vícenásobná záloha

U vícenásobného zálohování [24] se provádí ty samé úkony jako při plné záloze s tím rozdílem, že je vytvořeno více verzí záloh. V závislosti na důležitosti dat je pak možné vytvářet dvě a více verzí. Například pokud by byla originální data omylem smazána a na chybu se přišlo až druhý den, tak by každodenní záloha nijak ztrátě nezabránila. Naopak při vícenásobném zálohování by bylo možné se vrátit nejen k předchozímu dni, ale

třeba o týden nazpět a tam smazaná data najít. Nevýhoda tohoto řešení je ve velké paměťové náročnosti, kdy každá další verze zálohy vyžaduje 100% velikosti archivovaných dat.

Přírůstková záloha

Přírůstková záloha [24] vytvoří při prvním spuštění kompletní kopii zálohovaných dat. Každá další záloha už jen přidává změny, které nastaly od poslední verze. Výhodné použití tam, kde je nutné udržovat velké množství revizí archivovaných dat. S délkou historie roste i paměťová náročnost, avšak pouze o velikost provedených změn. Pro obnovu je vždy třeba kompletní kopie a revize předchozích změn. Z těchto informací se sestaví výsledná podoba zazálohovaných dat. Pokud jsou informace v jednom z přírůstků poškozeny, není možné kompletně obnovit žádný z následujících.

Rozdílová záloha

U rozdílové zálohy [24] představuje základ opět plná záloha. Každá další verze obsahuje pouze diferenci oproti plné záloze. Pro obnovení dat je potřeba mít úplnou a rozdílovou část zálohy. Poškození jedné z rozdílových záloh nemá žádný vliv na ostatní.

1.3.2 Zálohovací nástroje

Tento přehled [25] [26] se zaměřuje na ověřené nástroje open source nástroje používané v systémech GNU/Linux. Celou problematiku lze rozdělit do dvou skupin. Na jednoduché programy, které v propojení s ostatním systémovým vybavením zajistí tvárné a efektivní řešení, nebo komplexní zálohovací systémy s větší nabídkou voleb a horší přizpůsobitelností se speciálním požadavkům. Zvolená kombinace pro archivaci by měla opět vycházet z vhodně zvolené strategie a požadavků na zálohování.

tar

Program `tar` (Tape Archiver) [27] představuje jeden z nejstarších archivačních nástrojů. Je dostupný na většině systémů unixového typu, tedy i GNU/Linux. Původně byl navržen pro zálohu na magnetické pásky. Funkce `tar` spočívá pouze v přenosu souborové struktury a oprávnění do jediného souboru nebo zařízení a naopak. Pro podporu komprese lze využít nejrůznější standardně dostupné programy jako `bzip2`, `lzma`, `compress` nebo `gzip`. Uplatnění najde zejména při rychlé archivaci a nebo jako základní stavební prvek v zálohovacích skriptech.

rsync

Již z názvu je patrné, že `rsync` [28] zajišťuje nějaký druh synchronizace, konkrétně synchronizaci souborů a složek. Umožňuje jak lokální, tak i vzdálenou kontrolu a výměnu dat. Před zahájením procesu nejprve použije algoritmus na porovnání změn ve zdroji a cíli. V případě novějších zdrojových souborů zahájí jejich přenos. Charakteristické vlastnosti `rsync` jsou:

- podpora kopírování odkazů, zařízení a uživatelských oprávnění,
- vyloučení nechtěných souborů a adresářů,
- podpora vzdálených shellů `ssh` nebo `rsh` zajišťujících bezpečný přenos,
- nevyžaduje práva uživatele `root`, což má za následek menší bezpečnostní riziko,
- přenášení jen změn v souborech,
- komprese přenášených dat.

Program `rsync` představuje jednoduchý a rychlý nástroj pro inteligentní a bezpečný přenos. Díky tomu bývá hojně nasazován jako samotný zálohovací systém, ale rovněž i jako základ robustnějších systémů.

dd

Program `dd` [25] kopíruje a konvertuje soubor dle zadaných parametrů. V linuxovém pojetí jsou zařízení prezentována jako speciální soubor, tedy i na disk nebo jeho oddíly lze nahlížet jako na soubory. Této vlastnosti lze využít pro tvorbu speciálních záloh programem `dd`. Hlavním rozdílem oproti standardnímu zálohovacímu softwaru je to, že kopírování probíhá na nejnižší úrovni. Tím odpadají problémy s:

- rychlostí obnovy,
- nekompatibilitou souborových systémů,
- zachováním atributů a speciálních oprávnění,
- zálohou zavaděče systému a tabulky rozdělení disku.

Nevýhodou je fakt, že vytvořený soubor bude zabírat přesně tolik, kolik je kapacita zálohovaného zařízení nebo oddílu. Vytvořené zrcadlo je možné připojit jako další přípojný bod v systému. Vlastností programu `dd` lze výhodně využít pro zálohování zavaděče systému nebo celých disků a jejich oddílů.

dump

Softwarový balík `dump` [30] umožňuje obdobně jako `dd` vytváření zrcadel celých oddílů. Nepřistupuje k souborovému systému přes funkce systémového jádra, ale používá vlastní knihovny. To má za výhodu efektivní přístup k zálohovaným datům. Oproti `dd` přenáší pouze reálná data na oddílu a tím šetří místo na vytvořeném obrazu. Z aktuálně dostupných knihoven jsou podporovány pouze souborové systémy ext[2], ext2 a ext3. Pro ostatní souborové systémy mohou existovat podobné nástroje (např. xfsdump pro XFS).

rdiff-backup

Software s názvem `rdiff-backup` [29] je jednoduchý program napsaný v jazyku Python. Umožňuje tvorbu inkrementálních záloh a jejich obnovení pro zvolený časový údaj. Mezi nejvýznamnější vlastnosti patří:

- jednoduchost použití,
- vytváření zrcadel na lokální nebo vzdálené úložiště,

[2] ext (extended file system)

- přírůstkové zálohování,
- zachování atributů a oprávnění – pro nekompatibilní souborové systémy tvorba metadat,
- uchování statistik,
- efektivita přenosu – obdobně jako u `rsync` se přenáší se pouze změny,
- efektivní využití místa použitím pevných odkazů.

Nevýhodou plynoucí z vlastnosti inkrementálních záloh je, že pro obnovu starších dat je třeba přepočítat diference všech změněných souborů oproti těm v aktuální záloze. Tento proces může trvat poměrně dlouho v závislosti na hloubce zanoření.

amanda

Produkt `amanda` [25] [26] je typickým představitelem komplexního zálohovacího řešení, postaveném na nativních GNU/Linux programech jako `tar`, `dump` a dalších. Mezi klíčové vlastnosti patří:

- tvorba úplných a inkrementálních záloh,
- možnost jedním serverem zálohovat větší množství vzdálených klientů na pásková nebo disková pole,
- použití standardních nástrojů a formátů,
- silné zabezpečení přenášených dat AES šifrováním,
- unikátní organizátor pro spouštění velkého množství záloh,
- podpora operačních systémů GNU/Linux, MS Windows a BSD.

K Open Source verzi je k dispozici i placená varianta pod označením Amanda Enterprise s přídavnými funkcemi, grafickým rozhraním a produktovou podporou.

bacula

Program `bacula` [31] je sofistikované klient/server řešení pro tvorbu a správu záloh. Umožňuje zapisovat na více zálohovacích jednotek, čímž elegantně řeší problém s nedostatkem místa. Pro snadné ovládání a monitoring nabízí grafickou nadstavbu. Podporuje inkrementální, rozdílovou a úplnou zálohu. Vhodné uplatnění najde při zálohování většího počtu pracovních stanic, serverů a jako klíčový prvek DR (Disaster Recovery).

1.4 Monitoring systémů a sítí

S rozvojem IT stoupá i fixace na funkčnost a dostupnost poskytovaných služeb. Praxe i teorie ukázaly, že dříve či později k nějaké havárii dojde. Ta může být vyvolána úmyslným útokem, chybou v hardware, software nebo lidským selháním. Všechny tyto příčiny mohou způsobit nefunkčnost služby nebo dokonce celého systému. I přes použití špičkového hardware a dobrého zabezpečení se tomuto faktu nelze vyhnout. Řešení se nabízí v podobě pravidelného sledování nejrůznějších ukazatelů, které mohou včas

upozornit na blížící se nebezpečí a nebo objasnit příčiny vzniklého selhání a tím zamezit jejich opakování.

Stejně jako u datových záloh je i u monitoringu [33] třeba správně navrhnout prioritní systém. Tedy určit stěžejní služby, případně za jakých podmínek a na jak dlouho lze akceptovat jejich výpadek.

Samotné sledování by mělo pocházet od nejnižších vrstev jakými jsou hardware, operační systém, přes sledování služeb, až po síťové okolí. Tím je možné vidět problém v širších souvislostech a správně na něj reagovat. Zabrání se také stavům, kdy se služba zvenčí jeví jako funkční a přitom běží na kolabujícím systému, který jí znemožňuje chod. Příkladem může být FTP server na systému s docházejícím volným místem. FTP služba se bude jevit pro monitorovací nástroje v pořádku, ale připojovanému klientovi odmítne vstup, protože nebude mít dostatek místa pro uložení přihlašovacích informací. [1]

1.4.1 Systémové nástroje

V systému je dostupná celá řada programů [3] monitorujících hardware a běžící systém. Systémy založené GNU/Linux nabízí také možnost zobrazení informací o jeho stavu pouhým nahlédnutím do speciálních souborů v adresáři `/proc`. K získání základního přehledu o stavu operačního systému a hardware se s oblibou používají tyto nástroje:

- Program `htop` – je interaktivní monitor procesů. Umožňuje rychle a snadno získat informace o vytížení systému, zasílat procesům signály a jejich třídění dle zvolených parametrů.
- Program `iftop` – obdobně jako `htop` sleduje vytížení systému, pouze s tím rozdílem, že se zaměřuje na síťovou vrstvu. Poskytuje rychlý přehled o komunikaci na zvolených síťových rozhraních.
- Program `iotop` – sleduje využití vstupně-výstupních zdrojů. Vhodné na sledování toho, který proces nejvíce vytěžuje disk apod.
- Program `smartctl` – je analyzátor S. M. A. R. T. údajů pro pevné disky. Umožňuje vypsání všech dostupných hodnot nebo sumarizaci s výsledným sdělením, zda testovaný disk je nebo není zdravý.
- Program `mdstat` – je soubor obsažený v adresáři `proc`, který poskytuje informace o softwarovém RAID poli. Při selhání disku zobrazí, který disk byl z pole vyřazen a nebo jak probíhá rekonstrukce pro nově přidané zařízení.
- Program `du` – je jednoduchý program zjišťující využití diskového prostoru pro jednotlivé soubory i celé adresáře.
- Program `df` – zobrazuje velikost využitého a volného místa na připojených souborových systémech.
- Program `netstat` – tiskne informace o síťovém subsystému. Vhodný pomocník při zjišťování stavu procesů, obsazených portů a směrovacích tabulek.

Použitím výše zmíněných nástrojů lze získat rychlý přehled o stavu systému a hardware, avšak při sledování velkého počtu hostitelských stanic by takové řešení přestávalo být efektivní.

1.4.2 Sběr protokolů

Běžící služby nebo i samotný systém podává průběžně o svém stavu nejrůznější informace. Pro lepší přehlednost a škálovatelnost se tyto informace zpracovávají centrálně. Na GNU/Linux je pro tento účel k dispozici socket `/dev/log`, do kterého může jakákoliv aplikace nebo i systém zapsat zprávu. Správce protokolů tyto zprávy průběžně vyzvedává a dle zadaných pravidel s nimi nakládá. Pro zápis do socketu slouží speciální funkce, které musí aplikace volat. Ve skriptech je možné využít programů určených přímo pro zápis do zařízení `/dev/log` a tím integrovat i jejich zprávy do systémového protokolu.

Protokolovací systém má tedy ohromný význam. Díky němu je možné dohledat příčinu aktuálních nebo historicky starších problémů a tím v budoucnu podobným stavům předcházet. [2] [33]

logger

Program `logger` [34] slouží ke snadnému posílání zpráv do systémového logu, čehož se hojně využívá ve skriptech. Zprávy lze i zasílat na standardní chybový výstup, nastavovat jim prioritu a označovat je speciálním popiskem.

syslog-ng

Démon `syslog-ng` [35] je moderní nástroj určený pro vyzvedávání a zpracovávání zpráv ze systémového logu. Ty dle pravidel upravuje a třídí do nastavených souborů. V závislosti na důležitosti informace může být zpráva obsažena i na více místech (souborech). Zdroje zpráv mohou být jak lokálního, tak i vzdáleného charakteru. `Syslog-ng` je multiplatformní a nabízen ve verzi open source nebo komerční. Placená varianta přidává navíc podporu ukládání logů do databáze a zvýšenou bezpečnost přenosu v podobě SSL/TLS. Pro distribuce Ret-Hat a CentOS je použit jako výchozí systémový sběrač protokolů.

rsyslog

Program `rsyslog` [36] plní obdobnou funkci jako `syslog-ng`, tedy podporu pro zaznamenávání a strukturalizaci zpráv zaslaných na lokální socket nebo vzdáleně přes síť. Je vyvíjen jako open source a oproti `syslog-ng` poskytuje navíc tyto funkce:

- šifrování zpráv zasílaných přes síť,
- podpora ukládání logů do databází MySQL a PostgreSQL,
- FailoverSyslogServer představuje odolnost proti výpadku centrálního `rsyslog` serveru.

Pro pohodlnější správu je k rsyslogu nabízen placený loganalyzer, který přes webové rozhraní zpřístupňuje analýzu logů. Díky široké paletě pokročilých vlastností je rsyslog hojně nasazován. O tom svědčí i fakt, že byl zvolen jako výchozí správce protokolů pro distribuce Debian a Ubuntu.

logcheck

Do systémového logu proudí spousta informací, které je vhodné průběžně kontrolovat. V případě osobního počítače nebo notebooku se tato činnost dá integrovat nejrůznějšími grafickými doplňky, jakým je třeba program conky. Uživatel vidí v reálném čase změny probíhající v systému a může na ně reagovat. U vzdálených serverů je situace komplikovanější a s rostoucím počtem sledovaných stanic přestává být únosná. Program logcheck poskytuje řešení v podobě hromadné analýzy systémových protokolů. Dle nastavených pravidel a filtrů reaguje posláním mailu s reportem zjištěných nedostatků. [33] [2]

1.4.3 Dohledové systémy

Při sledování rozsáhlejší sítě nebo většího počtu hostů narůstá i potřeba všechny činnosti centralizovat, ideálně na jedno místo, ze kterého bude možné získat rychlý přehled nad stavem všech systémů. K tomuto účelu slouží dohledové systémy. [38]

mon

Software mon [37] je nástroj open source pro monitoring dostupnosti služeb. Pro přehled o stavu sledovaných systémů nabízí jak konzolový výpis, tak i konfigurovatelné webové rozhraní. Služby jsou definovány a testovány programem monitor, který analyzuje jejich stav. V základním nastavení je dostupná celá paleta monitorů pro standardní síťové služby, jakými jsou například WWW a FTP server, nebo odpověď na ping paket. Selhání některého z testů má za následek vyvolání předem nadefinované události, která má za cíl informovat správce systému o vzniklé komplikaci. Mon umožňuje kromě předefinovaných monitorů vytvářet i vlastní, a tím ho přizpůsobit specifickým požadavkům.

zabbix

Pokročilý dohledový systém zabbix [38] je dostupný pod licencí GPL verze dvě. V komerčním nasazení je možné přikoupit jednu z pěti variant uživatelské podpory. Hlavní část zabbixu tvoří zabbix-server a k němu volitelná skupina agentů. Server se stará o provádění předem nadefinovaných činností, jakými jsou například:

- aktivní kontrola vzdálených služeb,
- spouštění akcí,
- zpracování statistik,
- tvorba grafů,
- zobrazení map,

- a další.

Na hostitelských systémech lze sledovat kromě dostupnosti služeb i detailní údaje o systému a jeho běhu. Aby mohl `zabbix-server` k těmto údajům přistupovat, je nutné mít na sledovaných stanicích nainstalovaný `zabbix-agent`, který poskytne serverové části rozhraní pro přístup k vyexportovaným údajům.

Celý systém je možné spravovat přes pokročilé webové rozhraní nebo ruční editací konfiguračních souborů. Uživatelské nastavení se ukládá do databáze. Mezi podporovanými jsou MySQL a PosgreSQL. Zjištěné selhání sledovaných služeb je možné zasílat formou zpráv na mail, telefon nebo jabber. Open source projekt `zabbix` představuje sofistikovaný dohledový systém hodící se pro sledování většího počtu klientů a snadnou tvorbu statistik. Oproti staršímu programu `nagios`, nabízí `zabbix` pohodlnější konfiguraci přes webové rozhraní a širokou uživatelskou podporu.

Vlastní řešení

Využitím systémových nástrojů a některého ze skriptovacích jazyků, například `bash`, je možné propojit a naohýbat jednoduché programy pro řešení komplikovaného problému. Tato metoda využívá základní filosofie Unixu, tedy že by program měl:

- provádět právě jednu věc a tu dělat dobře,
- vzájemně spolupracovat s ostatními programy.

Monitoring systému i celé sítě se dá v menším měřítku elegantně řešit právě touto formou. S rostoucím počtem sledovaných zařízení a služeb může správa skriptů celý dohled komplikovat. Tehdy bývá vhodné se porozhlédnou po nějakém komplexnějším řešení.

1.4.4 Systémy detekce vniknutí

Systémy pro detekci vniknutí `[39] [41]` jsou jakýmsi poplašným zařízením na síti, které upozorňuje na podezřelé chování připojených účastníků. Existují přitom dva základní typy, síťové NIDS (Network Intrusion Detection System) a hostitelské HIDS (Host Intrusion Detection Systems). NIDS se umisťují na kritické body sítě, aby pokryly všechny její segmenty. Naopak HIDS se nachází přímo v hostitelských systémech, kde plní obdobnou úlohu jako NIDS, tedy monitoring a upozornění na podezřelé činnosti.

Hlavním důvodem proč se IDS nasazují, je předcházení útokům. Včasná informace o tom, že probíhá útok na spravovaný systém, nebo dokonce o tom, že už byl úspěšně proveden, je velice důležitá. Jen tehdy je možné provést nutný zákrok a zabránit napadení systému nebo alespoň minimalizovat škody. V opačném případě může útočník v systému parazitovat velice dlouho, napadat další systémy a v konečném důsledku způsobit škody v takovém rozsahu, že jejich obnova nebude uskutečnitelná. `[1] [2] [3]`

Pronájem IDS

Velká časová a odborná náročnost mnohdy neumožňuje provozovat IDS v rámci vnitřního IT zázemí. Proto jsou s oblibou najímány firmy, které zajistí provoz a správu

detekčních systémů 24/7. Většinou staví na nástrojích open source jako snort a firestorm v kombinaci s dalšími dohledovými systémy. [41]

snort

Snort [2] je open source systém pro prevenci a detekci nedovoleného vniknutí. Řadí se mezi nejpoužívanější a nejznámější NIDS. Díky své oblibě se pyšní širokou uživatelskou základnou a dobrou zdokumentovaností. Činnost snortu spočívá v analýze a vyhledávání vyhovujících signatur, které signalizují nedovolené průzkumy sítě nebo útoky. Ke svému provozu nabízí čtyři režimy:

- sniffer – pouze čte pakety z vybraného rozhraní a vypisuje na standardní výstup,
- packet logger – ukládá pakety na disk,
- NIDS – analyzuje provoz ze síťového rozhraní a dle nastavených pravidel zasílá varování,
- inline – odebírá pakety z iptables a na základě pravidel rozhoduje o jejich povolení nebo zahození.

Pro přehlednější zobrazení a pohodlnější nastavení, lze využít projekt base, který přes webové rozhraní navazuje na možnosti programu snort.

firestorm

Firestorm [41] [2] plní úlohu nenáročného a pohotového NIDS. Pyšní se rychlou analýzou, zasíláním upozornění a možností měnit nastavení senzorů za běhu systému. Efektivní zpracování sbíraných dat a malá zátěž pro systém i síť může být mnohdy klíčová, proto je i firestorm oblíbeným nástrojem pro detekci útoků. Naopak oproti snortu není firestorm tak intenzivně vyvíjen a tím může komplikovat vyřešení problému nebo hledání uživatelské podpory.

1.5 Zabezpečení GNU/Linux

Častý omyl počítačových uživatelů a celé řady organizací je, že zrovna jim hackerský útok nehrozí a odsouvají tak otázku bezpečnosti až na druhou kolej. Neutichající rozvoj informačních technologií, přesun každodenní komunikace a obchodu na Internet zvýšil ale zájem hackerů o cizí zdroje natolik, že se obětí může stát kdokoliv. Důvody k útoku se různí, od prostého využití konektivity, výpočetního výkonu a diskového prostoru, přes získání důvěrných dat, až po zneužití hostitelských zařízení k dalším útokům na vyšší cíle. Otázka tedy nezní, proč by si hacker vybral onu firmu nebo uživatele, ale jakou cenu představují uchovávané informace a dostupnost služeb.

Zabezpečení chodu systému, služeb a dat na něm uchovaných, by nemělo spočívat v jednom bezpečnostním prvku, ale v systematické ochraně všech částí, které mohou tento stav ovlivnit a v neposlední řadě zálohováním. [3]

1.5.1 Fyzický přístup

Ať je systém s důležitými daty připojen do sítě a Internetu nebo nikoli, mělo by jeho umístění znemožnit přístup nepovolaným osobám. Pokud tomu tak není, je bezpečnost systému a dat značně ohrožena, ať už neúmyslným poškozením nebo cíleným útokem. Vhodnou volbou je speciálně vyhrazená místnost jen pro servery, bez oken a s vlastní klimatizací. Pro zvýšení stupně ochrany by měl být přístup do této místnosti monitorován. [3]

1.5.2 Open source

Žádná lidská činnost se neobejde bez chyb. Proto i produkty pocházející z lidských zdrojů obsahují, obsahovaly a obsahovat budou, nedostatky. I když se tento fakt dá značně omezit zautomatizováním procesů a důkladnou kontrolou, nikdy ho nelze zcela odstranit. Software jakožto produkt lidského snažení je poměrně velkým zdrojem chyb. Pokud se však jedná o operační systém nebo kritickou aplikaci, je žádoucí, aby chybovost byla co nejnižší. V případě výskytu nějaké kritické chyby je pak ohrožen každý, kdo vadný software používá. Aby se toto nebezpečí co nejvíce eliminovalo, je software při vývoji testován na výskyt chyb. Po dobu života produktu jsou nově objevené chyby odstraňovány vydáním opravných balíčků.

Hnutí open source, pod kterým se šíří i operační systém GNU/Linux, si klade za cíl, kromě jiného, umožnit distribuci a úpravu zdrojových kódů bez restrikcí a poplatků komukoliv. Což má za následek ohromnou základnu dobrovolných testerů a programátorů z celého světa, kteří mohou tyto kódy zkoumat a vylepšovat jejich bezpečnost. Oprava objevené bezpečnostní slabiny a její distribuce přes otevřené kanály je velmi rychlou záležitostí. Pokud je systém nastaven na automatické aktualizace bezpečnostních chyb, je jeho zranitelnost redukována na minimum. Málo pravděpodobná se také jeví náchylnost k zadním vrátkům a jiným úmyslným chybám. Za vývoj pod licencí typu open source se staví mnoho komerčních firem jako například Ret Hat, Novell, Canonical, které čerpají výhody plynoucí z této filosofie a doplňují ji o chybějící prvek garantované uživatelské podpory.

Naopak proprietární software (software s uzavřeným kódem) je na testování a odstraňování nalezených chyb ve značné nevýhodě. Omezené kapacity lidských zdrojů a komplikované firemní procedury nikdy nemohou nahradit masy dobrovolných uživatelů open source. Dalším nedostatkem uzavřeného software je přímá závislost na jediné instituci, která může objevené chyby ignorovat a nebo úplně ukončit své působení a tím uživatelům produktu způsobit značné problémy s migrací na jiný program. [3]

1.5.3 Minimální oprávnění

Současná práce více uživatelů na jednom sdíleném zdroji, kterým může být i celý operační systém, si vyžaduje jasně daná pravidla jejich interakce. GNU/Linux byl navržen

jako víceuživatelský systém se speciálním účtem `root`, který může v systému vše a neplatí něj žádná omezení. Pokud je třeba přidělit vybraným uživatelům určitá oprávnění, vytvoří se pro ně nejprve skupina, které se nastaví práva a poté se přidají noví členové, kteří budou oprávnění pro skupinu sdílet.

Uživatelská práva a práva skupin by se měla přidělovat tak, aby vždy byla nejnižší možná. Použití účtu `root` a spouštění nových procesů pod ním se doporučuje pouze v nejnutnějších případech. Na vše ostatní je vhodné vytvářet speciální skupiny a uživatele, kteří budou mít jasně omezený rozsah pravomocí. [3]

1.5.4 Změna kořenového adresáře

Ne vždy je možné zajistit bezpečný běh aplikací. Ať nebezpečí představuje nedůvěryhodnost uživatelů nebo samotná aplikace, je nutné tento prvek buď ze systému odstranit nebo ho dostatečně izolovat. K tomuto účelu slouží prostředí `chroot`, které umožní běh programů uzavřených do specifikovaného adresáře. Do prostředí lze přidávat libovolné prostředky systému, čímž se zajistí vše potřebné pro správný běh potencionálně nebezpečné aplikace. Mnoho programů přímo implementuje `chroot` poskytovaných služeb. Pak není nutné nastavovat speciální prostředí pro celou aplikaci, ale pouze ji nastavit tak, aby poskytované služby běžely v režimu omezeného prostředí. [2] [3]

1.5.5 Atributy a implicitní práva

Aby uživatelé systému nebyli nuceni nastavovat po každém vytvoření souboru nebo adresáře přístupová práva, je k dispozici maska. Ta zajistí nastavení oprávnění pro nově vytvářený soubor nebo adresář. Správce systému tuto masku může přednastavit všem uživatelům a tím eliminovat nechtěné intervence mezi uživateli.

U souborů, kde se očekává malá změna jejich obsahu nebo dokonce není žádoucí žádná modifikace, lze nastavit souborové atributy, které omezující požadavky zajistí. Na souborových systémech, které tuto funkci podporují, to lze provést příkazem `chattr` pro GNU/Linux nebo na BSD je ekvivalentem `chflags`. Této metody se využívá zejména jako preventivní ochrany proti exploitům[3] a pokusům o nedovolenou změnu souborů s hesly nebo protokolů. [3]

1.5.6 Capabilities

Dle normy POSIX obsahuje i GNU/Linux interní podporu Capabilities, i když mnohdy různě implementovanou. Capabilities omezují práva spouštěných procesů. Pokud program potřebuje pro svůj běh `root` oprávnění, vzniká nebezpečí, že v případě chyby v programu útočník zneužije přidělená oprávnění k získání plné kontroly nad systémem. Capabilities umožní spustit proces, ale s omezeným `root` oprávněním. Například spuštění služby pro

[3] Označení exploit zastupuje skupinu programů nebo skriptů, které využívají programátorských chyb k původně nezamýšlené činnosti.

naslouchání na portu 90 může pouze `root`. Capabilities umožní připojení služby na nízký port, ale ostatní práva účtu `root` již spuštěný program mít nebude.

Vhodným nastavením kompetencí lze detailně omezit práva uživatelů a procesů. Nevýhoda tkví v nejednotné implementaci, takže přenositelnost nastavení mezi různými Unix, GNU/Linux a BSD systémy se neobejde bez komplikací. [3]

1.5.7 Blokování portů a služeb

Stejně jako je u souborového systému přidělováno minimální oprávnění pro jednotlivé uživatele, aplikuje se tato filozofie na systém jako celek. Uživatele, který se vzdáleně připojuje k systému přes některý z IP protokolů, nezajímá vnitřní struktura systému a umístění programů. Musí vědět pouze IP adresu a číslo portu, na kterém proces naslouchá. Aby bylo možné přistupovat k více procesům najednou, naslouchá každá služba na jiném portu. Prvních 1024 portů je vyhrazeno pro nejběžnější služby a může na nich naslouchat pouze proces spuštěný pod uživatelem `root`. Zbylé porty mohou být dynamicky přidělovány a nebo využity pro soukromé účely.

K zabezpečení tohoto modelu slouží firewall, který propouští pouze spojení na dovolených portech a splňuje další podmínky. Bezpečnou metodou nastavení firewallu je zakázat veškerý provoz a povolit pouze ty služby, které jsou nezbytné. Základním a také velice účinným firewallem na GNU/Linux je `iptables`.

Pro detailnější rozlišení komunikace lze využít aplikační firewall nebo transparentní proxy server fungující na aplikační vrstvě. Veškerá komunikace prochází přes firewall, který za klienty navazuje spojení podle nadefinovaných omezení. Účiným pomocníkem v tomto směru, jsou soubory `/etc/host.allow` a `/etc/host.denny`, které řídí přístup k řadě internetových služeb využívajících knihovnu `libwrap`. [1] [3]

1.5.8 Vzdálená správa

Pro pohodlnou správu vzdálených stanic založených na systému Unix, se s oblibou využíval protokol telnet. Pomocí příkazové řádky mohl připojený uživatel ovládat systém stejně tak, jako by byl připojený lokálně. S rozvojem Internetu a počítačové kriminality, byl telnet kvůli bezpečnostním nedostatkům nahrazen protokolem SSH. [2]

OpenSSH

OpenSSH je jednou z mnoha volných verzí protokolu SSH, slouží pro bezpečné připojení na vzdálený shell. Ochranu přenášených dat zajišťuje kombinace symetrické a asymetrické kryptografie. Mezi hlavní přednosti OpenSSH patří:

- open source vývoj a GPL licence,
- silné šifrování (3DES, Blowfish, AES),
- podpora tunelování nejrůznějších protokolů včetně grafického systému X Window,
- volba komprese dat,

- zabezpečený přenos souborů přes SFTP.

OpenSSH nemusí nahrazovat pouze telnet, ale díky jeho všestrannosti ho lze použít pro zabezpečení libovolné služby, například POP3, FTP a mnoho dalších. [5]

2. Analýza

Cílem této analýzy je získat přehled o potřebách společnosti 24SNAiLS, a. s., a ze stavu sítě, systémů a provozovaných služeb navrhnout možná vylepšení. IP adresy, porty a jména serverů použité v této kapitole jsou změněny tak, aby věrohodně zrcadlily popisovanou situaci a přitom neohrozily bezpečnost prověřované sítě.

2.1 Charakteristika klienta a jeho potřeb

Společnost 24SNAiLS, a. s., se zabývá poskytováním komplexních služeb v oblasti tvorby multimédií po stránce organizační, technické i obsahové. Hlavní náplň představuje produkční servis, výroba mediálních obsahů (video a audio pořady, animace, výroba dabingu, scénářů) a plná technická podpora v oblasti mediální produkce.

S rozšiřováním společnosti a rostoucím objemem multimediálního obsahu, je důraz kladen na manipulaci a uchování velkého množství dat. Prvním zefektivňujícím krokem bylo zavedení inteligentního přístupu a výměny dat pomocí vyvinuté aplikace QuickBox Sender.

Stávající data management jedné z vybraných sítí na QuickBox Sender navazuje a je postupně doplňován o další funkcionality zlepšující bezpečnost uchovávaných dat. Předmětem této kapitoly je analyzovat slabá místa, ke kterým budou navrhnuta možná řešení při dodržení těchto podmínek:

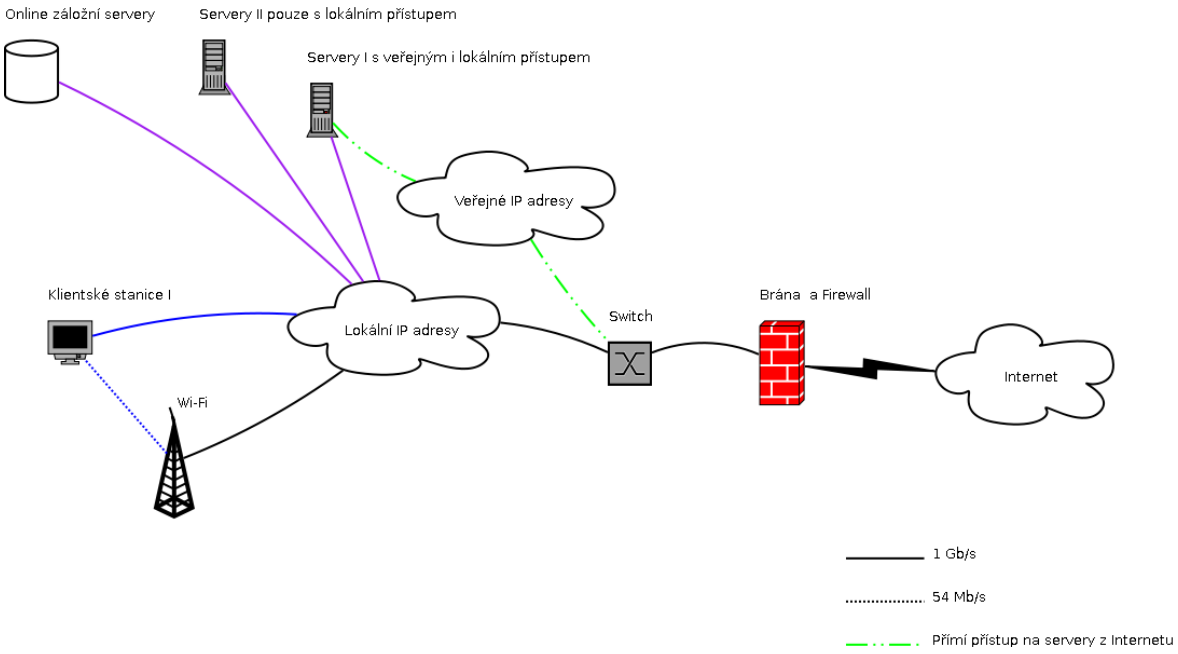
- vysoká bezpečnost a dostupnost datových úložišť,
- kapacita v řádech desítkách TB,
- vycházet z aktuálního data managementu,
- ekonomicky přijatelné řešení.

2.2 Struktura sítě

Síť je tvořena skupinou serverů a klientských stanic, které jsou navzájem propojeny přes gigabitový switch. Přístup do Internetu je řízen přes firewall.

2.2.1 Činnosti na síti

Celý síťový systém je zobrazený na obr. 2. Základní funkce, které je nutné při návrhu zohlednit budou později rozebrány.



Obr. 2: Analyzovaná síť

Přístup k serverům I

Servery ve skupině I jsou využívány přes místní síť, tak i přes Internet. Tato funkce je zajištěna dvojí adresou (lokální a veřejnou) na jednom síťovém rozhraní. Pro rychlý přístup z lokální sítě jsou na serverech nejčastěji využívány tyto služby:

- FTP,
- QuickBox Sender (FTP, WWW, LDAP),
- SSH,
- připojení síťových disků.

Z Internetu se klienti připojují na tyto služby:

- FTP,
- přístup na aplikaci pro výměnu dat QuickBox Sender (FTP, WWW, LDAP),
- webové prezentace,
- VPN.

Důležitou vlastností je dosažení maximální rychlosti mezi lokálními klienty a servery, jelikož výměna velkých objemů dat probíhá poměrně často a ta samá data jsou pak prezentována pro klienty přes Internet.

Přístup k serverům II

Na servery ve skupině II jsou kladeny podobné požadavky, jako na servery ve skupině I, tedy maximální propustnost a rychlost mezi lokálními klienty a servery.

Přístup k zálohám serverů

Pro případ havárie serveru ze skupiny I nebo II jsou k dispozici servery záložní, které obsahují den stará data.

Drátový a bezdrátový přístup klientů

Zaměstnanci potřebují pro svou práci jak data na lokální síti, tak i připojení na Internet. Pro pohodlnější přístup na síť a Internet je k dispozici omezený počet drátových přípojek a nebo neomezené bezdrátové připojení.

Fyzický přístup

Speciálně pro servery jsou vyhrazeny dvě místnosti, do kterých může jen vyvolená skupina zaměstnanců. Ke vstupu nepovolaných osob brání uzamčení místností.

2.2.2 Zhodnocení návrhu sítě

Struktura sítě [1] sice řeší otázku rychlé výměny dat mezi jejími účastníky, ale z hlediska bezpečnosti je to nevyhovující řešení. Takto postavená síť bude po prolomení ochrany jednoho serveru velmi snadno napadnutelná.

Není uplatňováno nastavení firewallu na veškerý provoz mezi lokální sítí a Internetem, ale skupina serverů I tato pravidla obchází. Tím vzniká další nebezpečí v nejednotné distribuci omezení a pravidel pro danou síť.

Nevhodné je umístění serverů a klientů do společné sítě, kde může nastat problém s kolizí IP adresy a tím ohrožení dostupnosti serverů.

Dostupnost Wi-Fi sítě zvyšuje i riziko útoku, proto je vhodné také tuto síť oddělit, aby v případě prolomení nemohl být kompromitován zbytek sítě.

Fyzické oddělení serverů do speciálně vyhrazených místností je pasivním prvkem ochrany, který neumožňuje detekovat neoprávněný přístup. Návrh řešení:

- oddělení serverů přístupných přes Internet do DMZ,
- oddělení Wi-Fi od zbytku sítě a filtrování provozu přes firewall,
- vytvoření samostatné sítě pro servery,
- vytvoření samostatné sítě pro klienty,
- posílení konektivity páteřní sítě a výkonu firewallu,
- monitoring přístupů s kamerovým systémem pro místnosti se servery. [4]

2.3 Porty

Ke zjištění otevřených portů [3] na serverech byl použit skenovací nástroj `nmap` v kombinaci s jeho grafickou nadstavbou `zenmap`. Přehled o všech aktivních klientech na síti 192.168.10.0/24 zajistil příkaz:

```
nmap 192.168.10.0/24
```

Celkový počet skenovaných serverů byl patnáct. Proto je v tabulce 1. uveden pouze sumarizační výsledek otevřených portů na všech serverech a k nim zhodnocení.

Tabulka 1: Otevřené porty

Port	Status	Služba	Zhodnocení
21/tcp	open	ftp	Povolit pouze na vybraných serverech.
23/tcp	open	telnet	Odinstalovat službu, vzdálený přístup zajistit přes `ssh`.
80/tcp	open	http	Povolit pouze na vybraných serverech.
111/tcp	open	rpcbind	Povolit pouze v lokální síti.
139/tcp	open	netbios-ssn	Povolit pouze v lokální síti.
443/tcp	open	https	Povolit pouze na vybraných serverech.
445/tcp	open	microsoft-ds	Povolit pouze v lokální síti.
514/tcp	open	shell	Povolit pouze v lokální síti.
548/tcp	open	afp	Povolit pouze v lokální síti.
901/tcp	open	samba-swat	Odinstalovat službu.
2049/tcp	open	nfs	Odinstalovat službu.
10000/tcp	open	snet-sensor-mgmt	Odinstalovat službu.

2.3.1 Zhodnocení otevřenosti portů

Velké množství otevřených portů bezpečnosti rozhodně nepřidá. Naopak v případě kritické chyby v aplikaci, běžící na otevřeném portu, je ohrožena bezpečnost systému. Proto je shledán aktuální stav z hlediska bezpečnosti za nevyhovující. Řešení představuje v první fázi zakázání nebo odstranění nepotřebných služeb a v druhé aktivaci firewallu v podobě `iptables`, který zakáže vše kromě povolených služeb. Tím se eliminuje riziko naslouchání na portech bez vědomí správce. [4]

2.4 Bezpečnost provozovaných služeb

Provoz jakékoliv služby znamená bezpečnostní riziko, pokud je tato služba dostupná z Internetu, riziko se mnohonásobně zvýší. Proto je nezbytné ověřit zabezpečení veřejně provozovaných služeb. [2]

2.4.1 SSH

Kritické chyby, které by umožňovaly až zneužití účtu `root`, nebyly na žádném z provozovaných serverů objeveny. Avšak řada menších bezpečnostních slabin by ve spojitosti s dobře plánovaným útokem mohla znamenat podobné riziko.

OpenSSH běželo u všech serverů s veřejnou IP adresou na portu 22, a tím bylo každý den vystavováno útokům od robotů, kteří testují otevřenost portu 22 nebo zkoušejí přihlášení na obecně známé účty jako `root`, `admin`, `guest`, `host` a další. Z toho plyne, že pouhý provoz `ssh` na portu 22 zvyšuje nejen riziko prolomení hesla, ale také zatížení serverové konektivity, a to někdy i dost zásadním způsobem.

Další nebezpečí plynulo z vyžadování hesla pro autentizaci uživatele. Pokud by útočník podvrhl SSH server, mohl by při dostatečně dlouhé době odchytit i heslo na `root` účet.

U většiny serverů byl povolen vzdálený přístup pro jakéhokoliv systémového uživatele, tedy i uživatele, který byl do systému přidán za jiným účelem a SSH získal automaticky. Což v kombinaci se slabinami ostatních služeb představuje vysoké riziko prolomení systému.

Návrh řešení pro SSH

Vhodnou ochranou proti testování otevřenosti SSH portu [3] je nastavit server na jiný nepoužívaný port, kde ho hackerský robot nebude čekat.

Pro větší bezpečnost je vhodné vyžadovat pouze autentizaci pomocí klíčů, aby se útočník nemohl ani pokusit testovat heslo. Aby nebylo `ssh` zneužito přes jinou službu, je nutné zakázat automatické přidělení přístupu a povolovat ho pouze pro specifikovanou skupinu uživatelů.

2.4.2 FTP

FTP protokol je stejně nevhodný pro bezpečný přenos souborů, jak protokol telnet pro přístup na vzdálený shell. V základním nastavení většiny serverů se heslo od klienta posílá nešifrovaně, stejně tak jako celá komunikace. Pokud by útočník naslouchal na síti dostatečně dlouho, může odchytit velmi snadno uživatelská jména a hesla. Kdyby zkompromitovaný uživatel měl přístup k důležitým souborům nebo dokonce získal oprávnění `root`, je ohrožen celý systém.

Za použití nástroje `wireshark` a připojení do lokální sítě byl získán přístup k účtu uživatele ihned po jeho přihlášení. Získaný účet umožňoval omezený pohyb po systému.

Návrh řešení pro FTP

V první řadě je důležité zabezpečit FTP provoz. To je možné tunelováním FTP přes SSL/TSL (FTPS). Pokud by z výkonnostního hlediska nebo kvůli nekompatibilitě klientů nebylo možné nasadit FTPS, nabízí se použití protokolu SFTP.

Druhým prvkem je oddělení uživatelských dat od systému, aby v případě zneužití uživatelského účtu nebyli ohroženi ostatní uživatelé nebo systém. K tomuto účelu dobře poslouží `chroot` provozovaného FTP nebo SSH serveru.

2.4.3 Webový server

Pro chod webových aplikací byl nejčastěji využit server `apache`. Mezi provozované služby patřil například QuickBox Sender, phpmyadmin, myadmin. U některých webových serverů byla přidělena plná práva pro složky s webovým obsahem, což otevírá útočníkům možnost spouštět škodlivý kód, který by ohrozil bezpečnost systému.

Pro výše uvedené aplikace nebyl zajištěn bezpečný přenos ani autentizace uživatele, což má za následek zvýšené riziko odposlechu `[2]` komunikace mezi klientem a serverem.

Návrh řešení pro webový server

Stejně jako u FTP je nutné i u webového serveru:

- Šifrovat komunikaci do důležitých částí webových stránek, aby ji nebylo možné odposlechnout. Pro tento účel lze využít knihovny OpenSSL a vytvořit vlastní certifikáty podepsané vlastní CA a nebo pro zvýšení bezpečnosti požádat o vygenerování a podepsání důvěryhodné CA autority.
- Zamezit spouštění nebezpečného kódu nastavením minimálně možných oprávnění pro webové stránky nebo uzamknout celý webový server do speciálního adresáře.

2.4.4 Sdílení dat

Pro síťový přístup k datům na serverech z Windows a Mac OS X byla použita `samba`. Ověřování účtů na servery samba probíhalo centralizovaně přes jediný server LDAP. Pokud by byl tento server mimo provoz, znemožnil by přihlášení na všechny servery, využívající jeho služby. Navíc komunikace mezi LDAP severem a jeho klienty neprobíhala šifrovaně, tedy jako u všech podobně nezabezpečených služeb hrozí odposlechnutí komunikace.

U serverů dostupných přes Internet, nebyla Samba nijak omezena, tedy kdokoliv mohl zkoušet prolomit některý z uživatelských účtů.

Některé záložní servery měly povolen zápis do sdílených složek, což by mohlo v některých situacích zapříčinit nevratnou ztrátu dat.

Návrh řešení pro sdílení dat

Doporučení plynoucí z nalezených nedostatků při sdílení dat zní:

- Nastavit `sambu` tak, aby ověřování uživatelů proti serveru LDAP probíhalo šifrovaně.
- Pro snížení rizika nedostupnosti souborových serverů vytvořit sekundární LDAP, který by výpadek primárního nahradil.
- U veřejně dostupných serverů omezit sdílení dat jen na lokální IP adresy a u záložních serverů sdílet pouze pro čtení. [4]

2.4.5 Wi-Fi

Bezdrátové připojení s sebou přináší, kromě většího pohodlí, i riziko nekontrolovatelné dostupnosti. Oproti drátovému řešení nelze fyzicky omezit, který klient se bude mít možnost připojit a který ne. Síť se tak stává dostupná pro každého v blízkém okolí. Testovaný přístupový bod byl zabezpečen 128bitovým heslem WEP, které má zajistit vstup pouze vybrané skupině uživatelů.

Za pomoci balíku aircrack-ng `[2]` a přepnutím karty do promiskuitního režimu byla po dobu dvou hodin sbírána data z veřejně dostupných prostor. Dalších 20 minut stačilo k tomu, aby program aircrack-ng z nastřádaných údajů útokem `KoreK` využil slabiny protokolu WEP a odhalil desetimístné heslo.

Po prolomením ochrany bezdrátové sítě by útočník získal stejný přístup ke zdrojům, jako kdyby byl připojen přímo kabelem.

Návrh řešení pro Wi-Fi

Pro Wi-Fi větší riziko ohrožení představuje použitá metoda zabezpečení. Tento problém řeší například použití vyššího stupně ochrany WPA nebo WPA2 a dlouhého, netriviálního hesla o minimální délce 10 znaků. Pro počet znaků menších než 8 hrozí i u WPA2 prolomení, za použití `rainbow tables` a slovníkových útoků.

2.4.6 VPN

Pro vzdálený přístup zaměstnanců k lokální síti byla nasazena OpenVPN v režimu klient/server. Autentizace klientů probíhala pomocí vygenerovaných certifikátů, které byly na klientské straně chráněny heslem. Bezpečnostní problém by mohl nastat, pokud by bylo vyžadováno předčasné odvolání některého z klientských certifikátů. Server byl nastaven tak, aby přijal jakéhokoliv klienta, který se prokáže platným certifikátem (neuplynula doba jeho platnosti). Klientovi s platným, ale odvolaným certifikátem by byl ale přístup umožněn.

Návrh řešení pro VPN

U VPN nebyly objeveny závažnější nedostatky, proto lze spíše upozornit na možný vznik problémů při zneplatnění klientských certifikátů. `[6]`

2.5 Dostupnost klíčových služeb

Dostupnost služeb na sítí byla zjišťována základními systémovými nástroji. Pro kontrolu dostupnosti celého systému posloužil příkaz `ping`. Podrobnější výpis o běžících službách zajistila buď kombinace `ssh` a `netstat` nebo sken portů programem `nmap`.

Při rozsáhlejším výpadku by bylo časově náročné získat přehled o celé síti. Se stále rostoucím počtem monitorovaných systémů klesá i přehlednost a efektivita jednoduchých skriptů na neúnosnou úroveň. S výhledem na budoucí rozšiřování by bylo vhodné nasadit některý z monitorovacích systémů, který by rychle poskytl přehled nad stavem systémů a služeb na nich dostupných.

2.6 Bezpečnost systémů a uchování dat

Bezpečnost uchovávaných dat byla zajištěna každodenní zálohou z produkčních serverů na servery záložní. O plánované zálohování se staral `cron`, který pravidelně spouštěl skript, v němž příkaz `cp` kopíroval zvolené adresáře do připojeného síťového disku vzdáleného serveru.

Každý server, ať už záložní nebo produkční, obsahoval jeden disk pro systém a další disky, které tvořily softwarové pole pro data v RAID 5.

Celá síť byla považována za bezpečné místo, které neoplývalo žádným systémem pro odhalení hackerských praktik nebo centrálním úložištěm, kde by bylo možné dohledat, co se stalo před zhroucením serveru, který nelze nastartovat. Nalezené nedostatky tohoto řešení byly:

- Data se přenášela po lokální síti nešifrovaně. Vzhledem k charakteru dat a velkému objemu to ale nepředstavuje až tak závažné bezpečnostní riziko.
- Méně dokonalé protokolování průběhu zálohy a neefektivní přenos dat.
- Absence offline záloh.
- Selhání systémového disku by vyřadilo celý server z provozu bez možnosti rychlého obnovení.

2.6.1 Návrh řešení pro vyšší bezpečnost dat

Řešení se nabízí v podobě sofistikovanějšího systému záloh `[1]` , který bude nejen bezpečně přenášet data, ale i poskytovat přehledné reporty. Vzhledem k objemu zálohovaných dat, který představuje zhruba 30 TB je výhodné použití nástrojů `rsync` a `rdiff-backup`. Vychází se ze stávajícího systému, kdy pro každý server existuje fyzicky jeho záložní.

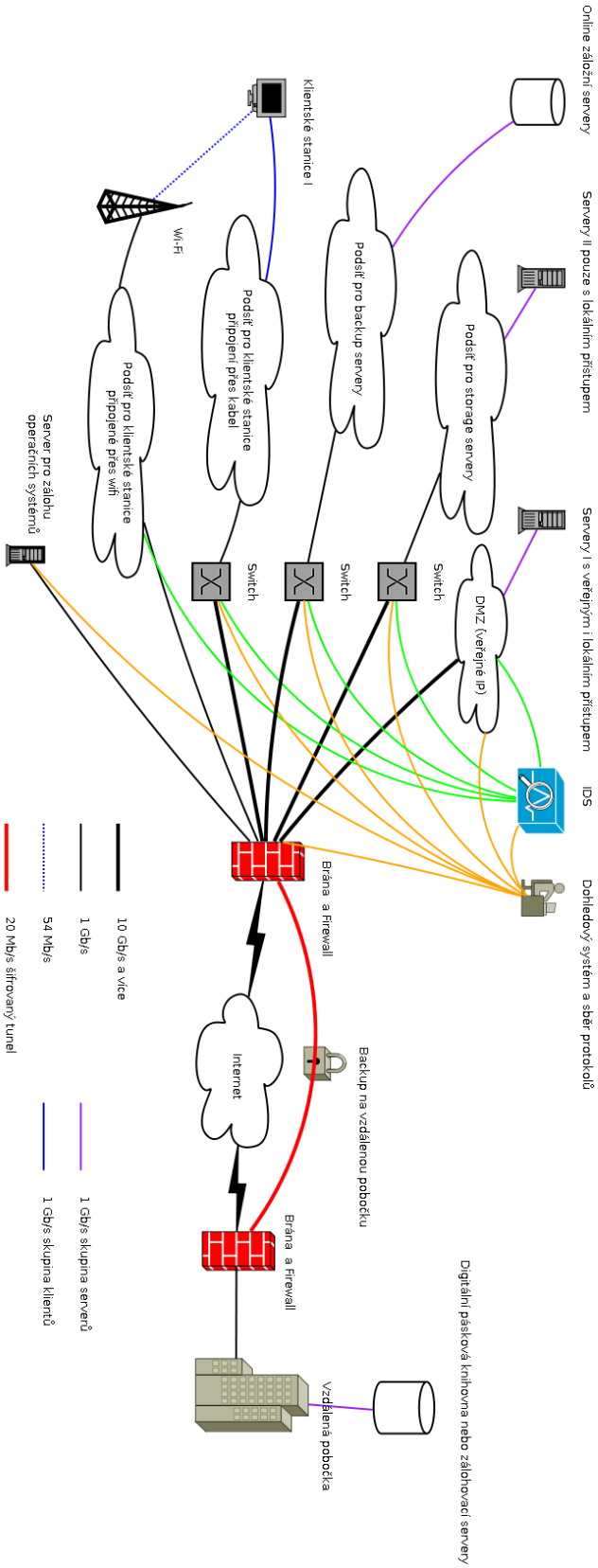
Centralizované řešení v podobě `baculy`, které by zálohovalo všechny servery na jeden, by bylo též vodné, jen by vyžadovalo zakoupení zhruba 40TB serveru. Alternativní varianta, seskupení více serverů do jednoho úložiště, byla pro zvýšenou pravděpodobnost selhání zamítnuta.

V úvahu přichází i nasazení centrálních souborových systémů [19] jako iSCSI nebo AoE, které by zvýšily dostupnost a odolnost systému proti výpadku. Navíc by bylo možné celé úložiště snadno rozšiřovat. Avšak pro smysluplný provoz se nedá vycházet ze stávajícího hardware, ale bylo by nutné zainvestovat do vhodnějších komponent. Tato varianta pro svá specifika zůstává spíše doporučením, kdyby společnost zvažovala výraznější rozšíření datového úložiště.

Pro větší bezpečnost dat by bylo vhodné k online zálohám přidat, nejlépe na jinou pobočku společnosti, jejich offline zálohy. Přenos rozdílových informací by se prováděl jednou týdně přes víkend, aby nevytěžoval linku. Fyzicky se může jednat buď o páskové knihovny, které jsou pro tento druh zálohy ideální, a nebo alternativu v podobě dalšího serveru. Softwarová realizace by opět vycházela z předchozího návrhu pro lokální síť, tedy `rsync`, `rdiff-backup` nebo `bacula` a jemu podobná řešení.

Ochranu proti selhání systémového disku by zajistil na hardwarové úrovni RAID 1, který ale pro omezenou kapacitu diskových pozic na serverech není možný. Proto je doporučeno řešit situaci na softwarové úrovni vytvořením úplné kopie systémového disku programem `dd` nebo `dump`, který by po výměně vadného disku obnovil systém do pár minut. Pokud by byl nasazen jeden ze zálohovacích systémů typu `bacula` nebo `amanda`, lze jej využít i pro tyto účely.

Data jsou tak bezpečná, jak je bezpečný systém, který s nimi pracuje a samozřejmě i prostředí, kde se nachází. Pro preventivní ochranu proti náhodným nebo cíleným útokům by bylo vhodné zavést systém na detekci útoků, který by upozornil ještě před vzniklou hrozbou. S tím souvisí i sběr protokolů ze všech aktivních systémů na centrální úložiště. Tak bude možné dohledat, co se kdy stalo i na systému mimo provoz. Výsledný návrh je zobrazen na obr 3. [2] [3]



Obr. 3: Návrh restrukturalizace sítě

3. Implementace

Předchozí analýza odhalila slabá místa, která by mohla negativně ovlivnit dostupnost služeb a bezpečnost uchovávaných dat. Implementační část se naopak zabývá odstraněním těchto slabin. Jednotlivé kroky realizace jsou vždy konzultovány se společností 24SNAiLS, a. s., aby bylo nalezeno vhodné řešení. IP adresy, porty, jména serverů, telefoní čísla a mailové adresy použité v této kapitole jsou změněny tak, aby věrohodně reflektovaly prováděné úkony a přitom neohrozily bezpečnost prověřované sítě.

3.1 Hromadná distribuce příkazů

Pro situace, kdy bylo nutné aplikovat jednu a tutéž proceduru na více serverů, byl vytvořen skript, který dané akce prováděl hromadně na všech zadaných IP adresách. Nasazení systému pro centrální správu bylo pro velkou rozmanitost úkonů a komplikované nasazení zavrženo. Skript `hromadne_prikazy.sh` umožňuje načíst libovolný počet souborů s IP adresami a jeden soubor s příkazy, oddělený přepínačem `-p`. Načtené hodnoty se provedou na všech cílových stanicích a výstup se uloží do souboru `report.log`. K oddělení jednotlivých IP adres a příkazů se očekává pouhé odřádkování.

```
#!/bin/bash
# Jako parametr je soubor s IP adresami serverů
INDEX=0
if [ $# -lt 1 ]
then
   echo Nebyly zadány potřebné argumenty.
   echo "-p (cesta k souboru)"
   echo Příklad: skript.sh ip_adresy.txt ip_adresyII.txt
   -p prikazy.txt
   else
   p_flag=false;#priznak cesty k prikazum
   for param
   do
        #Nastaveni priznaku pro soubor s prikazy
        if [ "$param" = "-p" ]
        then
        p_flag="true";
        else
             #soubor s prikazy
             if [ $p_flag = true ]
             then
             s_prikazy=$param;
             p_flag=false;
```

```
                else
                     while read SERVER;
                     do
                     IP_ADRESY[$INDEX]="$SERVER"
                     INDEX=$(($INDEX+1))
                     done < "$param"
                fi;
          fi;
     done
     #Soubor s příkazy do pole.
     INDEX=$((0));
     while read PR;
          do
          PRIKAZY[$INDEX]="$PR"
                    INDEX=$(($INDEX+1))
     done < $s_prikazy
     echo " " >report.log;# Vynulovaní souboru s návratovými
     hodnotami.
     #Vykonání příkazů na všech IP adresách načtených
     ze souborů.
     for j in "${IP_ADRESY[@]}"; do
          for i in "${PRIKAZY[@]}"; do
               ssh root@$j $i >>report.log;
          done
     done
fi;
```

3.2 Aktualizace systému

Neaktualizovaný systém zvyšuje riziko úspěšnosti hackerského útoku. Pravidelným aplikováním bezpečnostních záplat, lze nově objevené chyby úspěšně eliminovat. Operační systémy provozované na serverech, jsou až na pár výjimek distribuce Debian a jeden systém s Ubuntu. Společným prvkem pro tyto dva operační systémy je balíčkovací systém `apt`. Z toho vyplývá, že nastavení a aktualizace půjdou snadno rozdistribuovat na ostatní servery.

Aktuálnost systémů byla různá a mnohdy i způsobovala kritické nedostatky v zabezpečení. Proto byl proveden hromadný upgrade na poslední stabilní verzi Debian 6.0:

```
apt-get update && apt-get upgrade;
echo "
deb http://ftp.cz.debian.org/debian/ squeeze main
```

```
deb-src http://ftp.cz.debian.org/debian/ squeeze main
deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
" > /etc/apt/sources.list
apt-get update && apt-get upgrade
apt-get dist-upgrade
reboot
```

Při povýšení byla zvolena pravidelná automatická kontrola a instalace důležitých bezpečnostních aktualizací. O tuto funkci se stará od verze Debian 5.0. program `unattended-upgrades`. Dodatečné volby jako mail, na který se budou zasílat informace o stavu aktualizací, seznamy ignorovaných balíků a další, se mohou nastavit v souboru `/etc/apt/apt.conf.d/50unattended-upgrades`.

Pravidelné spouštění nastavených akcí lze měnit v souboru `/etc/apt/apt.conf.d/10periodic`, kde by neměly chybět řádky:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";
```

Obdobnou funkci na některých systémech plnil i `cron-apt`, který byl ale pro systémovější řešení odstraněn a nahrazen jednotně programem `unattended-upgrades`.

3.3 Zabezpečení služeb

V systému a síti byla detekována celá řada bezpečnostních děr, které by mohly při jejich zneužití způsobit nemalé problémy. Nejprve bylo nutné zabezpečit chod nejdůležitějších a nejvíce využívaných služeb a poté přikročit k drobnějším hrozbám. Avšak proces kontroly a následné opravy by měl být pravidelný, aby i s novými chybami přicházela nová bezpečnostní opatření. [4]

3.3.1 Firewall

U každého veřejně dostupného serveru, byl nasazen firewall [42] v podobě `iptables`. Výchozí politika pro všechny filtry zakázala veškerou komunikaci. Zbylá pravidla pouze povolují nezbytnou komunikaci pro funkci žádaných služeb na serverech. Příklad základního nastavení tabulky `filter` uloženého v souboru `/etc/network/if-pre-up.d/iptables`:

```
*filter
#Výchozí nastavení, vše zakázáno.
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
###### FORWARD #####
```

```
  ###### UOTPUT ######
  # Povolení odchozí komunikace.
  -A OUTPUT -j ACCEPT
  ###### INPUT #######
  #Povolení looback a zahazování komunikace na 127.0.0.0/8.
  -A INPUT -i lo -j ACCEPT
  -A INPUT -i ! lo -d 127.0.0.0/8 -j REJECT
  #Přijímat všechna příchozí spojení. Pro přesunutí navázané
komunikace FTP, SSH a pod, na jiné porty, než jsou povolené.
  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  # Povolení SSH na portu 22.
  -A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT
  # Povolení ping.
  -A INPUT -p icmp -m icmp --icmp-type echo-request -j
ACCEPT
  # Logování neúspěšných pokusů.
  -A INPUT -m limit --limit 5/min -j LOG --log-prefix
"iptables denied: " --log-level 7
  COMMIT
```

Zavedení pravidel do `iptables` při aktivaci síťového rozhraní, zařídí soubor `/etc/network/if-pre-up.d/iptables` s právy pro spuštění:

```
  #!/bin/bash
   /sbin/iptables-restore < /etc/iptables.up.rules
```

Každý server má v základním stavu povolenou odchozí komunikaci, `ssh`, rozhraní `loopback` a `ping`. Povolení dalších služeb se odvíjí od účelu jednotlivých serverů.

3.3.2 FTP

FTP servery jsou využívány k nejrůznějším účelům:

- jedna z cest k nahrání souborů do aplikace QuickBox Sender,
- úložiště pro externí klienty,
- přístup k souborům webových prezentací.

Vzhledem k charakteru dat a použitých uživatelských účtů lze situaci rozdělit na dvě skupiny. Do první skupiny spadá QuickBox Sender a FTP úložiště pro vzdálené klienty. Data na těchto FTP serverech jsou důležitá pouze po krátký časový úsek a velmi frekventovaně se za průtoku velkých objemů mění. Přesun dat musí být jednoduchý a realizovatelný pomocí základních nástrojů systémů Windows, MAC OS X a GNU/Linux.

Pro zabezpečení těchto serverů je nutné nejprve oddělit data FTP uživatelů od zbylého systému. Pokud by byl některý z účtů zneužit, může útočník spáchat minimální škody, protože obsah slouží k rychlé výměně. Důležitým předpokladem je, aby žádný z uživatelů FTP neměl přes stejný účet přístup na důvěryhodnou službu, jakou je třeba SSH. To

se vzhledem k napojení na LDAP musí zajistit až konfigurací SSH serveru. Pokud by byla klienty vyžadována vyšší bezpečnost, je možné se připojit na FTP přes TSL. Avšak vzhledem k nutnosti speciálního klientského programu se nepředpokládá masivní využití.

Druhá skupina umožňuje úpravu zdrojových souborů aplikace QuickBox Sender a dalších webových prezentací. Nedovolenou úpravou nebo smazáním těchto souborů by mohlo dojít ke značným ztrátám, způsobených nefunkčností QuickBox Sender a ostatních webový stránek. Proto je FTP přístup do těchto částí zakázán, aby nemohlo dojít k odposlechu hesla. Nahrazen je bezpečným SFTP, tedy protokolu založeném na SSH.

proftpd

Pro uzamčení uživatelů do jejich domovských adresářů bylo nutné přidat `DefaultRoot` v konfiguračním souboru `/etc/proftpd/proftpd.conf`:

```
DefaultRoot                     ~
```

pure-ftpd

U serverů využívajících program `purgeftpd` byla situace s uzamčeným prostředím obdobná. Pokud se server spouští jako `standalone`, stačilo pouze provést níže uvedený příkaz. V opačném případě se konfigurační soubory neberou v potaz a je nutné upravit soubor superserveru `inetd`.

```
echo "yes" > /etc/pure-ftpd/conf/ChrootEveryone
```

Aktivace podpory bezpečného připojení a přenosu přes TSL je vhodnou volbou po přístup z nedůvěryhodných míst. V první části je vygenerován a podepsán certifikát sám sebou. Pokud by FTP/TSL [44] začalo být využíváno ve větší míře, je možné nechat vydat certifikát ověřený důvěryhodnou CA autoritou.

Pro vygenerování certifikátu s dobou platnosti 7 000 dní a velikostí šifrovacího klíče 2 048 bitů posloužil OpenSSL projekt:

```
openssl req -x509 -nodes -days 7000 -newkey rsa:2048
-keyout /etc/ssl/private/pure-ftpd.pem -out
/etc/ssl/private/pure-ftpd.pem
```

Důležité je omezení přístupu k vygenerovanému certifikátu s klíčem:

```
chmod 600 /etc/ssl/private/pure-ftpd.pem
```

Podpora v `pure-ftpd` byla aktivována následujícím příkazem:

```
echo 1 > /etc/pure-ftpd/conf/TLS
```

3.3.3 Webový server

Na funkci webových serverů navazuje množství poskytovaných služeb. U každé z nich je třeba pečlivě zvážit způsob použití a podle toho navrhnout bezpečnostní opatření. Ochrana proti odposlechu komunikace spočívá v šifrování přenosu pomocí protokolu SSL nebo jeho nástupce TSL.

Pro aktivaci SSL/TSL je u webového serveru `apache` nutné nastavit naslouchání na portu 443 k vybraným doménám. Prakticky se jedná o obdobnou konfiguraci, jaká je v souboru `/etc/apache2/sites-available/000-default`, ve kterém se nachází nastavení `VirtualHost` pro webový server. Příklad nastavení v souboru `/etc/apache2/sites-available/ssl`, na který vede odkaz v adresáři `/etc/apache2/sites-enable/ssl`:

```
 <VirtualHost *:443>
SSLEngine On
ServerAdmin root@24snails.com
DocumentRoot /var/www/secure
ServerName www.quickboxsender.com
  <Directory /var/www/secure>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                     Order allow,deny
                allow from all
  </Directory>
  </VirtualHost>
```

Soubor s certifikátem a privátním klíčem `apache` očekává na místě uvedeném v souboru `/etc/apache2/mods-enabled/ssl.conf`:

```
SSLCertificateFile    /etc/apache2/ssl/24snails_free.pem
```

Vlastní CA

Webové aplikace určené pro vnitrofiremní použití není třeba větší měrou zabezpečovat. Ale i zde mohou po síti cestovat důvěrné informace. Pro tyto účely postačí na kritické části aplikace vyžadovat HTTPS přenos s certifikátem podepsaný neověřenou autoritou.

Uživatelé přistupující do zabezpečených částí aplikace budou proškoleni na práci s certifikáty tak, aby neignorovali upozornění prohlížečů a o změně certifikátu ihned informovali správce.

Vytvoření a podepsání certifikátu sama sebou bylo provedeno obdobně jako u FTP serveru:

```
  openssl req -x509 -nodes -days 7000 -newkey rsa:2048
-keyout /etc/Apache/ssl/`hostname`.pem
-out /etc/Apache/`hostname`.pem
```

Důležitá je též změna oprávnění, aby nikdo kromě uživatele root nemohl zjistit soukromý klíč:

```
  chmod 600 /etc/Apache/ssl/`hostname`.pem
```

Pokud by bylo více domén, je možné vytvořit vlastní CA autoritu, která všechny certifikáty pro servery bude podepisovat. Uživatelé by pouze jednou přidali výjimku pro

CA autoritu a jakýkoliv další certifikát jí podepsaný by byl považován za důvěryhodný. Vzhledem k podepsání certifikátu důvěryhodnou CA tento návrh nebude implementován.

Důvěryhodné CA

Tam, kde se SSL/TSL nasazuje veřejně, nejen pro vnitrofiremní komunikaci, je vhodné využít důvěryhodných certifikačních autorit. Kořenové certifikáty těchto autorit jsou již předinstalovány přímo v prohlížečích. Pokud takováto autorita podepíše certifikát pro konkrétní doménu, stává se automaticky i vygenerovaný certifikát důvěryhodný. Vzdálený uživatel si poté může být jist, že se opravdu připojuje k žádanému serveru.

Nevýhodou komerčních CA bývají nemalé poplatky za využívání těchto služeb, které se při správě více domén mohou značně prodražit. Kompromis [44] mezi maximální bezpečností (placené CA) a nedůvěryhodnými certifikáty (podepsané neověřenou CA) tvoří bezplatné certifikační autority. Mezi hlavní představitele paří StartCom a CAcert.

CAcert představuje zcela nekomerční CA autoritu, kterou používá řada organizací, například FSFE. Nemá však zatím velkou podporu v prohlížečích. Takže uživatelé vstupující na web, který má podepsaný certifikát od CAcert, budou upozorněni, že je autorita nedůvěryhodná.

StartCom nabízí základní balík certifikátů Class 1 zdarma a k tomu přidává možnost si zakoupit vyšší stupeň ověření Class 2. Kromě jiného poskytuje i bezplatně službu OpenID. Kořenový certifikát organizace StartCom je předinstalovaný ve většině webových prohlížečů. Takže podepsané certifikáty touto CA budou pro návštěvníka webového portálu důvěryhodné.

Pro zabezpečení komerčně poskytovaných služeb, jakou je QuickBox Sender, byl použit podepsaný certifikát od StartCom. Hlavním důvodem pro vystavení certifikátů u StartCom bylo to, že proti ostatním řešením nabízí bezplatné ověření identity s předinstalovaným kořenovým certifikátem ve webových prohlížečích.

Oprávnění

Pro složku s daty webových stránek je vhodné nastavit oprávnění tak, aby obsah mohl měnit pouze uživatel `www-data`, pod kterým webový server běží. Členové skupiny `www-data` mají právo na procházení adresářů a čtení souborů, ostatní pouze přístup pro čtení.

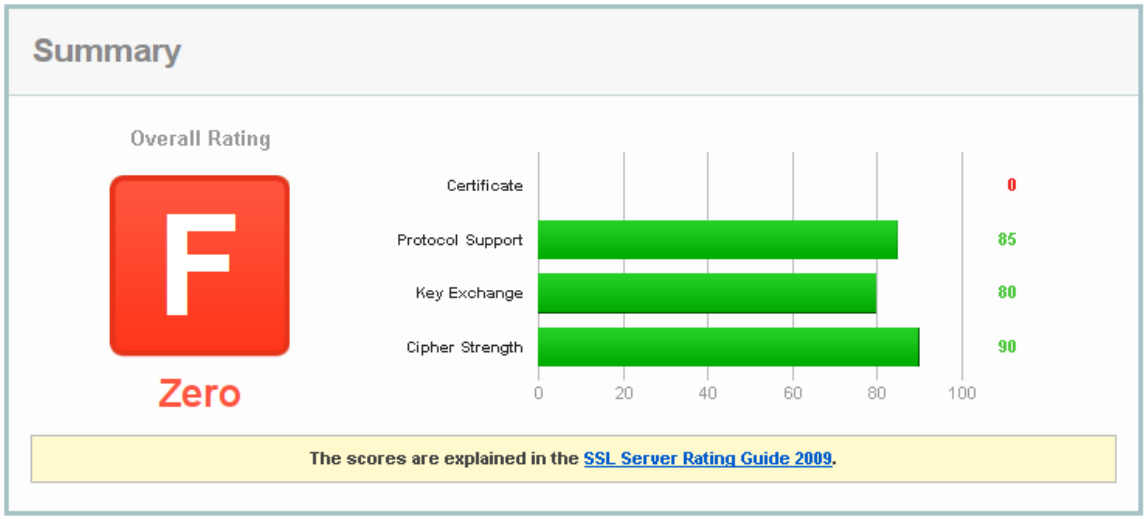
```
chown -R www-data:www-data /var/www/ && chmod -R 755 /var/www/ && chmod 754 /var/www/
```

Podrobné nastavení práv k jednotlivých adresářům se zajišťuje na úrovni `apache` serveru. Vzhledem k rozumné restriktivní politice nebylo třeba dalších úprav.

Zhodnocení provedených změn

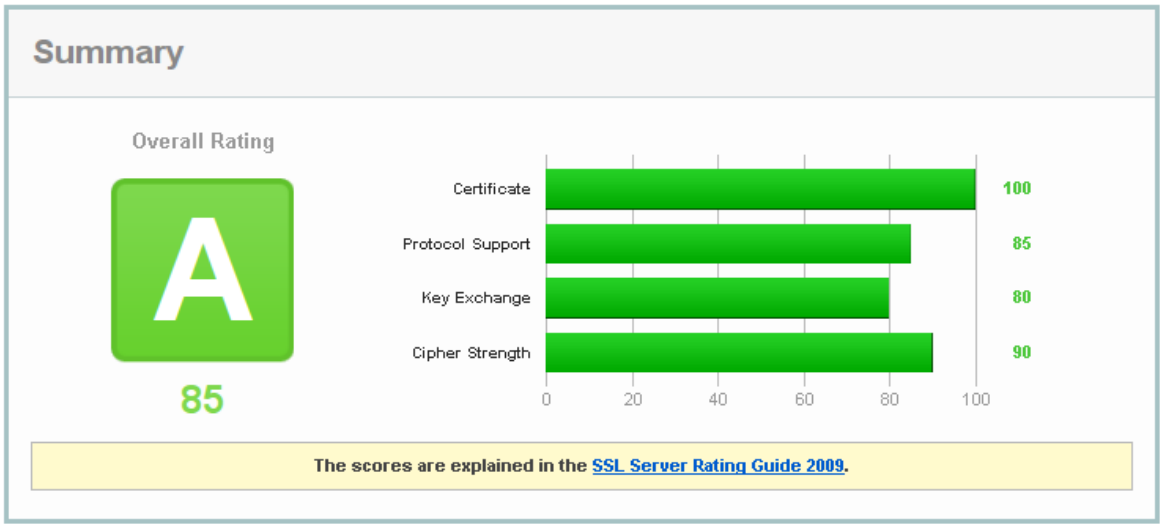
K demonstraci účinnosti výše zmíněných kroků byla otestována webová aplikace QuickBox Sender na bezpečnost konfigurace SSL. Pro test byla zvolena renomovaná organizace `Qualys`, která poskytuje online testovací aplikaci `SSL Server Test`.

První měření na obr. 4 proběhlo před provedením změn s certifikátem podepsaným vlastní CA. Nulové skóre bylo zapřičiněno především absencí důvěryhodného certifikátu, na kterém stojí celý ověřovací proces.



Obr. 4: SSL Server Test - před úpravami

Druhé měření na obr. 5, po provedení výše popsaných postupů, dopadlo už o poznání lépe. Stejné body získala například aplikace `Internetbanking` od České spořitelny.



Obr. 5: SSL Server Test - po úpravách

Dosažené celkové hodnocení 85 bodů ze 100 není nejlepší, ale pro daný typ aplikace naprosto dostačující. Odstranění nedostatků, které plynou z detailů testu, nebylo realizováno. Hlavní důvod představovaly dodatečné náklady a omezení rychlosti, která je pro aplikaci zaměřenou na výměnu dat důležitá.

3.3.4 SSH

Program `ssh` zpřístupňuje příkazovou řádku na vzdáleném serveru, proto by toto privilegium měla mít jen omezená skupina uživatelů. Na všech serverech byl upraven konfigurační soubor /etc/ssh/sshd_config tak, aby `ssh` server neumožňoval připojení nikomu, kromě uživatele `root_ssh`:

```
AllowUsers root_ssh
```

Autentizace probíhala pouze pomocí RSA klíčů a bylo zakázáno alternativní přihlašování:

```
RSAAuthentication yes
PubkeyAuthentication yes
PasswordAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
```

Server naslouchal na jiném než standardním portu:

```
Port 100
```

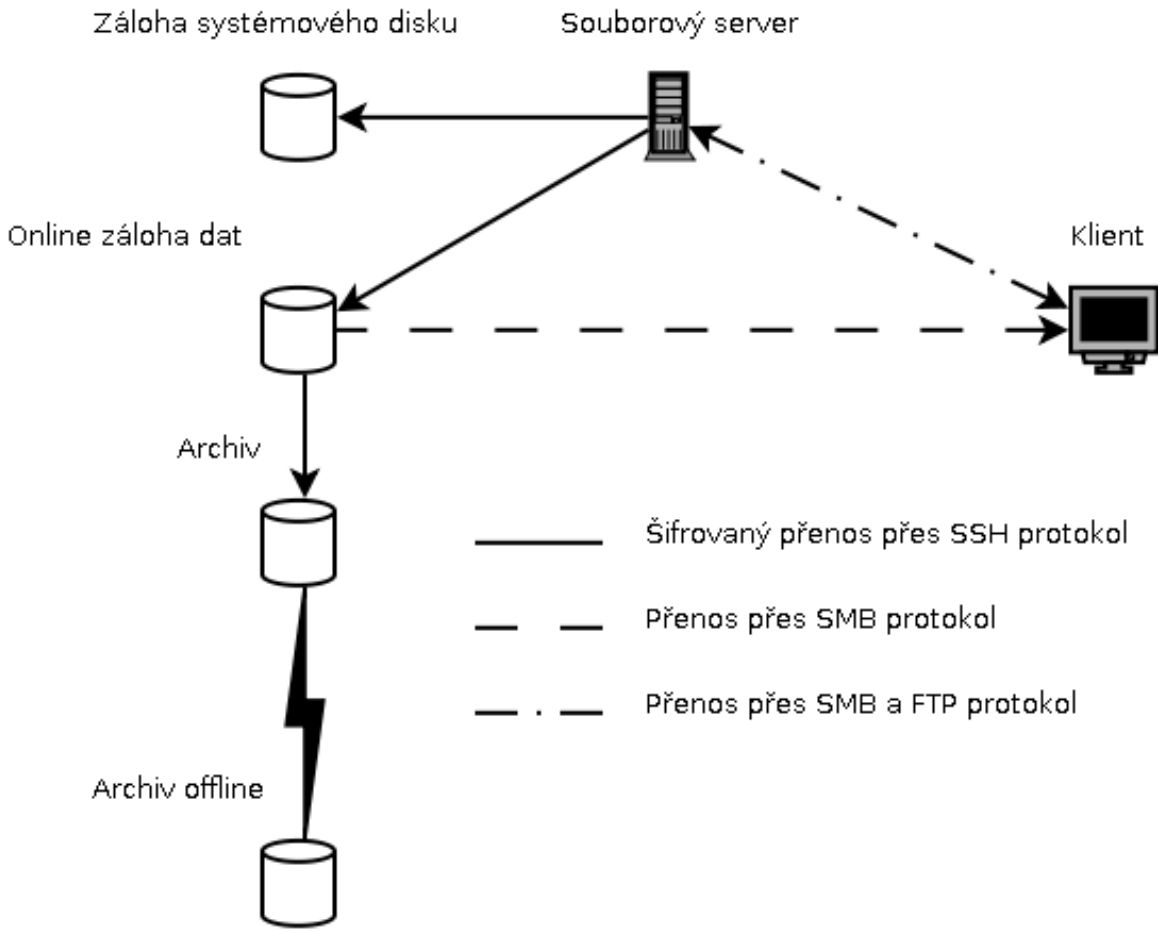
Konfigurace u speciálních serverů může být doplněna o další body, avšak vždy při zachování těchto omezení. Pokud by některý server byl využíván primárně pro výměnu dat přes SFTP, bylo by vhodné zavést uzamykání uživatelských účtů do speciálních adresářů, stejně tak jako u FTP nebo webového serveru.

3.3.5 Wi-Fi

U přístupových bodů bylo nahrazeno šifrování zastaralým WEP bezpečnějším WPA2--PSK. Délka netriviálního hesla převyšuje 10 znaků s dobou platnosti jeden rok. Tedy úspěšnost útoku hrubou silou je velmi nepravděpodobná.

3.4 Záloha dat

Celé řešení spočívá v několika různých zálohách, aby pro každý druh archivace byl zajištěn ten nejlepší způsob. Důraz je kladen na vysoký stupeň bezpečnosti dat a jejich přenosu. Koloběh archivovacího procesu a dostupnost online záloh znázorňuje obr. 4.



Obr. 6: Dostupnost záložních serverů

3.4.1 RAID

Na úrovni systému chrání data proti selhání disku vytvořené RAID 5 pole nebo u výkonnostně zaměřených serverů RAID 10. Rekonfigurací balíku `mdadm`, nebo přidáním řádky do souboru `/etc/mdadm/mdadm.conf`, lze zajistit automatické zasílání upozornění v případě problému s polem:

```
MAILADDR root@24snails.com
```

Rychlý přehled o stavu polí poskytne skript `hromadne_prikazy.sh`:

```
./hromadne_prikazy.sh storage_servers.ip backup_servers.ip
-p commands.cmd
```

Soubor `commands.cmd`:

```
hostname;
cat /proc/mdstat;
echo "-----------";
```

Ukázka souboru `storage_servers.ip`:

```
192.168.10.10
192.168.10.13
192.168.10.14
```

Ukázka výstupu v souboru `report.log`:

```
storage
3907039744 blocks 64K chunks 2 near-copies [8/8]
[UUUUUUUU]
-----------
storage3
5860535808 blocks level 5, 64k chunk, algorithm 2 [4/4]
[UUUU]
-----------
storage4
1953535744 blocks level 5, 4k chunk, algorithm 2 [5/5]
[UUUUU]
```

Obdobným způsobem lze zjistit zdraví disků ze S. M. A. R. T. údajů. Pro vysokou zátěž systému a pro vyžadování plného přístupu k diskovému subsystému, není tento skript prováděn automaticky, ale pouštěn ručně až po zadání fráze k soukromému RSA klíči:

```
./hromadne_prikazy.sh storage_servers.ip -p
commands_smart.cmd
```

Soubor s IP adresami zůstává nezměněn, pouze se použije jiný skript `commands_smart.cmd`:

```
hostname;
for i in /dev/sd?1;do echo $i; smartctl --all  $i; done |
grep -e "/dev/sd*" -e "healt*"
echo "----=====**********************************========----"
```

Ukázka části výstupního souboru `report.log`:

```
storage3
/dev/sda1
SMART overall-health self-assessment test result: PASSED
/dev/sdb1
SMART overall-health self-assessment test result: PASSED
/dev/sdc1
SMART overall-health self-assessment test result: PASSED
/dev/sdd1
```

```
  SMART overall-health self-assessment test result: PASSED
  /dev/sde1
  SMART overall-health self-assessment test result: PASSED
  /dev/sdf1
  SMART overall-health self-assessment test result: PASSED
  /dev/sdg1
  SMART overall-health self-assessment test result: PASSED
  /dev/sdh1
  SMART overall-health self-assessment test result: PASSED
  ----======**********************************===----
  storage4
  SMART overall-health self-assessment test result: PASSED
  /dev/sdb1
  SMART overall-health self-assessment test result: PASSED
```

Pokud by byly shledány ručně spouštěné kontroly za nevyhovující, lze rozdistribuovat příkazy z `commands_smart.cmd` na jednotlivé servery a spoustět je s omezením `command` v souboru `/etc/authorized_keys`.

3.4.2 Záloha dat z MS Windows

Většina programů z Windows se zálohuje na sdílené serverové disky automaticky. U kterých to není možné, probíhá záloha ze strany serveru. Jelikož MS Windows neimplementují SSH server, je samotný proces řešen za použití `samba`, `cron` a `rsync` nástrojů.

Skript `backupall.sh` se za pomocí `cron` démona spouští v pravidelných intervalech a vykoná inkrementální zálohu:

```
  #Připojení stanice s Windows
  set -e; mount -t cifs -o
iocharset=utf8,username=xxxxxx,password=xxxxxxx //192.168.20
.5/data /media/backup/ucetnictvi/original_data/
  #Provedení zálohy dat
  rdiff-backup /media/backup/original_data/
/media/backup/backup_data/ > /var/log/backup_windows/ucd.txt
2>/var/log/backup_windows/ucdErr.txt || true
  #Odpojení
  umount /media/backup/original_data/
```

Detailní informace přenesených souborech nebo chybách jsou dostupné v adresáři `/var/log/backup_windows/`, kam má přistup jen uživatel, pod kterým je spouštěn zálohovací proces.

3.4.3 Záloha dat z GNU/Linux a Mac OS X

Pro zálohování dat ze systémů unixového typu byl využit bezpečný přenos přes SSH protokol ve spolupráci s programy `cron`, `rsync`, `dd` a `rdiff-backup`. Zvolená kombinace nabízí, oproti centralizovaným zálohovacím systémům, obdobné možnosti zálohy bez dalších investic za nový hardware.

Data na serverech pro online zálohu jsou klientům nepřetržitě dostupná v módu pro čtení.

Příprava systému

Aby mohlo zálohování probíhat automaticky bez zásahu administrátora, je nutné na zálohované servery rozdistribuovat veřejné RSA klíče cílových systémů. Zálohování se spouští pod speciálním uživatelem `backuper`, patřící do skupiny `24snails`. Ten přebírá práva své skupiny, která může data číst a upravovat. Při napadení záložního serveru je znemožněno poškození originálních dat, protože uživatel `backuper` může upravovat pouze svá data a data ostatních jen číst. Před každou zálohou je vykonán příkaz přidávající práva skupině pro čtení. To zajistí, aby uživatelé nemohli ohrozit zálohu svých adresářů a souborů:

```
chmod -R g+rX /media/data
```

Informace o stavu proběhlé zálohy budou ukládány do adresáře `/var/log/backup`, kam má přístup pouze uživatel `backuper`. Pravidelné provádění záloh zajišťuje `cron` spouštěním souboru `backupall` a `backupsys`.

Záloha uživatelských dat

Typ zálohy se volí dle charakteru dat. Pro multimediální obsah velkých velikostí je tvořena úplná kopie originálních dat programem `rsync`. Obsah souboru `backupall` pro zálohu serveru `Storage1 s` IP adresou 192.168.10.10.:

```
rsync -ave ssh --delete -r -H --rsh='ssh -p 100'
backuper@192.168.10.10:/media/data/ /media/data/storage1/
> /var/log/backup/storage1.txt
2>/var/log/backup/storage1err.txt
```

Pro účetnictví, webové stránky a jiné záznamy menších velikostí s potřebou dlouhé historie se provádí inkrementální záloha programem `rdiff-backup`. Ukázka souboru `backupall`, pro archivaci stránek:

```
rdiff-backup /var/www/
backuper@192.168.10.18::/media/data/hosting/www >
/var/log/backup/hosting_www.txt
2>/var/log/backup/hosting_wwwerr.txt
```

Záloha systému

Záloha operačního systému probíhá zhruba v měsíčních cyklech programy `dd` a `ssh`. Toto řešení bylo zvoleno především pro vysoký stupeň zabezpečení, bezproblémovost a rychlost obnovy libovolného souborového systému. Systémový disk lze nahradit i diskem s jinou geometrií, avšak kapacita nového musí být stejná nebo větší. Ukázka souboru `backupsys`, pro zálohu operačního systému ze serveru `Storage1` s IP adresou 192.168.10.10. a systémem umístěném na disku `sdb`:

```
  dd if=/dev/sdb | gzip -1 - | ssh -p 100 root@192.168.10.10
dd of=/media/data/Sys/Storage1_image.gz
```

K obnovení je nutné zavést systém z připraveného externího média (flash disk, CD) a zjistit označení nového systémového disku. Níže uvedený příkaz provede nízkoúrovňovou kopii ze zrcadla na cílový disk umístěný na serveru s dočasnou IP 192.168.10.99.:

```
  ssh root@192.168.10.99 dd if=Storage1_image.gz | gunzip -1
- | dd of=/dev/sda
```

Při záloze může dojít k nekonzistenci dat. Uživatelská data a systém jsou na oddělených discích, proto se nanejvýš mohou objevit neúplné informace v nepodstatných místech, která funkčnost systému neohrozí. Z těchto důvodů, a pro možné omezení běžících služeb, nebylo přistoupeno k čisté záloze provedené přepnutím systému na nižší `run level` a připojení disku pouze pro čtení.
Vzhledem k nutnosti `root` oprávnění, byl zálohovací server umístěn do sítě s omezeným přístupem. Samotný proces archivace je spouštěn až po dešifrování soukromého klíče RSA.

Detekce chyby

V případě, že by na některém zálohovacím serveru došlo k chybě při archivaci, objevilo by se hlášení v protokolu. Ty jsou průběžně testovány, zda v nich nepřibyla zpráva. Pokud by měl některý z chybových souborů nenulovou velikost, zašle se ihned upozornění.

```
  if [ `du /var/log/backup/hosting_wwwerr.txt | mawk
'{print($1)}'` -gt 0 ]; then cat
/var/log/backup/hosting_wwwerr.txt | mail
prijem_logu@24snails.com -s "Chyba při záloze na
`hostname`"; fi;
```

Záměnou příkazu pro poslání mailu za návratovou hodnotu 0 nebo 1 by mohl být skript využit i jako zdroj dat pro porgram `zabbix-agent`.

3.5 Dohledový systém

Zvolený monitorovací systém má poskytnout rychlý přehled o stavu sítě a běžících službách na serverech. Výstup by měl být srozumitelný jak administrátorovi, tak i uživateli

se základní znalostí práce na počítači. Z těchto důvodů, a po konzultaci s vedením společnosti 24SNAiLS, a. s., byl vybrán monitorovací systém `mon`, který přes webové rozhraní zobrazí nasbírané informace a upozorní na vzniklé komplikace.

Otestován byl i zástupce robustního dohledového systému `zabbix`, který dokázal shromažďovat nejrůznější data o vnitřním stavu serverů a síťových prvků. Formou grafů a přehledných statistik poskytl přes webové rozhraní grafický výstup.

3.5.1 mon

Stejně jako všechny webové aplikace, kde je vyžadováno heslo, byl i pro `mon` nastaven bezpečný přístup přes HTTPS. Aplikace slouží k vnitrofiremnímu použití, proto stačilo vygenerovat certifikát, který nebyl podepsán důvěryhodnou autoritou.

Přístup k aplikaci byl omezen na úrovni webového serveru vyžádáním platného jména a hesla. Vytvoření přístupových údajů pro uživatele `dohled`, provedl příkaz:

```
htpasswd -c /etc/apache2/passwd dohled
```

A stejně jako u jiných citlivých údajů, nesmí chybět omezení práv:

```
chmod 600 /etc/apache2/passwd
```

V souboru `/etc/apache2/sites-enabled/ssl` byly u adresáře, který definuje přístup k webové aplikaci `mon`, přidány direktivy pro autentizaci uživatele:

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/apache2/passwd
```

Konfigurace sledovaných systémů se nachází v souboru `/etc/mon/mon.cf`. Pro větší počet monitorovaných systémů lze nadefinovat členy skupiny a poté definovat společné vlastnosti ke sledování:

```
hostgroup Backups 192.168.30.11 192.168.30.12
192.168.30.13
```

V menším počtu je ale přehlednější pro každý cílový systém vytvořit novou skupinu a nadefinovat jí pravidla monitorů odděleně. V následující ukázce jsou pro server `Backup` definovány tři monitory sledující:

- Dostupnost systému pomocí odezvy na ICPM ping.
- Funkčnost TCP spojení na port, kde naslouchá `ssh` démon.
- Schopnost vypsání informací o sdílených médiích na serveru.

Monitory jsou nastaveny tak, aby:

- Spuštění monitorů probíhalo každých pět minut.
- Při nedostupnosti služby, alespoň dvakrát v intervalu dvaceti minut, bylo posláno upozornění na mail a telefon.
- Varování byla zaslána každé tři hodiny a to do vyčerpání počtu pěti upozornění. Po znovu-obnovení služby byl zaslán informační mail o této skutečnosti.
- Testování probíhalo nepřetržitě 24/7.

```
  hostgroup Backup
  watch Backup
         service Ping
         interval 5m
         monitor ping.monitor 192.168.30.11
         description Tests if responding on ICMP ping
         period wd {Mon-Sun}
         alert mail.alert informator@24snails.com
+420724999999@sms.cz.o2.com
         upalert mail.alert -S "Server dostupný."
informator@24snails.com
     alertafter 2 20m
     alertevery 3h
     numalerts 5
     ### SSH
     service SSH
          interval 5m
          monitor tcp.monitor -p 22 192.168.30.11
          description Tests if responding on port 22
          period wd {Mon-Sun}
          alert mail.alert informator@24snails.com
+420724999999@sms.cz.o2.com
          upalert mail.alert -S "SSH běží."
informator@24snails.com
     alertafter 2 20m
     alertevery 3h
     numalerts 5
     ### Sdílení dat - Samba
     service Samba
          interval 5m
          monitor smblist.monitor 192.168.30.11
          description It tests if SAMBA works
          period wd {Mon-Sun}
          alert mail.alert informator@24snails.com
+420724999999@sms.cz.o2.com
          upalert mail.alert -S "Samba běží."
informator@24snails.com
           alertafter 2 20m
     alertevery 3h
     numalerts 5
```

Hromadné statistiky o dostupnosti jednotlivých služeb, serverů a době jejich výpadků jsou k dispozici přes webové rozhraní, které je viděl na obr. 7.

Host Group	Service (legend)	Last Checked	Est. Next Check
Backup	DNSName, Ping, SSH, Samba	-7m14s, -7m14s, -7m14s, -7m14s	+2m44s, +2m44s, +2m44s, +2m44s, (test all on Backup)
Backup2	DNSName, Ping, SSH, Samba	-7m15s, -7m15s, -7m15s, -7m16s	+2m43s, +2m43s, +2m43s, +2m43s, (test all on Backup2)
Backup3	Ping, SSH, Samba	-7m15s, -7m15s, -7m16s	+2m43s, +2m43s, +2m43s, (test all on Backup3)
CPCStorage	DNSName, FTP, FTPPublicIP, Ping, SSH	-7m15s, -7m16s, -7m16s, -7m15s, -7m16s	+2m43s, +2m43s, +2m43s, +2m43s, +2m43s, (test all on CPCStorage)
GW	DNSName, Ping, SSH	-7m13s, -7m13s, -7m13s	+2m45s, +2m45s, +2m45s, (test all on GW)
Hosting	Apache, DNSName, Ping, QuickboxFTP, QuickboxWeb, SSH, Web	-7m14s, -7m14s, -7m14s, -7m14s, -7m14s, -7m14s, -7m14s	+2m44s, +2m44s, +2m44s, +2m44s, +2m44s, +2m44s, +2m44s, (test all on Hosting)
HostingExt	WebITExt	-26m58s	+32m55s
PDC41	DNSName, LDAP, Ping, SSH, SambaDomain	-7m16s, -7m15s, -7m16s, -7m16s, -7m16s	+2m43s, +2m43s, +2m43s, +2m43s, +2m42s, (test all on PDC41)
Storage	Apache, DNSName, HardlinkUT, Ping, SSH, Samba	-7m16s, -7m14s, -7m16s, -7m16s, -7m16s, -7m16s	+2m43s, +2m43s, +2m43s, +2m43s, +2m43s, +2m43s, (test all on Storage)
Storage3	DNSName, Ping, SSH, Samba	-7m16s, -7m16s, -7m16s, -7m16s	+2m43s, +2m43s, +2m43s, +2m42s, (test all on Storage3)
Storage4	DNSName, Ping, SSH, Samba	-7m14s, -7m14s, -7m14s, -7m14s	+2m44s, +2m44s, +2m44s, +2m44s, (test all on Storage4)

Service color legend: (top of table)	Unchecked	Good	Failed (no alerts sent)	Failed (alerts sent)	Disabled

Obr. 7: Dohledový systém mon

3.5.2 zabbix

Zabbix poskytuje široké možnosti testování. Od základního ověřování dostupnosti služeb po velice podrobné systémové informace, získané prostřednictvím speciálních agentů nebo vlastních skriptů.

Pro obdobnou funkčnost jakou nabízí `mon` by stačila základní sada položek ze skupiny `Simple check`. Avšak pro demonstraci uplatnění `zabbixu` v rozlehlých sítích byli na serverové stanice nainstalováni i agenti. Agenti byli nastaveni tak, aby se o informace mohl dožadovat pouze legitimní server s danou IP adresou a jménem. Konfigurace souboru `/etc/zabbix//zabbix_agentd.conf`:

```
Server=192.168.20.2
Hostname= monitoring
```

V serverové části byly vytvořeny šablony pro různé typy serverů:

- Template_Linux_all – pro všechny servery,
- Template_Linux_backups – pro záložní servery,
- Template_Linux_storage – pro datové servery.

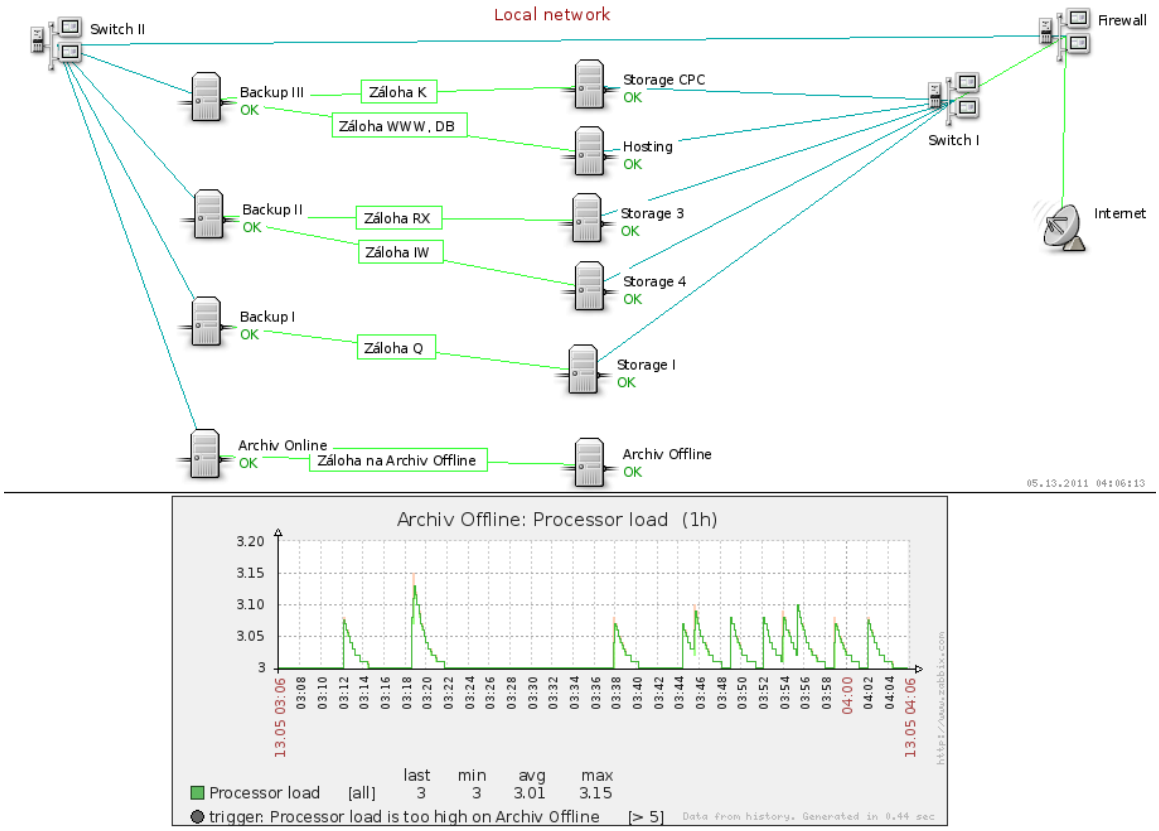
Každá šablona obsahuje položky (například: obsazenost disku, dostupnost služby) a na ně byly navázány tzv. spouště (například: co se stane, když má položka určitou hodnotu). Ukázka položky pro zjištění docházejícího místa (méně než deset procent) v adresáři `/mnt/data`:

```
  vfs.fs.size[/mnt/data,pfree]
```

K využití informace o docházejícím místu bylo třeba nastavit spoušť:

```
  {Template_Linux_all:vfs.fs.size[/mnt/data,pfree].last(0)}<
10
```

Pro přehledné zobrazení aktuální situace byla vytvořena aktivní mapa, která zobrazuje stavy jednotlivých systémů. Navíc bylo mezi datovými servery a jejich online zálohami vytvořeno propojení, které reaguje na hodnotu spouště o docházejícím místu a běhu ssh démonů na serverech. Pokud by jeden z těchto údajů byl chybný, spoj by ihned změnil barvu ze zelené na červenou, což by indikovalo možný problém při záloze. Na spouště se mohou také napojit akce, které definují upozornění na mail, telefon, jabber nebo spuštění vlastního skriptu. Tato možnost nebyla využita, vzhledem k obdobné implementaci v programu `mon`. Ukázka aktivní mapy ze systému zabbix je zobrazena na obr. 8.



Obr. 8: Dohledový systém zabbix

3.6 Systém pro detekci útoků

Na vybrané části sítě byl napojen detekční systém [45] snort, který by upozornil na provádění známých typů útoků. Zvolen byl především pro silnou vývojovou a komunitní větev, která zajišťuje aktualizaci a popis nově vznikajících útoků.

3.6.1 snort

Po instalaci byl snort spuštěn v řežimu NIDS, tedy jako síťový analyzátor. Aby mohl přijímat komunikaci z celé sítě, bylo nutné připojené rozbočovače nastavit tak, aby zrcadlili provoz na specifikovaný port a síťové karty přepnout do promiskuitního režimu:

```
ifconfig eth0 promisc
```

Nastavení pravidel, která snort používá pro generování upozornění, se nachází v souboru /etc/snort/snort.conf. Vzniklá varování na nežádoucí praktiky na síti byla směrována do souboru /var/log/snort. K němu byl nastaven přístup pouze pro uživatele root a skupinu adm.

Domácí síť byla nastavena pro jednotlivé interface, aby obsahovala všechny podsítě, ostatní sítě spadají do externích:

```
var HOME_NET 192.168.0.0/16
var EXTERNAL_NET any
```

Aby snort zbytečně netestoval služby, které na daných serverech neběží, byl nadefinován seznam serverů a služeb na nich běžících:

```
var FTP_SERVERS $HOME_NET
var SQL_SERVERS 192.168.15.10
var HTTP_SERVERS 192.168.15.10
var SMTP_SERVERS 192.168.15.10
var DNS_SERVERS 192.168.15.1
```

Pro nestandardní porty, například ssh naslouchající na portu 100, byl snort také přizpůsoben:

```
SSH_PORT 100
```

V konfiguračním souboru byly připojeny již nadefinované typy útoků. Tento seznam je neustále rozšiřován komunitou, takže i v budoucnu bude systém schopen detekovat dnes neznámé útoky. Pro vyzkoušení byla síť nejprve skenována nástrojem nmap:

```
nmap 192.168.15.0/24
```

Část hlášení o provedeném skenu:

```
[**] [1:1418:11] SNMP request tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
```

Poté byl proveden útok na webový server k získání výpisu souboru /etc/passwd:

```
[**] [1:1122:5] WEB-MISC /etc/passwd [**]
[Classification: Attempted Information Leak] [Priority: 1]
```

Program snort definuje čtyři závažnosti upozornění info (4), low (3), medium (2) a high (1 – 0). Vhodným kompromisem mezi informovaností a množstvím planých poplachů byl pro danou síť vyhodnocen stupeň high. Informační email s výsledky je zasílán skriptem send_NIDS.sh spouštěným pravidelně cron démonem:

```
  if [ /var/log/snort/alert -nt /var/log/snort/old ]; then
    grep /var/log/snort/alert -e "Priority: [0-1]" | mail
prijem_logu@24snails.com -s "Detekce útoku";
  fi; touch /var/log/snort/old
```

Mail se pošle pouze pokud přibylo nové upozornění stupně high. Soubor s varováními lze také napojit na výstražný systém zabbixu. Pokud by byl dohledový server mimo provoz nebo napaden útočníkem, je vhodné mít právě takovouto nezávislou cestu pro informování pověřené osoby.

3.7 Sběr systémových protokolů

Aby bylo možné dohledat informace o systémech, které jsou dočasně mimo provoz nebo zjistit příčinu vzniklého selhání, byl vytvořen centralizovaný sběr dat. Nastřádané protokoly lze automatizovaně procházet a vyhodnocovat dle nich aktuální stav systémů. Tuto funkcionalitu již ale zastávají monitorovací nástroje, proto od další analýzy programy watchlog nebo log-check bylo upuštěno.

Informační smysl této archivace spočívá v uchování kompletního stavu systémů s dlouhou historií. Ta poskytne v případě potřeby všechny důležité údaje pro hloubkovou analýzu.

3.7.1 rsyslog

Pro většinové nasazení operačního systému Debian 6.0., byl pro práci se systémovými protokoly zvolen program rsyslog, který je v distribuci již předinstalován. Oproti konkurenčnímu projektu syslog-ng nabízí rsyslog více možností nastavení.

Pro vzdálený sběr logů bylo vyžadováno, aby server přijímal požadavky pro TCP a UDP spojení na portu 514, což zařídí úprava souboru /etc/rsyslog.conf:

```
  $ModLoad imudp
  $UDPServerRun 514
  $ModLoad imtcp
  $InputTCPServerRun 514
```

Přijaté zprávy jsou tříděny dle jména serveru nebo IP adresy do adresářů jejich odesílatele. To zajišťuje šablona all_remote, která definuje pravidla zařazení a následné aplikování na všechny typy zpráv:

```
  $template all_remote, "/var/log/remote/%HOSTNAME
%/messages.log"
  *.*     ?all_remote
```

U všech odesílatelů systémových protokolů, byl proveden příkaz zajištující odesílání logů na sběrový server přes TCP protokol:

```
echo "*.*      @@192.168.13.10:514" >> /etc/rsyslog.conf
```

Pro znemožnění odposlechu zasílaných informací lze využít program `stunnel`, který vytvoří šifrovaný kanál mezi zdrojovým a cílovým serverem.

3.7.2 ntp

K centrálnímu ukládání logů je nutné mít na všech systémech stejný čas. Proto byl na servery doinstalován `npt` démon starající se o časovou synchronizaci pomocí počítačové sítě:

```
apt-get install ntp
```

K zajištění funkčnosti bylo nutné přidat časový server do konfiguračního souboru `/etc/ntp.conf`:

```
echo "server ntp.cesnet.cz " >> /etc/ntp.conf
```

3.8 Vzdálená správa

Aby nebyly systémy zbytečně zatěžovány a ohrožovány dalšími aplikacemi pro vzdálený přístup, byl využit všude přítomný a zabezpečný `ssh`. Samotná kombinace programu `ssh` a `ssh-agent` už umožní vzdálenou správu libovolného serveru s platným klíčem RSA. Avšak pro plnohodnotné testování funkčnosti služeb (FTP, SAMBA a další) byl třeba i vnější přístup k serverům. Tedy VPN (Virtual Private Network), která umožní jak přímý přístup na servery, tak i využívání ostatních síťových služeb.

Zaměstnanecká OpenVPN pro tyto účely byla zamítnuta. Hlavní důvodem bylo možné odříznutí přístupu ke správě při přetížení nebo pádu OpenVPN serveru. Využívání různého média ke správě a pro ostatní činnosti má také za následek posílení bezpečnosti celého řešení.

3.8.1 VPN přes SSH

Program `ssh` umožňuje kromě jiného vytvářet i šifrované tunely, do kterých ukryje libovolnou komunikaci. Této vlastnosti je využito ve skriptu `vpn.sh`. [43]

Klient

Klientská část nejprve vytvoří šifrovaný tunel s adresou 10.0.0.2 a poté do něj adresuje komunikaci určenou pro vzdálené sítě:

```
sudo -v
ssh root@77.66.55.22 -p 80 -c blowfish -w 0:0 -i
/root/.ssh/vpn_rsa &
```

```
  while ! sudo ifconfig tun0 10.0.0.2 netmask 255.255.255.0
2>/dev/null && echo "Čekám na tun0.";do sleep 2; done
  route add -net 192.168.10.0 netmask 255.255.255.192 dev
tun0;
  route add -net 192.168.20.0 netmask 255.255.255.0 dev
tun0;
  route add -net 192.168.15.0 netmask 255.255.255.0 dev
tun0;
  echo "VPN spojení navázáno."
  echo "Pro odpojení, zavřete okno."
  while true;do sleep 100;done
```

U klienta byl vygenerovaný nový RSA klíč určený pouze pro VPN:

```
  ssh-keygen -b 2048 -f /root/.ssh/vpn_rsa
```

Soukromý klíč je vhodné při vytváření ochránit heslem a zamezit k němu přístup ostatním uživatelům:

```
  chmod  u=rwX,go=--- /root/.ssh/
```

Server

Na straně serveru byla přidána do souboru `authorized_keys` veřejná část VPN klíče. Omezení `command` zajistí, že se po připojení klienta provede pouze nastavení IP adresy zařízení `tun0`:

```
  command="/sbin/ifconfig tun0 10.0.0.1 netmask
255.255.255.0;echo Spojení k serveru navázáno!" ssh-rsa
AAAAB3…
```

Dále bylo nutné povolit přeposílání paketů mezi rozhraními a upravit `nat` tabulku, aby maskovala pakety na adresu rozhraní `eth0`:

```
  iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth0 -j
MASQUERADE
  sysctl -w net.ipv4.ip_forward=1
```

Pro co nejvyšší dostupnost z různě zabezpečených sítí, byl nastaven `ssh` server tak, aby naslouchal na portu 80, kde je očekáván webový server. To umožní průchod přes firewally blokující veškerou komunikaci kromě protokolu HTTP. VPN přes protokol SSH je univerzálním doplňkem vzdálené zprávy, kterou lze provozovat se základním softwarovým vybavením na GNU/Linux, MAC OS X a dalších systémech unixového typu.

Závěr

Cílem práce bylo zvýšit bezpečnost uchování velkého množství dat, čehož bylo docíleno splněním všech bodů zadání. Provedená analýza dala dostatečné množství informací k eliminaci nalezených nedostatků.

Na veřejné servery byl pro větší bezpečnost aplikován firewall s povolením pouze nezbytných portů. Taktéž veřejně nabízené služby byly dle jejich zaměření patřičně zabezpečeny. Citlivé části webových stránek byly zajištěny proti odposlechu důvěryhodným SSL certifikátem. Následné prověření bezpečnosti SSL Server Testem vykázalo nadprůměrné výsledky. Nebezpečné služby v podobě FTP serveru byly buď ze systémů odstraněny, nahrazeny SFTP nebo byl jejich běh oddělen od zbylého systému, aby ho nemohly ohrozit.

Ochrana proti ztrátě dat byla zajištěna plánovanou archivací, která probíhá po šifrovaném kanálu. Aby bylo dosaženo co nejlepšího výsledku, byl pro každou charakteristickou skupinu dat vybrán jeden z typů zálohy: úplná, inkrementální a úplná-nízkoúrovňová. Preventivní ochranu proti hardwarovému selhání zajišťují testy stavu polí RAID a kondice disků. Případný výpadek disku je možné předpovídat a zajistit jeho včasnou výměnu ještě před tím, než by ohrozil chod pole.

Dohled nad sítí a běžícími službami byl zajištěn centrálním monitorovacím nástrojem `mon`. Jako možná alternativa pro pozdější rozšíření byl otestován i robustnější dohledový systém `zabbix` s detailním monitoringem vnitřního stavu serverů.

Na bezpečí celé sítě dohlíží program `snort` ve spolupráci se skriptem na zasílání varovných zpráv administrátorovi.

Pro nalezení informací o běhu systémů a služeb, i bez jejich funkčního stavu, byl aplikován centrální sběr systémových protokolů.

Přístup ke vzdálené správě byl zajištěn, po odstranění slabých míst, programem `ssh`. K pohodlnému a zároveň bezpečnému připojení do celé sítě, byla pro administrátorské účely realizována oddělená VPN založená také na `ssh`.

Zavedená opatření, jsou ke spokojenosti 24SNAiLS, a. s., využívána v každodenním provozu. Zlepšila se reakční doba na vzniklé problémy, zvýšila se odolnost systémů proti fyzickému selhání disků a v neposlední řadě byla vylepšena bezpečnost uchování dat.

Použitá literatura

[1] NORTHCUTT, STEPHEN, et al. *Bezpečnost sítí: Velká kniha*. Vyd. 1. Brno : CP Books, c2005. 589 s. ISBN 80-251-0697-7.

[2] MCCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. *Hacking bez tajemství*. 3. akt. vyd. Brno : Computer Press, c2003. 612 s. ISBN 80-722-6948-8.

[3] HATCH, Brian; LEE, James; KURTZ, George. *Hacking bez tajemství: Linux*. 3. dop. vyd. Brno : Computer Press, c2003. 644 s. ISBN 80-7226-869-4.

[4] KÁLLAY, Fedor; PENIAK, Peter. *Počítačové sítě a jejich aplikace LAN/MAN/WAN*. 2. akt. vyd. Praha : Granada Publishing, c2003. 356 s. ISBN 80-247-0545-1.

[5] KRČMÁŘ, Petr. *Linux: postavte si počítačovou síť*. Praha : Granada Publishing, c2008. 184 s. ISBN 978-80-247-1290-1.

[6] GRAHAM, Steven; SHAH, Steve. Administrace systému LINUX. překlad třetího vydání. Praha: Grada, 2007. 550 s.

[7] RANKIN, Kyle. *Linux Knoppix na maximum*. Vyd. 1. Brno : Computer Press, a. s., 2006. 298 s.

[8] *Linuxmanpages* [online]. 2010 [cit. 2011-04-08]. MDAMD. Dostupné z WWW: <http://www.linuxmanpages.com/man8/mdadm.8.php>.

[9] *LDP* [online]. 2010-03-06 [cit. 2011-04-15]. The Software-RAID HOWTO. Dostupné z WWW: <http://tldp.org/HOWTO/Software-RAID-HOWTO.html>.

[10] *AbcLinuxu* [online]. 28. 1. 2009 [cit. 2011-04-10]. LVM – 1 (úvod, vytvoření oddílu. Dostupné z WWW: <http://www.abclinuxu.cz/clanky/system/lvm-1-uvod-vytvoreni-oddilu>.

[11] *Geoinformatics* [online]. 2010 [cit. 2011-04-16]. RAID. Dostupné z WWW: <http://www.geoinformatics.upol.cz/app/prostredkygis/hardware/HW/RAID.htm>.

[12] ČIHAŘ, Michal. *Cihar* [online]. 2011 [cit. 2011-03-12]. Přechod na softwarový RAID 1 snadno a rychle. Dostupné z WWW:

<http://cihar.com/publications/linuxsoft/prechod-na-softwarovy-raid-1-snadno-a-rychle.html>.

[13] *OpenAFS* [online]. April 2000 [cit. 2011-05-15]. OpenAFS Administration Guide. Dostupné z WWW: <http://www.openafs.org/main.html>.

[14] DRAŽIL, Jiří. *Data Storage technologie* [online]. 10. října 2005 [cit. 2011-02-11]. Storage over IP (iSCSI). Dostupné z WWW: <http://www.storage.cz/index.php?option=com_content&task=view&id=45&Itemid>.

[15] *Google labs* [online]. February 2007 [cit. 2011-01-05]. Failure Trends in a Large Disk Drive Population. Dostupné z WWW: <http://labs.google.com/papers/disk_failures.pdf>.

[16] *SANtools* [online]. 2010 [cit. 2011-02-18]. S.M.A.R.T. Disk Monitor. Dostupné z WWW: <http://www.santools.com/smartmon.html>.

[17] ŠVAMBERG, Michal. *Google dokumenty* [online]. 7. listopadu 2010 [cit. 2011-05-15]. OpenAFS. Dostupné z WWW: <http://docs.google.com/viewer?a=v&q=cache:Jzr_Kl_L8vQJ:lvb.sti.fce.vutbr.cz/public/LinuxAlt_2010/2010_11_07_LA_06_OpenAFS/2010_11_07_LA_06_OpenAFS.pdf+openAFS&hl=cs&pid=bl&srcid=ADGEEShFWwLb-BNSNGkJ-d2m3FwzeXJkd0JS0MjxXL1yjwQLTsheTxczok9Az9n623DcS5qvIWvc9MswCLFQajO00J3474TDrIOPFbfmKgbb5Krbcl-HdaDOQY5bebH1B33PbmgP8L-7&sig=AHIEtbSUMD9UfouFKwMhBeIBBmp1fqz6cQ>.

[18] *Západočeská univerzita : server uživatelské podpory* [online]. 2. 7. 2010 [cit. 2011-02-19]. AFS. Dostupné z WWW: <http://support.zcu.cz/index.php/AFS>.

[19] *Datacentrum WEDOS* [online]. 23.08.2010 [cit. 2010-09-11]. Úložné systémy – NAS vs. SAN. Dostupné z WWW: <http://datacentrum.wedos.com/a/78/ulozne-systemy-nas-vs-san.html>.

[20] VÍTEK, Jan. *Svět Hardware* [online]. 4.9.2007 [cit. 2011-01-15]. Test NAS boxů: 1.

část. Dostupné z WWW: <http://www.svethardware.cz/art_doc-38BF655343D6BDB5C125734B002D5401.html?lotus=1&Highlight=0,AirLive>.

[21] SHARPE, Richard. *Samba* [online]. 8-Oct-2002 [cit. 2011-01-18]. Just what is SMB?. Dostupné z WWW: <http://www.samba.org/cifs/docs/what-is-smb.html>.

[22] *Sourceforge* [online]. 2006-05-02 [cit. 2010-08-09]. Linux NFS-HOWTO. Dostupné z WWW: <http://nfs.sourceforge.net/nfs-howto/>.

[23] MLEJNEK, Miroslav. *SWMAG* [online]. 18. listopadu 2007 [cit. 2011-02-120]. Zálohování dat. Dostupné z WWW: <http://www.swmag.cz/150/zalohovani-dat/>.

[24] *Wikipedie* [online]. 3. 4. 2011 [cit. 2011-04-20]. Záloha (informatika). Dostupné z WWW: <http://cs.wikipedia.org/wiki/Z%C3%A1loha_%28informatika%29>.

[25] *Linuxtopia* [online]. 2011 [cit. 2011-02-12]. Linux Backup Guide. Dostupné z WWW: <http://www.linuxtopia.org/online_books/linux_administrators_security_guide/05_Linux_Backup_Guide.html>.

[26] *Tech-FAQ* [online]. 2011 [cit. 2011-02-22]. How to Backup Unix. Dostupné z WWW: <http://www.tech-faq.com/how-to-backup-unix.html>.

[27] *GNU* [online]. 2001, 2011/03/12 [cit. 2011-03-20]. Tar. Dostupné z WWW: <http://www.gnu.org/software/tar/tar.html#introduction>.

[28] *Die.net* [online]. 2010 [cit. 2011-01-19]. Rsync(1) – Linux man page. Dostupné z WWW: <http://linux.die.net/man/1/rsync>.

[29] *Savannah* [online]. 2007 [cit. 2010-09-11]. Rdiff-backup – Features. Dostupné z WWW: <http://www.nongnu.org/rdiff-backup/features.html>.

[30] *Sourceforge* [online]. 2003 [cit. 2011-04-15]. Is dump really deprecated?. Dostupné z WWW: <http://dump.sourceforge.net/isdumpdeprecated.html>.

[31] *Bacula* [online]. April 2, 2011 [cit. 2011-02-25]. Bacula Bacula Main Reference. Dostupné z WWW: <http://www.bacula.org/en/dev-manual/main/main/Bacula_Main_Reference.html>.

[32] *The Storage Architect* [online]. Friday, 13 April 2007 [cit. 2011-04-23]. AoE/FCoE/iSCSI. Dostupné z WWW: <http://storagearchitect.blogspot.com/2007/04/aoefcoeiscsi.html>.

[33] *BSD Devcenter* [online]. 05/17/2001 [cit. 2011-01-16]. System Logging. Dostupné z WWW: <http://onlamp.com/pub/a/bsd/2001/05/17/Big_Scary_Daemons.html>.

[34] *Linux* [online]. 03-04/2001 [cit. 2011-01-25]. Jak na systémový log?. Dostupné z WWW: <http://www.linux.cz/noviny/2001-04/clanek03.html>.

[35] *BalaBit* [online]. 2001 [cit. 2011-02-10]. The reliable syslog solution. Dostupné z WWW: <http://www.balabit.com/network-security/syslog-ng/opensource-logging-system>.

[36] *Rsyslog* [online]. 2011 [cit. 2011-02-15]. Dostupné z WWW: <http://www.rsyslog.com>.

[37] *Wikipedia* [online]. 17 March 2011 [cit. 2011-03-24]. Mon – Service Monitoring Daemon. Dostupné z WWW: <https://mon.wiki.kernel.org/index.php/Main_Page>.

[38] *Zabbix* [online]. 2009/08/25 [cit. 2011-02-25]. Dostupné z WWW: <http://www.zabbix.com/wiki/start>.

[39] *Snort* [online]. 2010 [cit. 2011-03-12]. Dostupné z WWW: <http://www.snort.org/>.

[40] *Scaramanga* [online]. 2010 [cit. 2011-01-07]. Firestorm NIDS. Dostupné z WWW: <http://www.scaramanga.co.uk/firestorm/>.

[41] APRIAS, Roman. *Fakulta elektrotechniky a informatiky, VŠB-TUO* [online]. 2010 [cit. 2011-03-05]. Systémy detekce průniku v Linuxu. Dostupné z WWW: <http://www.cs.vsb.cz/grygarek/SPS/projekty0405/IDS/ids.html#teor>.

[42] *Debian* [online]. 2009-03-16 [cit. 2011-02-13]. Iptables. Dostupné z WWW: <http://wiki.debian.org/iptables>.

[43] *Bodhizazen* [online]. 2010 [cit. 2011-01-11]. VPN over SSH. Dostupné z WWW: <http://bodhizazen.net/Tutorials/VPN-Over-SSH#Background_Information>.

[44] KUČERA, František. *AbcLinuxu* [online]. 18. 12. 2009 [cit. 2011-01-17]. Bezplatné CA – nebojte se šifrovat s S/MIME. Dostupné z WWW: <http://www.abclinuxu.cz/clanky/bezpecnost/bezplatne-ca-nebojte-se-sifrovat-s-s-mime#serverove-certifikaty>.

[45] *Actinet* [online]. 4/2004 [cit. 2011-02-15]. Jak nasadit systém detekce průniků. Dostupné z WWW: <http://www.actinet.cz/bezpecnost_informacnich_technologii/l19/cl18/st1/j1/Jak_nasadit_system_detekce_pruniku.html>.