Česká společnost uživatelů otevřených systémů EurOpen.CZ
Czech Open System Users' Group
**www.europen.cz**

**XXVII. konference – XXVIIth conference**

# Sborník příspěvků

Conference proceedings

**Hotel Srní a Šumava**
**Srní**
**23.–26. října 2005**

Sborník příspěvků z XXVII. konference EurOpen.CZ, 23.–26. října 2005
© EurOpen.CZ, Univerzitní 8, 306 14 Plzeň
Plzeň 2005. První vydání.
Editor: Vladimír Rudolf, Jiří Felbáb
Sazba a grafická úprava: Ing. Miloš Brejcha – Vydavatelský servis
Vytiskl: IMPROMAT CZ, spol. s r. o., Kopírovací centrum RICOH
Smetanovy sady 6, 301 37 Plzeň

ISBN 80-86583-09-0

**Upozornění:**
Všechna práva vyhrazena. Rozmnožování a šíření této publikace jakýmkoliv způsobem bez výslovného písemného svolení vydavatele je trestné.

Příspěvky neprošly redakční ani jazykovou úpravou.

# Obsah

Jiří Orság
Jak (ne)provozovat DNS ..... 5

Pavel Satrapa
Vývoj DNS standardů a technologií ..... 21

Aleš Padrta
OpenSource nástroje pro správu DNS a DHCP ..... 31

Jaroslav Šnajdr
Detekce spamu ..... 51

Miloš Wimmer
Praktická zkušenost s implementací poštovní brány s antivirovou a antispamovou ochranou založené na svobodném software ..... 53

Zdeněk Šustr
Otevřená okna ..... 61

Petr Cahyna
Wireless security ..... 73

Jaroslav Čížek
WiFi, zkušenosti z projektu pokrytí areálu ZČU a projektu Eduroam 75

Petr Grolmus, Michal Švamberg
Single Sign-On řešení pro webové aplikace ..... 87

Václav Pergl
Řízení SW projektů – příručka pro přežití ..... 101

Antonín Bulín
ICT projekty v 21$^{st.}$ – nic složitějšího ..... 105

Josef Basl
Uplatnění teorie omezení (TOC) v projektech IS/ICT ..... 107

Milan Šárek, Pavel Voral
Neurovědy a IT ..... 115

Otto Dostál, Michal Javorník
Regionální řešení zpracovávání medicínských obrazových informací . . 125

Jan Vejvalka
Algorithms in medicine, their value, limitations and perspectives . . . . 131

## Programový výbor

Jiří Sitera, ZČU v Plzni
Jiří Novotný, MU Brno
Václav Pergl, Kerio Plzeň
Vladimír Rudolf, ZČU v Plzni

# Jak (ne)provozovat DNS

**Jiří Orság**

E-mail: j@o.cz

**Abstrakt**

*Domain Name System DNS, existující více než 20 let se stal součástí kritické infrastruktury internetu. Distribuovaný charakter provozu, různorodost softwarových i hardwarových platforem, rozdílná kvalita administrace vytvářejí náročné podmínky pro stabilní provoz. Příspěvek na základě dat z hostcountu domény .cz od roku 1995 dokumentuje změny v kvalitě provozování domén druhé úrovně pod .cz za posledních 10 let. Zvláštní pozornost je věnována struktuře jmenných služeb pro domény druhé úrovně v současnosti, typům a zastoupení používaných jmenných serverů a jejich funkčnosti. Diskutovány jsou nejčastější problémy v konfiguraci delegací a jejich důsledky. V závěru jsou formulována doporučení pro rozumné provozování DNS.*

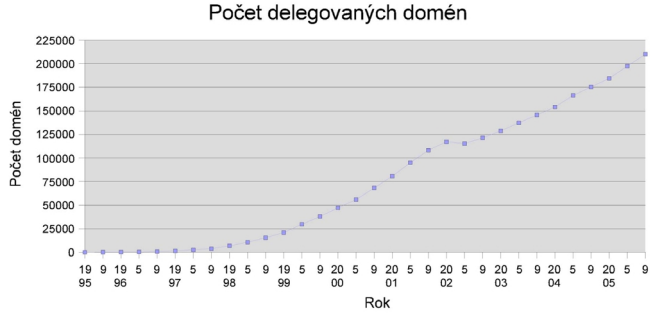
## 1 Úvod

Původním cílem projektu *Hostcount* [1] organizovaného v rámci RIPE bylo analyzovat a dokumentovat růst evropských sítí připojených k Internetu. Z původních 19 zemí v roce 1990 se projekt rozrostl na současných 84 aktivně zpracovávaných top level domén. Pro transfer dat ze zón jmenných serverů v rámci TLD se používá program *host* [2].

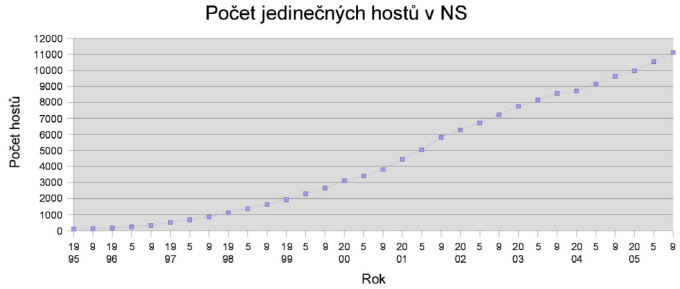
Pro tento příspěvek jsem využil některá vedlejší data vzniklá při zpracování hostcountu pro TLD .cz a výsledky některých dalších analýz. Zajímaly mne trendy ve vývoji jmenných služeb pro domény pod .cz za posledních 10 let, typické konfigurační chyby a změny jejich četnosti ve sledovaném období, zastoupení implementací nameserverů v jmenné službě pro .cz, schopnost nameserverů poskytovat informace. Na základě těchto dat hledám doporučení jak jmenné služby provozovat, čím se řídit při výběru provozovatele nameserveru. Na aktuálních zónových datech se snažím ukázat, že ne každá úspora práce nebo „vylepšení“ vede i k spolehlivějšímu provozu.

# 2 Vývoj jmenných služeb pod .cz

Vývoj ve jmenné službě během uplynulých 10 let asi nejlépe dokumentují uvedené 4 grafy. Pro pochopení souvislosti změn je třeba připomenout některá historická data. Zpoplatnění domén spolu s provozem registračního systému RSD bylo zahájeno 1. 9. 1999. Provoz distribuovaného registračního systému DSD s jedním registrátorem LRR byl zahájen 15. 9. 2003, provoz komerčních registrátorů v DSD začal 13. 10. 2003. Pokles počtu delegovaných domén na obr. 1 je způsoben změnou pravidel znemožňující delegaci nezaplacených a tedy i většinou spekulativně blokovaných domén. Tentýž pokles je i promítnut do obr. 3. Obr. 4 více odráží změny provozovatelů s převažujícím podílem trhu. Až do poloviny roku 1999 byli největšími provozovateli jmenných služeb CESNET a EUnet. Od druhé poloviny roku 1999 se jako největší provozovatelé střídaly hostingové firmy (Zoner, Globe, Internet CZ).

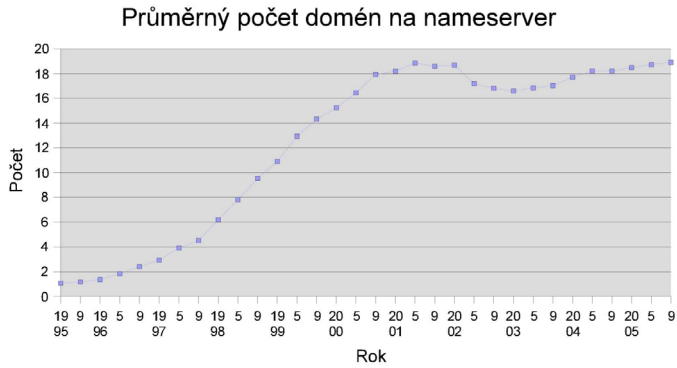



Obr. 1 Počet delegovaných domén

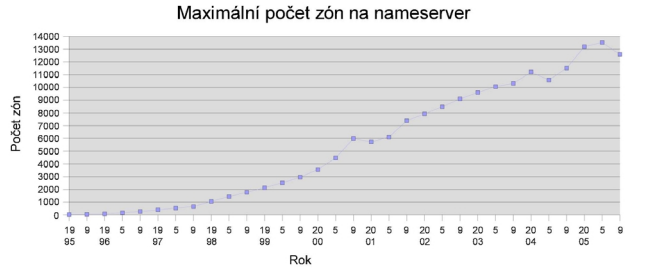



Obr. 2 Počet jedinečných hostů v NS záznamech

Poklesy na obr. 4 jsou způsobeny změnami u největších poskytovatelů jmenných služeb, ať už v důsledku technických změn (migrace části domén na nové servery) nebo v důsledku akvizic a organizačních změn u těchto firem. Přesto stojí za zdůraznění, že na pouhých 30 serverech bylo v září 2005 provozováno 209




Obr. 3 Průměrný počet domén na nameserver

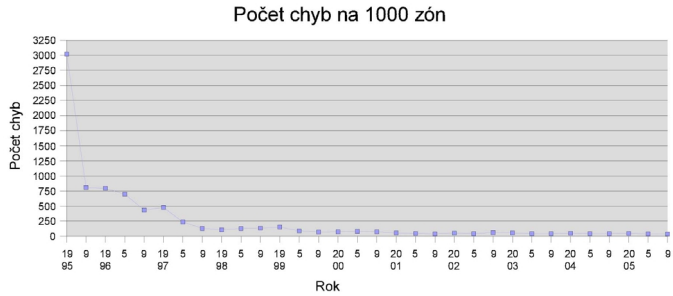



Obr. 4 Maximální počet zón na nameserver

tis. zón pod .cz. Výpadek nebo konfigurační problém na těchto serverech může ovlivnit významnou část služeb na českém internetu. Z tohoto pohledu musíme posuzovat i konfigurační problémy na těchto serverech. Stagnující průměrný počet domén na server svědčí o rostoucím počtu malých a střední poskytovatelů jmenných služeb. Maximální počty zón na serveru zůstávají na technicky přijatelných hodnotách. Celkový vývoj struktury jmenných služeb je tedy možné považovat za spíše pozitivní.

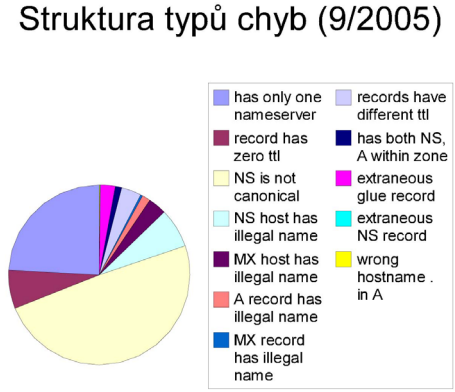
# 3 Typické chyby v konfiguraci jmenné služby

Schopnost detekovat konfigurační chyby u tak velkého objemu serverů je dána schopností provést transfer zóny (AXFR). Podíl zón, u kterých je transfer zóny ze všech nameserverů v delegaci zakázán, byl v září 2005 29,9 %. Technické testy pro delegaci domény z .cz, používané v registračním systému před 15. 9. 2003, vyžadovaly transfer zóny. Proto se tehdy podíl zón se zakázaným transferem pohyboval kolem 5 %. Vývoj počtu chyb na 1 000 delegovaných zón je znázorněn na obr. 5.




Obr. 5 Počet chyb na 1 000 delegovaných zón

Z obrázku je vidět, že i když absolutní počet chyb trvale roste, počet chyb vztažený na počet zón se příliš nemění. Optimistická interpretace ocení, že i přes růst počtu zón a příchod nových administrátorů, relativní poměr chyb neroste. Pesimista poukáže na to, že nejsme schopni poučit se z vlastních chyb a dlouhodobě se spokojujeme s konfiguracemi, jejichž funkčnost je diskutabilní. Poměrně významně se ale v čase mění struktura detekovaných chyb. To je přímým důsledkem používání nových implementací jmenných serverů, ale i změn v přístupu administrátorů. V roce 1995–1997 představoval neúplný HINFO záznam kolem 6 % všech chyb. V současnosti v testovaných zónách existuje sice 6 786 HINFO záznamů (z toho 5 156 v doménách pod zcu.cz), ale již od roku 2 000 nebyl detekován neúplný HINFO záznam. Stejně díky důslednějším implementacím správy cache vymizely nadbytečné SOA záznamy v zónách. Struktura detekovaných chyb ve výsledcích ze září 2005 je rozdělena podle obr. 6.




Obr. 6 struktura typů chyb pro data ze září 2005

Dále popíšeme jednotlivé typy chyb. Pro připomenutí názvosloví uvedeme příklad:

*profiindex.cz.* *18000* *IN* *NS* *ns.system.cz.*

v tomto záznamu je *profiindex.cz.* označován jako vlastník, *18000* je hodnota TTL (Time To Live) v sekundách, *IN* je třída (IN pro Internet, CH pro Chaos), *NS* je označení typu záznamu (např. A, CNAME, MX, HINFO, SOA), *ns.system.cz.* je označován jako RDATA. Celý záznam bývá označován jako RR (resource record). Podrobnější popis lze nalézt např. v [3].

## 3.1 NS is not canonical

Tato konfigurační chyba je dlouhodobě nejběžnější (49 %). Může se objevovat v řadě variant s různými provozními dopady. Internetový standard [3] požaduje v kapitole 3.6.2, aby RDATA neobsahovala alias vytvořený pomocí záznamu CNAME:

„*Domain names in RRs which point at another name should always point at the primary name and not the alias. This avoids extra indirections in accessing information.*“

Jako příklad této chyby můžeme uvést současnou definici domény *profik.cz*, která z .cz domény je delegována jako:

| | | | | |
|---|---|---|---|---|
| *profik.cz.* | *18000* | *IN* | *NS* | *ns.profik.cz.* |
| *profik.cz.* | *18000* | *IN* | *NS* | *ns.profik.net.* |
| *profik.cz.* | *18000* | *IN* | *NS* | *sns.profik.cz.* |
| *ns.profik.cz.* | *18000* | *IN* | *A* | *81.31.15.60* |
| *sns.profik.cz.* | *18000* | *IN* | *A* | *195.47.85.77* |

Ve vlastní zóně je pak jako NS list uvedeno:

| | | | | |
|---|---|---|---|---|
| *profik.cz.* | *60* | *IN* | *NS* | *ns.profik.net.* |
| *profik.cz.* | *60* | *IN* | *NS* | *ns.profik.cz.* |
| *ns.profik.cz.* | *60* | *IN* | *A* | *81.31.15.60* |
| *sns.profik.cz.* | *60* | *IN* | *CNAME* | *ns.profik.net.* |

Většina dotazů na informace v zóně *profik.cz* skončí u novějších verzí nameserveru s vcelku použitelným obsahem cache:

| | | | |
|---|---|---|---|
| *profik.cz.* | *874* | *NS* | *ns.profik.cz.* |
| | *874* | *NS* | *ns.profik.net.* |
| | *874* | *NS* | *sns.profik.cz.* |
| *; glue* | | | |
| *ns.profik.cz.* | *874* | *A* | *81.31.15.60* |
| *; glue* | | | |
| *sns.profik.cz.* | *874* | *A* | *195.47.85.77* |

Poměrně vysoká je, ale i pravděpodobnost kombinace záznamů v cache:

| | | | | |
|---|---|---|---|---|
| *sns.profik.cz.* | *363* | *IN* | *CNAME* | *ns.profik.net.* |
| *sns.profik.cz.* | *874* | *A* | *195.47.85.77* | |

Chování pro jednotlivé uživatele na síti bude záviset na verzích použitých nameserverů, jejich konfiguraci (např. fetch-glue, rrset-order), typu a konfiguraci rezolveru. Výsledkem mohou být pomalejší odezvy, občasné výpadky rezolucí, celková menší dostupnost služeb pro servery v zóně. A to v konečných důsledcích pro všech 83 zón, které tento nameserver obsluhuje.

## 3.2 only one namerver

Jedná se o druhou nejběžnější chybu (24 % v září 2005). Ilustrativním příkladem této konfigurační chyby jsou jmenné služby v doméně *mfcr.cz*. Doména *mfcr.cz* je z .cz delegována takto:

| | | | | |
|---|---|---|---|---|
| *mfcr.cz.* | *18000* | *IN* | *NS* | *ns.eunet.cz.* |
| *mfcr.cz.* | *18000* | *IN* | *NS* | *ns1.mfcr.cz.* |
| *mfcr.cz.* | *18000* | *IN* | *NS* | *nic.eunet.cz.* |

Z domény *mfcr.cz* je delegována doména Celní správy *cs.mfcr.cz*:

| | | | | |
|---|---|---|---|---|
| *cs.mfcr.cz.* | *86400* | *IN* | *NS* | *ns1.mfcr.cz.* |
| *cs.mfcr.cz.* | *86400* | *IN* | *NS* | *ns2.mfcr.cz.* |
| *cs.mfcr.cz.* | *86400* | *IN* | *NS* | *ppc.cs.mfcr.cz.* |
| *ppc.cs.mfcr.cz.* | *86400* | *IN* | *A* | *193.179.220.100* |

Zóna *cs.mfcr.cz* obsahuje v seznamu NS záznamů pouze jednu delegaci:

| | | | | |
|---|---|---|---|---|
| *cs.mfcr.cz.* | *3600* | *IN* | *NS* | *ppc.cs.mfcr.cz.* |

Tato konfigurace je v rozporu se standardem [3], který požaduje:

„*The administrators of both zones should insure that the NS and glue RRs which mark both sides of the cut are consistent and remain so.*“

Doporučení pro počet nameserverů v delegaci je obsaženo i v informačním RFC [4]:

„*You are required to have at least two nameservers for every domain, though more is preferred. Have secondaries outside your network. If the secondary isn't under your control, periodically check up on them and make sure they're getting current zone data from you.*“

Důsledky takto provozované služby můžeme ukázat na 2 případech dotazů na SOA *cs.mfcr.cz* položených v náhodném časovém intervalu:

| | | | | |
|---|---|---|---|---|
| *;; QUESTION SECTION:* | | | | |
| *;cs.mfcr.cz.* | | *IN* | *SOA* | |
| *;; ANSWER SECTION:* | | | | |
| *cs.mfcr.cz.* | *3255* | *IN* | *SOA* | *ppc.cs.mfcr.cz.* |
| | | | | *hostmaster.cs.mfcr.cz.* |
| *4002022251 3600 600 86400 3600* | | | | |

```
;; AUTHORITY SECTION:
cs.mfcr.cz.          86400    IN  NS     ns1.mfcr.cz.
cs.mfcr.cz.          86400    IN  NS     ns2.mfcr.cz.
cs.mfcr.cz.          86400    IN  NS     ppc.cs.mfcr.cz.
;; ADDITIONAL SECTION:
ns1.mfcr.cz.         86400    IN  A      193.86.123.21
ns2.mfcr.cz.         86400    IN  A      193.86.123.22
```

Druhý dotaz zhruba po 1 hodině poskytne odlišnou odpověď:

```
;; QUESTION SECTION:
;cs.mfcr.cz.                  IN  SOA
;; ANSWER SECTION:
cs.mfcr.cz.          3600     IN  SOA    ppc.cs.mfcr.cz.
                                         hostmaster.cs.mfcr.cz.
4002022251 3600 600 86400 3600
;; AUTHORITY SECTION:
cs.mfcr.cz.          2581     IN  NS     ppc.cs.mfcr.cz.
;; ADDITIONAL SECTION:
ppc.cs.mfcr.cz.      2581     IN  A      193.179.220.100
```

Je zřejmé, že tento způsob provozování jmenné služby může vést k pozoruhodným efektům. Druhým, poněkud odlišným příkladem je doména aesa.cz, která je z .cz delegována takto:

```
aesa.cz.             18000    IN  NS     gw.autoesa.cz.
aesa.cz.             18000    IN  NS     www.autoesa.cz.
```

Seznam nameserverů ve vlastní zóně pak poskytne NS dotaz:

```
;; QUESTION SECTION:
;aesa.cz.                     IN  NS
;; ANSWER SECTION:
aesa.cz.             86400    IN  NS     localhost.
;; ADDITIONAL SECTION:
localhost.           604800   IN  A      127.0.0.1
```

O potenciálu problémů spojených s provozováním podobné konfigurace je asi zbytečné mluvit.

## 3.3 host has illegal name

Jedná se o případy, kdy je závada v části záznamu, který je označován jako RDATA. Těchto chyb je méně (10,5 %). Obvykle je v těchto případech tam, kde má být plně kvalifikované doménové jméno použita IP adresa. Tyto konfigurace odporují standardu [3]. Jako příklady z aktuálních zón můžeme uvést:

| | | | | |
|---|---|---|---|---|
| *obedy.ipvz.cz.* | *3600* | *IN* | *NS* | *212.47.22.60.* |
| *obedy.ipvz.cz.* | *3600* | *IN* | *NS* | *tubera.ipvz.cz.* |
| *tubera.ipvz.cz.* | *3600* | *IN* | *A* | *212.47.22.60* |

nebo pro MX záznam:

| | | | | |
|---|---|---|---|---|
| *c.cz.* | *86400* | *IN* | *NS* | *ns.pardub.cz.* |
| *c.cz.* | *86400* | *IN* | *NS* | *ns2.pardub.cz.* |
| *c.cz.* | *86400* | *IN* | *MX* | *0 212.71.158.93.* |
| *c.cz.* | *86400* | *IN* | *A* | *193.85.233.66* |

V obou případech jsou jsou citace zóny uvedeny i se záznamy, které měly podle autorů zřejmě vylepšit funkčnost. V prvním příkladu tak byla přidána i správná verze NS záznamu, takže při běžné rezoluci opravdu získáme:

| | | | | |
|---|---|---|---|---|
| *;; QUESTION SECTION:* | | | | |
| *;obedy.ipvz.cz.* | | *IN* | *NS* | |
| *;; ANSWER SECTION:* | | | | |
| *obedy.ipvz.cz.* | *999* | *IN* | *NS* | *tubera.ipvz.cz.* |

V druhém případě sice vždy získáme MX záznam s IP adresou, ale je tady i použitelný A záznam. Výsledek bude tedy záviset na použitém programu odesílajícím e-mail, část pošty skončí při pokusu o doručení na nefunkčním MTA odkazovaném MX záznamem, část na serveru odkazovaném A záznamem. Příklad ukazuje, že „uživatelsky vstřícná“ implementace nemusí vždy přinést lepší funkčnost.

## 3.4 Problémy spojené s nastavením TTL pro záznam

TTL je časová hodnota v sekundách vyjádřená jako 32 bitové číslo. Standardy možné nastavení TTL neomezují. Hodnota TTL nastavená na 0 znamená, že záznam není uložen do cache. Toto nastavení je spolu s dynamickými update často používáno v konfiguraci síťových zařízení pro rozkládání zátěže. Vzhledem k různorodosti implementací rezolverů bych ale považoval za bezpečnější a stejně účinné použití malého kladného čísla (např. TTL 2 nebo 3). Druhým extrémem je použití vysokého TTL, např. v zóně *vzp.cz, vsb.cz*:

| | | | | |
|---|---|---|---|---|
| *ibm.vsb.cz.* | *9999999* | *IN* | *NS* | *cws.ibm.vsb.cz.* |
| *ibm.vsb.cz.* | *9999999* | *IN* | *NS* | *decsys.vsb.cz.* |
| *abacus.ibm.vsb.cz.* | *9999999* | *IN* | *CNAME* | *abacus.vsb.cz.* |
| *cws.ibm.vsb.cz.* | *9999999* | *IN* | *A* | *158.196.254.100* |
| *vzp.cz.* | *99999999* | *IN* | *MX* | *10 dns.vzp.cz.* |
| *vzp.cz.* | *99999999* | *IN* | *MX* | *20 relay.iol.cz.* |

Lze jen těžko věřit tomu, že administrátoři *vsb.cz* jsou schopni plánovat změny s předstihem 116 dní a 3 roky u *vzp.cz* je v situaci našeho zdravotnictví mimo veškerou realitu. Naštěstí řada implementací nameserverů umožňuje pro cachovací

jmenný server omezit maximální hodnotu TTL (např. u ISC BIND 9 direktiva *max-cache-ttl*). Vzhledem k různosti implementací nameserverů ale používání podobných hodnot TTL rozhodně nelze doporučit.

Již v spíše v kategorii konfiguračních chyb je použití různých hodnot TTL pro záznamy stejného typu a stejného vlastníka:

| | | | | |
|---|---|---|---|---|
| *bmw.cz.* | *1080* | *IN* | *MX* | *10 pascal31.sprinx.cz.* |
| *bmw.cz.* | *64800* | *IN* | *MX* | *50 seneca.sprinx.cz.* |

U takto nastavených MX v zóně *bmw.cz* bude pošta v první tři hodiny po rezoluci doručována na *pascal31.sprinx.cz*, dalších 15 hodin pak bude doručována na *seneca.sprinx.cz.* Minimálním dopadem (při dobré konfiguraci sekundárního MTA) bude tedy výrazné zpoždění při doručování elektronické pošty.

Problematických hodnot TTL je z detekovaných chyb zhruba 10 %.

## 3.5 Zbytečný záznam v zóně

Jedná se zejména o případy zbytečných spojovacích záznamů, kdy ze zóny je delegována subdoména. Pokud jsou nameservery z této subdomény, musí v rodičovské zóně být spojovací záznamy. V uvedeném příkladě je ale záznam pro *ns.finos.cz* zbytečně.

| | | | | |
|---|---|---|---|---|
| *znalecka.cz.* | *3600* | *IN* | *A* | *195.146.96.125* |
| *znalecka.cz.* | *3600* | *IN* | *NS* | *ns.finos.cz.* |
| *znalecka.cz.* | *3600* | *IN* | *NS* | *ns.czcom.cz.* |
| *znalecka.cz.* | *3600* | *IN* | *MX* | *0 hk.finos.cz.* |
| *znalecka.cz.* | *3600* | *IN* | *MX* | *10 smtpb-in-2.worldonline.cz.* |
| *archiv.znalecka.cz.* | *3600* | *IN* | *NS* | *ns.finos.cz.* |
| *ns.finos.cz.* | *3600* | *IN* | *A* | *195.146.96.113* |

Podobné efekty mohou být způsobeny, jak chybou administrátora při vytváření zóny, tak chybou implemetace při uložení zóny. Vzhledem k rizikům „pharmingu“ byly filtrace přenesených dat v běžných implementacích významně zlepšeny. To snížilo i možný dopad rizik spojená se zbytečnými spojovacími záznamy.

## 3.6 record has illegal name

Jedná se o případy, kdy je závada v části záznamu, který je označován jako vlastník. Syntaxe doménového jmén je diskutována v [5, 6, 7] včetně často diskutované otázky použití znaku „_“. Podíl těchto chyb je relativně malý (pod 2 %). Jsou to jednak nevhodné komentářové znaky nebo escapové sekvence, jednak nepřípustné znaky v jménu.

| | | | | |
|---|---|---|---|---|
| *313masa\197\153\195\173k.ueb.cas.cz.* | *900* | *IN* | *A* | *147.231.138.110* |
| *3+S.tiscali.cz.* | *120* | *IN* | *A* | *195.146.101.222* |
| *#relay.vzp.cz.* | *86400* | *IN* | *A* | *194.228.11.131* |
| *#relay2.vzp.cz.* | *86400* | *IN* | *MX* | *9 relay2.vzp.cz.* |
| *//www.roman-horky.cz.* | *1800* | *IN* | *CNAME* | *virtwww.trnet.cz.* |
| *\@.zusdunicka.cz.* | *86400* | *IN* | *A* | *82.208.14.69* |

Tyto chybné zaznamy sice neposkytují požadovanou funkcionalitu, ale představují i poměrně nízké provozní riziko. K rozsáhlejším dopadům by mohla vést kombinace exotické implementace nameserveru spolu s nedokonale ošetřenou rezolucí v internetové aplikaci. Poněkud nebezpečněji vypadá následující použití „.“ místo kvalifikovaného jména:

| | | | | |
|---|---|---|---|---|
| *helpweb.cz.* | *86400* | *IN* | *NS* | *ns.flyweb.cz.* |
| *www.* | *86400* | *IN* | *CNAME* | *.* |
| *free.helpweb.cz.* | *86400* | *IN* | *A* | *81.95.103.34* |
| *.* | *86400* | *IN* | *A* | *81.95.103.34* |

V tomto případě se vždy jedná o použití stejné implementace jmenného serveru PowerDNS, bez podrobnější analýzy ovšem není zřejmá geneze tohoto typu chyby, ani nelze odhadnout jestli při jisté změně konfigurace (např. použití rekurzivního backendu) nedojde k problémům.

## 3.7 Použití * v doménovém jméně

Wildcard byl v dřivějších dobách poměrně užitečný nástroj usnadňující především správu elektronické pošty. S růstem spamového provozu, ale podle mého názoru přínos wildcardu klesá. Kromě MX záznamů, je možné používat wildcard i v A a CNAME záznamech. Tam je však použití wildcard ještě problematičtější, jak ukazuje následující příklad:

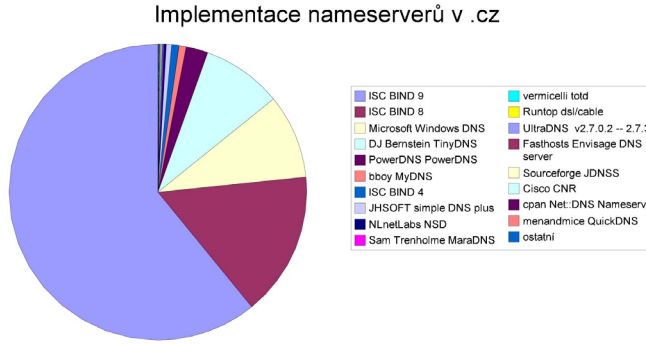
| | | | | |
|---|---|---|---|---|
| **.fbi.cz.* | *86400* | *IN* | *A* | *82.208.14.91* |
| *www.*.fbi.cz.* | *86400* | *IN* | *A* | *82.208.14.90* |

Na dotaz na libovolné jméno ve tvaru *www.cokoliv.fbi.cz* dostaneme adresu *82.208.14.91*, což je v rozporu s očekáváním autora, ale v plném souladu s definicí. Je pravděpodobné, že s větším užíváním DNSSEC a v důsledku větší komplexity NSEC s wildcard, bude konfigurací s wildcard pro A nebo CNAME záznamy v budoucnu ubývat.

# 4 Implementace jmenných serverů a jejich zastoupení v .cz

Pro odhad celkové stability a bezpečnosti jmenných služeb je klíčovou informací rozložení četnosti jednotlivých implementací nameserverů. Při testování jsem kromě odhadu typu nameserveru pomocí fingerprintu [8], zjišťoval i informaci o typu nameserveru v třídě CHAOS a to jestli server dovoluje rekurzivní dotazy bez omezení zdrojové adresy klienta. Data vychází z opakovaného testování 11111 nameserverů. 464 nameserverů (4,2 %) bylo nedostupných nebo za velmi špatnou konektivitou. U dalších zhruba 800 nameserverů se jednalo buď o neexistující doménová jména nebo o neexistující doménu.

Kvalitnější analýza těchto zhruba 10 % problémových namerverů bude vyžadovat další programové úpravy. Výsledky testů jsou uvedeny na obr. 7.




Obr. 7 implementace na nameserverech pro domény 2. úrovně pod .cz (září 2005)

Podrobnější rozdělení na verze implementací je uvedeno v Příloze A. Při porovnání zjištěných informací s posledními bezpečnostními informacemi, jsem se pokusil odhadnout podíl nameserverů s pravděpodobným bezpečnostním problémem. To není úplně jednoduché vzhledem k nemožnosti zkoušet přímo bezpečnostní problémy (často DoS). Někteří výrobci navíc dodávají bezpečnostní opravy beze změny verze, některá bezpečnostní rizika jsou vázána na specifickou konfiguraci. Přesto odhad 40% nameserverů s „podezřelými“ verzemi jmenného serveru mi připadá extrémně nepříznivý, zvláště když po dalším zpracování je možné zjistit, že tyto namervery obsluhují více než polovinu zón pod .cz. I tento výsledek bude vyžadovat další detailnější analýzu.

# 5 Systémové problémy jmenných serverů pro doménu .cz

V předchozí kapitole jsme ověřovali instalované verze jmenných serverů. Kromě bezpečnostních rizik spojených s instalovanou verzí existují rizika daná vlastním návrhem DNS a rizika způsobená přetížením či nedostatečnou konektivitou jmenných serverů. Významné ohrožení představuje „pharming“, tedy podvržení nesprávného výsledku rezoluce do cache nameserveru. Toto riziko lze do značné míry eliminovat návrhem topologie DNS (např. oddělením serverů poskytujících rezoluce klientům od serverů s delegovanými zónami, omezením odpovídání na rekurzivní dotazy, eliminací cache). Z pohledu ochrany konzistence cache jmenných serverů není příliš povzbudivá skutečnost, že z 11111 serverů používaných v zónách pod .cz poskytuje neomezené rekurzivní služby 5 882 serverů (53 %) na něž je delegováno 179 196 zón.

Pro posouzení přiměřené funkčnosti jmenného serveru je dobré sledovat množství dotazů za sec, které je nameserver schopen zodpovědět a podíl odpovězených dotazů. Pro toto měření byl použit nástroj [15]. Servery obsluhující více než 500 zón jsem testoval sadou 100 dotazů z obsluhované zóny. Dotazy obsahovali existující a neexistující záznamy v poměru 1 : 1. Testování proběhlo 2×, jednou během noci, jednou během pracovního dne. V příloze B jsou uvedeny výsledky obou testů pro servery s horším výkonem než 100 dotazů za sekundu. Pro porovnání je dobré vědět, že odezva nameserveru za 64 kbps linkou je kolem 50 dotazů za sec, primární server pro .cz ns.tld.cz je schopen odpovědět mezi 25 000–30 000 dotazy za sec, průměrný výsledek v testu se pohyboval kolem 2000 dotazů za sec. Je zřejmé, že nameservery uvedené v obou tabulkách nejsou zárukou spolehlivého fungování DNS.

# 6 Závěrečná doporučení

Závěrečná doporučení jsem se pokusil shrnout do následujících bodu:

- O funkčnosti DNS rozhoduje spíše kvalitní návrh a pečlivá konfigurace, než výkonný HW.
- Nameserver „na páteři ISP“ není zárukou služby srovnatelné s nameserverem v komoře za 64 kbps digitální linkou.
- Připojení nameserveru přes bezdrátové technologie vede často k nepříjemným ztrátám dotazů a velkému rozptylu zpoždění.
- DNS je snadno čitelnou vizitkou ISP, není důvod čekat funkční síť u někoho, kdo neumí provozovat své DNS.

- Při implementaci nových technologií by stejně jako vždy měla mít přednost stabilita a bezpečnost. Použít TSIG pro ochranu komunikace mezi nameservery je dobrý nápad.
- DNS je velmi odolné k hrubému zacházení, má ale paměť a prohřešky vrací i po dlouhé době a poměrně záludně. Účinná ochrana je dodržování standardů.
- Pokud vám nefungují uspokojivě služby, podívejte se i na své DNS, než změníte poskytovatele konektivity.

# Literatura

[1] Projekt *Hostcount*, http://www.ripe.net/hostcount/

[2] Wassenaar Eric, Program *host*, NIKHEF.

[3] Mockapetris, P.: *Domain Names – Concepts and Facilities.*, STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.

[4] Barr, D.: *Common DNS Operational and Configuration Errors.* RFC 1912, The Pennsylvania State University, February 1996.

[5] Mockapetris, P.: *Domain Names – Implementation and Specification.* STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[6] Braden, R.: *Requirements for Internet Hosts – Application and Support.* STD 3, RFC 1123, IETF, October 1989.

[7] Lottor, M.: *Domain Administrators Operations Guide.* RFC 1033, USC/Information Sciences Institute, November 1987.

[8] Arends, R., Schlyter, J.: *Fingerprinting DNS servers.* 2004.

[9] *CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND.* January 29, 2001.

[10] *CERT Advisory CA-98.05, Multiple Vulnerabilities in BIND.* April 08, 1998.

[11] *US-CERT Vulnerability Note VU#327633, BIND 8.4.4 and 8.4.5 vulnerable to buffer overflow in q_usedns.*

[12] *US-CERT Vulnerability Note VU#938617, BIND 9.3.0 vulnerable to denial of service in validator code.*

[13] *US-CERT Vulnerability Note VU#109475, Microsoft Windows NT and 2000 Domain Name Servers allow non-authoritative RRs to be cached by default.*

[14] *US-CERT Vulnerability Note VU#714121, Incorrect NXDOMAIN responses from AAAA queries could cause denial-of-service conditions.*

[15] Jacob, S.: *DNS Query Performance Testing Tool.* June 2004.

# Příloha A

| | |
|---|---|
| Cisco CNR | 3 |
| DJ Bernstein TinyDNS 1.04 | 7 |
| DJ Bernstein TinyDNS 1.05 | 861 |
| Fasthosts Envisage DNS server | 5 |
| ISC BIND 4.8 – 4.8.3 | 4 |
| ISC BIND 4.9.3 – 4.9.11 | 70 |
| ISC BIND 8.1-REL – 8.2.1-T4B | 275 |
| ISC BIND 8.2.2-P3 – 8.3.0-T2A | 120 |
| ISC BIND 8.3.0-RC1 – 8.4.4 | 1141 |
| ISC BIND 8.4.1-p1 | 5 |
| ISC BIND 9.0.0b5 – 9.1.3 | 4 |
| ISC BIND 9.1.0 – 9.1.3 | 114 |
| ISC BIND 9.2.0a1 – 9.2.0rc3 | 2 |
| ISC BIND 9.2.0a1 – 9.2.2-P3 | 12 |
| ISC BIND 9.2.0rc4 – 9.2.2-P3 | 2 |
| ISC BIND 9.2.0rc7 – 9.2.2-P3 | 2150 |
| ISC BIND 9.2.3rc1 – 9.4.0a0 | 3682 |
| JHSOFT simple DNS plus | 56 |
| Meilof Veeningen Posadis | 1 |
| Microsoft Windows DNS 2000 | 701 |
| Microsoft Windows DNS 2003 | 134 |
| Microsoft Windows DNS NT4 | 73 |
| Mikrotik dsl/cable | 1 |
| NLnetLabs NSD 1.0.3 – 1.2.1 | 1 |
| NLnetLabs NSD 1.2.2 | 2 |
| NLnetLabs NSD 1.2.3 – 2.1.2 | 8 |
| NLnetLabs NSD 2.1.3 | 14 |
| PowerDNS PowerDNS 2.8 – 2.9.3 | 105 |
| PowerDNS PowerDNS 2.9.4 – 2.9.11 | 129 |
| Runtop dsl/cable | 9 |
| Sam Trenholme MaraDNS | 12 |
| Sourceforge JDNSS | 3 |
| UltraDNS v2.7.0.2 – 2.7.3 | 8 |
| bboy MyDNS | 75 |
| cpan Net::DNS Nameserver | 2 |
| menandmice QuickDNS | 2 |
| vermicelli totd | 11 |
| **Celkem** | 9804 |

# Příloha B

Tabulka 1 výsledky 20050928 noc

| *Nameserver* | *% zodpovězených dotazů* | *Dotazů za sec (qps)* | *Počet obsluhovaných zón* |
|---|---|---|---|
| ns2.skynet.cz | 98 | 19,51 | 2 570 |
| ns.goodlife.cz | 99 | 19,8 | 2 142 |
| gold.ns.cz | 90 | 17,97 | 1 140 |
| ns2.onlinehosting.cz | 96 | 19,04 | 1 096 |
| ns.miton.cz | 99 | 19,8 | 933 |
| ns2.gigaweb.cz | 97 | 19,35 | 799 |
| ns2.hostingzdarma.cz | 99 | 19,77 | 534 |
| slave.casablanca.cz | 98 | 19,48 | 523 |
| ns.banan.cz | 98 | 19,58 | 511 |

Tabulka 2 výsledky 20050929 den

| *Nameserver* | *% zodpovězených dotazů* | *Dotazů za sec (qps)* | *Počet obsluhovaných zón* |
|---|---|---|---|
| ns.forpsi.net | 97 | 19,32 | 8 386 |
| ns2.skynet.cz | 92 | 18,27 | 2 570 |
| ns.explorer.cz | 80 | 15,83 | 2 155 |
| ns1.one.cz | 88 | 17,6 | 1 170 |
| ns1.onlinehosting.cz | 80 | 15,69 | 1 096 |
| ns.mitoncz.com | 99 | 19,8 | 934 |
| secure.abzone.cz | 80 | 15,91 | 901 |
| ns2.gigaweb.cz | 83 | 16,56 | 799 |
| ns2.gigant.cz | 80 | 15,65 | 766 |
| ns.kraxnet.com | 85 | 16,81 | 765 |
| ns1.antee.cz | 99 | 19,72 | 599 |
| ns.tripsi.com | 99 | 19,77 | 524 |
| slave.casablanca.cz | 96 | 19,04 | 523 |
| ns.banan.cz | 73 | 7,22 | 511 |
| ns.profiwh.cz | 99 | 11,27 | 503 |

# Vývoj DNS standardů a technologií

**Pavel Satrapa**

E-mail: pavel.satrapa@vslib.cz

**Abstrakt**

*Dvacetiletá historie DNS ověřila jeho vynikající koncepci a škálovatelnost. Během této doby se však objevila řada novinek a požadavků, s nimiž se DNS muselo vypořádat. Nejvýznamnějšími milníky jsou podpora IPv6, mezinárodní doménová jména, ukládání informací pro IP telefonii a zabezpečení DNS. Příspěvek poskytuje základní informace o těchto rozšířeních DNS a jejich nelehké cestě do reálného života.*

**Abstract**

*Twenty years of DNS proved its excellent concept and scalability. Many novelties and requirements arose during this time demanding DNS to handle them. The most important are IPv6 support, internationalized domain names, storing of IP telephony-related information, and securing DNS. This article provides basic information about these DNS extensions and about their difficult deployment in real world.*

Domain Name System již oslavil své dvacáté narozeniny. Jeho základní myšlenky a protokoly vznikly počátkem osmdesátých let – v době, kdy počet strojů připojených k Internetu čerstvě překročil tisícovku. Skutečnost, že slouží stále stejně dobře i po dvaceti letech, během nichž počet počítačů v Internetu vzrostl přibližně 350 000krát (viz [16]), během nichž nedošlo k žádnému fatální výpadku, ani nebylo nutno DNS radikálněji měnit, hovoří sama za sebe.

Nicméně dvacet let představuje ve světě informačních technologií obrovský časový prostor – jako příklad se stačí podívat na vývoj PC od roku 1981. V Internetu se během té doby pochopitelně odehrály významné změny, jež se do DNS nemohly nepromítnout. Odrážely se však především na jeho obsahu, tedy na vývoji a nasazení nových typů záznamů. Podívejme se alespoň na několik nejvýznamnějších změn, které zásadnějším způsobem zasáhly do historie DNS.

# 1 Podpora IPv6

Existence nového základního protokolu celého Internetu se v DNS odráží ve dvou oblastech: je třeba umožnit jeho použití jako přenosového protokolu pro výměnu dotazů a odpovědí, ale především musí DNS mít prostředky pro poskytování informací o IPv6 adresách. IPv6 jako transportní médium pro DNS komunikaci nepředstavuje žádný zásadnější problém. Službu obsahově nijak nemění, de facto znamená jen nutnost upravit implementaci DNS serverů a klientů. V nejrozšířenějším serveru, programu BIND, se objevil počínaje verzí 8.4 v roce 2003.

Naproti tomu pronikání IPv6 adres do obsahu DNS bylo daleko bolestnější a při pohledu na jeho historii se vnucuje pojem „tragikomedie“. Koncem roku 1995 vyšlo [2] s velmi přímočarou definicí podpory IPv6. Jednalo se vlastně o přímou analogii IPv4. Pro běžné dotazy zavedla tato specifikace záznam AAAA obsahující IPv6 adresu přiřazenou danému jménu – například

```
cosi  AAAA  2001:718:1c01:16:20d:56ff:fe77:52a3
```

Reverzní dotazy vycházely ze standardního zápisu IPv6 adres v šestnáctkové soustavě. Zápis adresy se otočí a každá její číslice představuje samostatnou doménu. Používá se standardní záznam PTR, stejně jako pro reverzní záznamy IPv4 adres. Liší se pouze doména připojovaná na konec obráceného zápisu – v případě IPv6 se jedná o *ip6.int*.

Vrátíme-li se k předchozímu příkladu, jeho reverzní záznam by měl kompletní tvar *3.a.2.5.7.7.e.f.f.f.6.5.d.0.2.0.6.1.0.0.1.0.c.1.8.1.7.0.1.0.0.2.ip6.int*. Jelikož počáteční tři čtveřice IPv6 adresy tvoří prefix sítě, byl by záznam uložen v zónovém souboru pro doménu *1.0.c.1.8.1.7.0.1.0.0.2.ip6.int* a měl by tvar

```
3.a.2.5.7.7.e.f.f.f.6.5.d.0.2.0.6.1.0.0  PTR  cosi.kdesi.cz.
```

Pravda, reverzní domény vypadají vyloženě děsivě, ovšem princip vycházel z ověřeného IPv4. O pět let později vyšlo [3] s pokusem o malou revoluci. Už samotný jeho název napovídá, že autoři se snažili, aby v sobě DNS záznamy odrážely strukturu adresního prostoru (agregace) a umožňovaly elegantní přeadresování.

Pro zjištění IPv6 adresy měl sloužit záznam A6. Byl postaven na myšlence převzít prefix adresy z jiného A6 záznamu a doplnit k němu specifickou část. Ke zjištění kompletní IPv6 adresy by pak bylo třeba prozkoumat celý řetězec A6 záznamů. Například záznam počítače by využíval prefix podsítě, jehož A6 záznam by se odkazoval na prefix celé zákaznické sítě. Ten by se mohl nacházet v doméně poskytovatele a odkazovat se například na prefix celé sítě tohoto poskytovatele, jenž by mohl být definován v doméně jeho mateřského mezinárodního poskytovatele. Ke zjištění adresy by pak bylo nutné obstarat čtyři A6 záznamy ze tří různých serverů.

Právě tato vlastnost RFC 2874 se stala terčem silné kritiky, protože DNS zpomaluje a navíc činí choulostivějším. Také reverzní dotazy se dočkaly výrazných změn. Objevily se v nich DNAME záznamy umožňující mapovat reverzní dotazy do běžných domén, v zápisu adresy nebylo nutné obracet pořadí číslic, delegovat poddoménu bylo možné na hranici libovolných dvou bitů a celé reverzní dotazování se přestěhovalo z domény *int* do *arpa*.

Nové RFC prohlásilo svého předchůdce za zastaralého a novou metodu za jedině správnou. Jenže řada správců domén se postavila na odpor a vypuklo schizma. Argumentační přestřelky se táhly několik let, zatímco v DNS jste mohli potkávat oba „konkurenční" typy záznamů, a to v doménách *ip6.int* i *ip6.arpa*.

Právě roztříštěnost domén nejvyšší úrovně byla odstraněna jako první. Již v roce 2001 vychází [5], jež jednoznačně přesídluje reverzní domény do *ip6.arpa*. Za zmínku stojí lišácká změna významu zkratky – místo původní zkratky agentury (Advanced Research Projects Agency), jež stála za vznikem Internetu, nyní znamená „Address and Routing Parameters Area".

Postupný odklon od A6 záznamů naznačuje kratičké [6], které doporučuje používat záznamy AAAA, zatímco A6 podle tohoto dokumentu sice přinášejí zajímavé vlastnosti, ale také rizika a je třeba posoudit, zda převažují klady či zápory. Analogicky nedoporučuje používat DNAME pro reverzní záznamy. RFC 2874 je prohlášeno za experimentální.

Celý proces dovršilo [11], jež je de facto opakováním RFC 1886. Definuje AAAA záznamy a reverzní dotazy tvořené doménami po jednotlivých šestnáctkových číslicích zápisu adresy v obráceném pořadí. Jedinou významnější změnou proti RFC 1886 je přesídlení reverzních záznamů do domény *ip6.arpa*. Kruh se uzavřel, RFC 2874 je efektivně zapomenuto.

Jeho příznivci se smířili se skutečností, že elegantní, ale křehké mechanismy nedokáží prosadit. I oni si jistě uvědomují, že vítězství libovolné z variant je lepší, než nejistota a zmatek plynoucí z přetlačování dvou soupeřících koncepcí. A RFC 2874 na vítězství v tomto sporu zkrátka nestačilo.

Podpora IPv6 se nyní konečně nachází ve stabilizovaném stavu a úměrně s postupně rostoucím nasazováním protokolu do reálných sítí roste i počet odpovídajících záznamů.

## 2 Národní jazyky v doménových jménech – IDN

Specifikace DNS (konkrétně [1]) povoluje v doménových jménech používat pouze písmena anglické abecedy, číslice a pomlčky. To je leckde vnímáno jako příliš velké omezení. Jazyky vycházející z latinky, k níž doplňují akcentované znaky (jako třeba čeština), se s ním vyrovnávají celkem snadno. Je sice pravda, že doména *hracky.cz* může sloužit pro hračky nebo hráčky, ale z této dvojznačnosti nikoho hlava příliš nebolí.

V daleko horší situaci jsou jazyky se zcela svébytnými znaky – počínaje azbukou až po čínštinu či japonštinu. Pro ně představuje doménové jméno zapsané latinkou zcela neorganický prvek. Právě z Asie proto pocházel silný tlak na podporu jiných jazyků v DNS.

Nakonec vyústil ve vznik Internationalized Domain Names (IDN), čili specifikace podporující prakticky libovolná doménová jména. Klíčovým požadavkem pochopitelně bylo, aby se kvůli naplnění tohoto požadavku nemusely měnit základy DNS. Výsledné řešení je postaveno na stejné myšlence jako MIME pro elektronickou poštu – jádro systému zůstává beze změny, rozšíření je implementováno v klientech. Proto je podstatnou částí i specifikace IDNA, čili Internationalizing Domain Names in Applications. Celý koncept popisuje [8].

Základem IDN je přesně definovaná konverze jména z kódování UTF8 na sadu znaků přípustných v doménových jménech dle RFC 1034. Je označována jako ACE (ASCII Compatible Encoding). Na doménových serverech jsou jména uložena v podobě zakódované podle ACE, tedy ve tvaru vyhovujícím původním pravidlům. Jestliže uživatel zadá jméno obsahující speciální znaky, DNS klient (resolver) pro ně provede ACE konverzi. Ta má tři kroky:

1. Nameprep (definovaný v [9]) nejprve pomocí různého mapování převede jméno na kanonickou podobu (velká písmena změní na malá a různé zápisy téhož převede na základní variantu, např. různé zápisy čínštiny). Jméno zatím zůstává v UTF8, cílem prvního kroku je omezit počet jeho variant.

2. Punycode (dle [10]) převede znaky mimo rozsah povolený RFC 1034 na sekvence ASCII znaků. Konverze je poněkud zvláštní, protože speciální znaky odkládá až na konec jména, od nějž jsou odděleny pomlčkou.

3. K výsledku předchozího kroku se předřadí předpona `xn--`, jež slouží k rozlišení jmen kódovaných podle ACE.

Například z domény *hračka.cz* by podle těchto pravidel vznikla *xn--hraka-jya.cz*, zatímco *hráčka.cz* se převede na *xn--hrka-6na8x.cz*.

Aplikace by měla provádět převod oběma směry (například při prezentaci výsledků reverzních dotazů, třeba u *traceroute*) a zcela tak před uživatelem skrýt zakódovanou podobu jmen. Na druhé straně pokud aplikace či systém nepodporuje IDNA, přesto se uživatel dokáže k národním doménám dostat. Musí si ale jména konvertovat do ACE sám, například pomocí konvertoru na adrese http://http://josefsson.org/idn.php/. Do WWW klienta pak místo *www.hračka.cz* bude muset napsat *www.xn--hraka-jya.cz*. Vše bude plně funkční, utrpí jen uživatelský komfort.

S neoficiální podporou IDN se začalo ještě před vydáním příslušných RFC. V čele byla Asie, konkrétně Čína (2000) a Japonsko (2001). Od poloviny roku 2003, kdy byla publikována sada RFC s definicí IDN, přibývá domén nejvyšší

úrovně oficiálně podporujících mezinárodní jména. Podíváme-li se na Evropu, první byli hned v září 2003 Poláci. Následovalo Švédsko, Dánsko a Norsko, na jaře 2004 německy mluvící země (Německo, Rakousko, Švýcarsko) a Litva, letos přibylo Řecko, Finsko a Maďarsko. Chcete-li sledovat postup jeho zavádění, doporučuji průběžně aktualizovanou stránku

http://en.wikipedia.org/wiki/Internationalized_domain_names.

Sdružení CZ.NIC, správce domény *cz*, přistupuje k IDN rozumně. Jako podklad pro jeho zavádění si nejprve provedlo průzkum mezi domácími subjekty. Ukázalo se, že valná většina z nich nemá o podporu diakritiky zájem. CZ.NIC proto s jeho zavedením v dohledné době nepočítá. Upřímně řečeno se také domnívám, že se bez háčků a čárek v doménách nějak protlučeme.

Přestože IDN je z hlediska technického vyřešeno velmi pěkně, vyvolává některé nepříjemné otázky politicko-organizačního charakteru. Například jestli má být vlastník „odháčkované“ domény (*hracka.cz*) být nějak upřednostněn při získávání odpovídající domény s diakritikou (*hračka.cz*), nebo zda ponechat zcela volnou soutěž. Obě varianty mají své klady i zápory. A pochopitelně se neustále vtírá principiální otázka, jestli v zemích s jazyky odvozenými od latinky celou tuhle záležitost opravdu potřebujeme, nebo se spíše snaží registrátoři elegantně znásobit své příjmy z domén.

# 3 IP telefonie a ENUM

Internet se postupně vyvíjí jako univerzální médium přenášející všechny možné druhy komunikace, mimo jiné i telefonní hovory. IP telefonie se stává běžnou záležitostí, a to jak mezi operátory, tak mezi koncovými uživateli. Objevují se vize, podle nichž se – vzhledem k objemu přenášených dat – telefonní systém stane v nedaleké budoucnosti jen jednou z mnoha služeb počítačových sítí.

Narážejí tu na sebe ovšem dva světy. V tom telefonním je zvykem identifikovat cíl prostým volacím číslem. Naproti tomu svět počítačových sítí je zvyklý pracovat s protokoly, porty a IP adresami. Nabízí se použít DNS jako databázi, která by umožnila tento rozpor překlenout.

Proto vznikl návrh pojmenovaný ENUM. Jeho cílem je odvodit z telefonního čísla parametry potřebné pro navázání spojení s IP telefonem, jemuž dotyčné číslo náleží. Jeho definici najdete v [12], jež nahrazuje původní [4]. Změny ovšem nejsou nijak dramatické.

Základní myšlenkou ENUM je, že telefonní číslo se přetvoří na doménové jméno a dotazem do DNS se zjistí parametry cílového zařízení – jakým protokolem je oslovit, na jaké adrese a podobně. Transformace čelí podobným problémům jako reverzní dotazy: zatímco v doménových jménech bývají obecné domény na konci a směrem dopředu se zpřesňují, u telefonních čísel je pořadí právě opačné. Začínají identifikací země, následuje město atd. Řešení je také ob-

dobné – otočení pořadí domén. Transformace telefonního čísla na jméno proto probíhá následovně:

1. Vyjde se z kompletního telefonního čísla (včetně kódu země, řekněme +420-485-353-685), z nějž se nejprve vypustí všechny nečíselné znaky. V našem případě vznikne 420485353685.

2. Každá číslice se stane jménem jedné domény:
   4.2.0.4.8.5.3.5.3.6.8.5

3. Obrátí se jejich pořadí, aby se obecné domény ocitly na konci. To umožňuje efektivní delegaci jednotlivých domén podle prefixů telefonních čísel. V našem příkladu vznikne
   5.8.6.3.5.3.5.8.4.0.2.4

4. Na konec se připojí doména *e164.arpa*. Dotaz tedy bude směřovat na
   5.8.6.3.5.3.5.8.4.0.2.4.e164.arpa

K uchovávání informací o IP telefonech slouží záznamy NAPTR (Naming Authority Pointer) definované v [7]. Volající telefon (nebo gateway převádějící hovor z klasického telefonního hovoru na internetový) se tedy obrátí na DNS s žádostí o NAPTR záznam pro doménu odpovídající volanému číslu.

NAPTR záznamy jsou velmi silné. Vedle cílových informací mohou obsahovat i pravidla (v podobě regulárních výrazů), podle nichž má tazatel upravit doménu a vznést nový dotaz. Jinými slovy NAPTR záznam odpovídající telefonnímu číslu může obsahovat instrukci „ptejte se raději na tohle“.

Pronikání ENUM do reálného života teprve začíná – v řadě zemí běží testy či pilotní projekty, jejichž cílem je ověřit vlastnosti této specifikace v reálném provozu a vypracovat mechanismy pro správu odpovídajícího adresního prostoru. Budeme-li mluvit konkrétně o situaci u nás, správa domény *0.2.4.e164.arpa* byla svěřena správci domény *cz*, sdružení CZ.NIC. To připravuje pravidla pro jeho využití.

# 4 DNSSEC – věrohodné DNS

S prorůstáním Internetu do všech sfér života nabývá na důležitosti otázka věrohodnosti informací poskytovaných DNS. Podvrhnout IP adresu nebo přesměrovat něčí telefonní hovory na svůj IP telefon může lákat crackery, nebo být předmětem zájmu kriminálních skupin.

Na obranu před podobnými snahami vznikl projekt zabezpečeného DNS, čili DNSSEC. Na jeho vývoji se pracovalo od poloviny devadesátých let a v roce 1999 dospěly práce do podoby RFC 2535. Zdálo se, že se problém podařilo vyřešit,

objevila se i implementace v BINDu verze 9. Jenže pokusy o praktické nasazení ukázaly, že navržený model je velmi nepružný a v praxi víceméně nepoužitelný.

Trvalo více než pět let, než jeho aktualizovaná verze dozrála do podoby RFC. Stalo se tak na jaře 2005, kdy vyšlo [13] popisující základní pojmy a principy fungování nové verze DNSSEC. Na něj navazují [14] s definicí nových záznamů a [15] popisující změny přenosového protokolu.

Základem zabezpečení je nový záznam RRSIG obsahující digitální podpis skupiny záznamů. Do ní patří všechny záznamy stejného typu pro jedno jméno. Nepodepisují se tedy individuální záznamy, ale celá skupina, jež bývá odesílána najednou jako odpověď na dotaz.

Podepisuje se obvyklými kryptografickými metodami s veřejným klíčem. Tajný klíč je pochopitelně uložen mimo DNS a v ideálním případě zcela mimo Internet. Naopak veřejný klíč, jímž lze ověřit pravost záznamů, je uložen přímo do domény obsahující podepisovaná jména. Slouží k tomu záznamy DNSKEY.

Jejich důvěryhodnost je zajištěna záznamem typu DS, jenž obsahuje otisk (digest) klíče, je uložen v nadřazené doméně a je podepsán jejím klíčem. Jeho otisk je pak opět o patro výše atd. atd., až se ověřování vyšplhá do kořenové domény. Oficiálně se tato sekvence postupného ověřování nazývá řetězec důvěry. Pokud by celý strom DNS využíval DNSSEC, stačila by klientům znalost veřejného klíče kořenové domény, aby dokázali ověřit libovolný záznam.

Zejména v počátečních fázích pochopitelně nelze plošnou přítomnost DNSSEC očekávat. Proto budou klienti zřejmě vybaveni několika veřejnými klíči z nižších pater, jimiž dokáží ověřit alespoň ty části doménového stromu, které jsou pro ně podstatné.

DNSSEC má poskytovat i ověřené negativní informace, čili spolehlivou zprávu o tom, že hledaný údaj v DNS neexistuje. Slouží k tomu záznamy NSEC, které jsou přiřazeny ke každému existujícímu jménu. Obsahují:

- seznam typů záznamů existujících pro toto jméno
- následující jméno

Záznamy jsou v zóně seřazeny abecedně. Pokud se klient shání po adrese počítače se jménem *cecilie.kdesi.cz* a jako odpověď dostane NSEC záznam pro *babicka.kdesi.cz* sdělující, že dalším jménem v doméně je *dedecek.kdesi.cz*, má jistotu, že *cecilie* neexistuje.

Jedním z problémů DNSSEC je velikost dat. Podepsáním zónového souboru jeho délka vzroste na pěti- až desetinásobek. Přidaných záznamů je hodně – ke každému jménu jeden NSEC a ke každému typu jeho záznamů jeden RRSIG, navíc klíče a jejich otisky – a jsou velké. Z prostého záznamu

```
koule.kdesi.cz.  IN      A~147.230.16.37
```

vznikne podepsáním

```
koule.kdesi.cz. 86400 IN A~147.230.16.37
        86400 RRSIG A~1 3 86400 20050623165042 (
              20050524165042 52042 kdesi.cz.
              LCS2UeIbo4UGOOZTqhIMRFYP176hiN734XaL
              KfwgjdJHCIMONepUXAS3p0yv1KS+WlrDvg46
              4ItFeMhRL2B5uA== )

        86400 NSEC www.kdesi.cz. A~RRSIG NSEC
        86400 RRSIG NSEC 1 3 86400 20050623165042 (
              20050524165042 52042 kdesi.cz.
              aceY2Od7ZUThQRETueGXa8PQvB7RlYET5Kxw
              /UUr6cdQQtK/vOFps6RCke9JP6e1Rdcmbnkq
              sRv2MITgO1dQVQ== )
```

Navíc je třeba, aby DNSSEC pokud možno rychle proniklo do nejvyšších pater doménové hierarchie. Jak jsem uvedl výše, pokud je v řetězci důvěry mezera (nepodepsaná doména), potřebuje klient znát důvěryhodné veřejné klíče z nižších pater stromu, což celému mechanismu silně ubírá na atraktivitě. Bohužel právě domény nejvyšší úrovně jsou velké a problémy s velikostí dat se u nich objevují nejpalčivěji (podepsaná doména *com* zabírá kolem 10 GB).

Pokud se domény *cz* týče, DNSSEC se v ní hned tak nedočkáme. Není reálné očekávat jeho doplnění do stávajícího systému. Naopak je poměrně pravděpodobné, že bude zařazeno do nového registračního systému, který by měl být nasazen v roce 2007.

# 5 Shrnutí

Při pohledu na klopotnou cestu vývoje výše popsaných rozšíření a především na jejich velmi pomalé prosazování do reálného života je jasně patrná obtížnost, s níž lze změnit velký distribuovaný systém, jakým DNS je. Vzhledem k obrovskému počtu existujících DNS serverů představuje DNS velmi konzervativní svět, u nějž je každá výraznější změna extrémně náročná.

Většina návrhů na změny se proto týká obsahu (nových typů záznamů), aniž by zasahovala do komunikačního protokolu. Nejopatrnější je pak IDN, které veškeré změny vysouvá do klientů a na straně serverů nemění vůbec nic. Jen přibudou nové domény. I to je možná příčinou, proč se jeho praktické nasazení rozebíhá zdaleka nejrychleji.

# Literatura

[1] Mockapetris, P.: *Domain names – concepts and facilities.* RFC 1034, IETF, 1987.

[2] Thomson, S., Huitema, C.: *DNS Extensions to support IP version 6.* RFC 1886, IETF, 1995.

[3] Huitema, C., Crawford, M.: *DNS Extensions to Support IPv6 Address Aggregation and Renumbering.* RFC 2874, IETF, 2000.

[4] Faltstrom, P.: *E.164 number and DNS.* RFC 2916, IETF, 2000.

[5] Bush, R.: *Delegation of IP6.ARPA.* RFC 3152, IETF, 2001.

[6] Bush, R., Durand, A., Fink, B., Gudmundsson, O., Hain, T.: *Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS).* RFC 3363, IETF, 2002.

[7] Mealling, W.: *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database.* RFC 3403, IETF, 2002.

[8] Faltstrom, P., Hoffman, P., Costello, A.: *Internationalizing Domain Names in Applications (IDNA).* RFC 3490, IETF, 2003.

[9] Hoffman, P., Blanchet, M.: *Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN).* RFC 3491, IETF, 2003.

[10] Costello, A.: *Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA).* RFC 3492, IETF, 2003.

[11] Thomson, S., Huitema, C., Ksinant, V., Souissi, M.: *DNS Extensions to Support IP Version 6.* RFC 3596, IETF, 2003.

[12] Faltstrom, P., Mealling, M.: *The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM).* RFC 3761, IETF, 2004.

[13] Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: *DNS Security Introduction and Requirements.* RFC 4033, IETF, 2005.

[14] Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: *Resource Records for the DNS Security Extensions.* RFC 4034, IETF, 2005.

[15] Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: *Protocol Modifications for the DNS Security Extensions.* RFC 4035, IETF, 2005.

[16] Zakon, R.: *Hobbes' Internet Timeline.*
http://www.zakon.org/robert/internet/timeline/

# OpenSource nástroje pro správu DNS a DHCP

**Aleš Padrta**

E-mail: apadrta@civ.zcu.cz

**Klíčová slova:** DNS, DHCP, OpenSource, Sauron, Delegovaná správa

**Abstrakt**

*Článek se zabývá OpenSource nástroji pro jednotnou správu DNS a DHCP konfigurací. Nejprve jsou představeny výhody společné správy obou konfigurací a dělení pravomocí mezi více uživatelů. Následuje přehled dostupných OpenSource systémů a jejich stručné charakteristiky. Dále je pozornost věnována konkrétnímu produktu – systému Sauron. Po stručném úvodu do filozofie tohoto systému je kladen důraz na praktické aspekty používání – od importu stávající konfigurace DNS a DHCP, přes aktualizaci záznamů a správu uživatelů, až po generování výsledných konfiguračních souborů.*

**Abstract**

*This paper is concentrated on OpenSource tools for the unified managment of the DNS and DHCP configurations. The advantages of the unified managment of the configurations and the advantages of a delegation of the competences are introduced. Then, the list of the available OpenSource tools and their brief description follow. Next, a specific tool – system Sauron – is described in detail. The practical aspects of the usage – the import of the current configuration, the managing hosts and users and the creating of the configuration files – are presented after a brief introduction to the philosophy of Sauron.*

## 1 Úvod

Každá větší organizace využívající výpočetní techniku se neobejde bez vlastních služeb DHCP a DNS. Se zvyšujícím se počtem strojů rostou také nároky na jejich správu. Plně manuální správa se ukazuje jako neefektivní, proto jsou vytvářeny odpovídající nástroje. Tento článek se zabývá takovýmito nástroji z rodiny OpenSource produktů.

Nejprve jsou v kapitole 2 představeny výhody společné správy DNS a DHCP, v kapitole 3 následuje přehled nejpoužívanějších OpenSource systémů. Výběrem optimálního systému se zaobírá kapitola 4 a jeho podrobný popis je k nalezení v kapitole 5.

# 2 Proč společná správa?

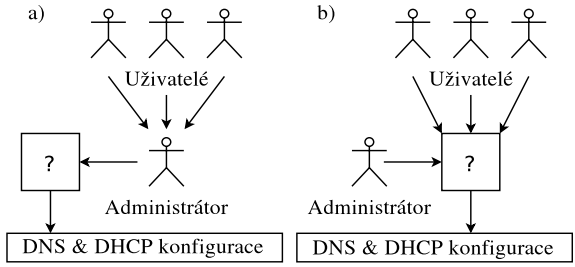
Než se dostaneme k samotným systémům pro společnou správu DNS a DHCP, bude dobré si připomenout jak spolu obě služby souvisejí a ujasnit si, co vlastně očekáváme od systému, který je bude spravovat.

Není jistě tajemstvím, že DHCP přiděluje, kromě dalších nastavení, určité hardwarové adrese odpovídající IP adresu. Pomocí DNS lze pak tuto IP adresu převést na doménové jméno a opačně. Každému běžnému stroji (zpravidla se o nich také hovoří jako o hostech) tak připadá trojice údajů hardwarová adresa – IP adresa – doménové jméno, které jsou však po dvojicích rozděleny do dvou systémů. Při změně IP adresy je třeba modifikovat konfigurace jak DNS a DHCP. Je-li tato činnost prováděna ručně, může docházet k rozsynchronizování údajů v DNS a DHCP. Nehledě na možnost různých duplicitních či nekonzistentních záznamů.

Z předchozího odstavce tedy vyplývají dva základní požadavky na systém pro správu DNS a DHCP. Musí být zajištěna

- synchronizace mezi DNS a DHCP konfiguracemi a
- konzistentní stav dat, zejména jde o duplicitní IP či hardwarové adresy.

Když už budou data uložena v centrální databázi, je velmi snadné přiřadit přístupová práva k jednotlivým záznamům a lze začít uvažovat také o možnosti rozložit správu mezi více uživatelů. Přínos delegované správy je ve zefektivnění práce – každý uživatel má na starost pouze část záznamů, o kterých je schopen udržovat si snadno přehled. Dále se také urychlí registrace či změny v registraci jednotlivých hostů, jelikož není nutné žádat centrální autoritu. Centrální a delegovaný systém správy DNS a DHCP je ilustrován na obrázku 1.




Obr. 1 a) Centrální správa b) Delegovaná správa

Z předchozího odstavce tedy vykrystalizovaly další požadavky na vybíraný systém. Kromě dříve specifikovaných požadavků musí být ještě schopen

- delegovat přístupová práva jednotlivým uživatelům a
- umožnit přehlednou správu svěřených strojů.

# 3 Přehled OpenSource systémů

V této kapitole jsou stručně představeny charakteristiky tří běžně používaných OpenSource systémů pro správu DNS a DHCP, které splňují požadavky specifikované v předchozí kapitole.

## 3.1 OSL Maintain

OpenSource systém *OSL Maintain* [1] byl vyvinut v roce 2001 na Oregon State University jako náhrada řady skriptů pro editaci konfigurační souborů DNS a DHCP. Celý systém byl navržen jako modulární, momentálně jsou však k dispozici pouze dva moduly umožňující registraci strojů a jejich převody mezi doménami. Jejich funkce v dostatečné míře pokrývají specifikované požadavky, tj. přes webové rozhraní umožňují autentizovaný přístup více uživatelů, kterým je možno přiřadit odpovídající přístupová práva. Dále pak umožňuje vyhledávání záznamů podle doménového jména, hardwarové adresy a IP adresy.

Co se týká programové realizace, *OSL Maintain* je založen na *PHP* s podporou databáze *My-SQL* a webového serveru *Apache*. Nové vlastnosti systému mohou být dodávány pomocí dalších modulů. Generovanou DNS konfiguraci je doporučeno využívat *TinyDNS*, nicméně by měla být kompatibilní i s *ICS BIND server*. Konfigurace pro DHCP předpokládá používání *ISC DHCP server*.

## 3.2 Netreg

*NetReg* [2] je dalším OpenSource systémem pro komplexní správu DNS a DHCP. Byl vytvořen v letech 2000–2005 na Carnegie Mellon University pro potřeby sjednocené správy sítě. Opět umožňuje přístup více uživatelů se specifikovanými právy přes webové rozhraní.

Celý systém je založen na skriptech v *PERL*u s podporou databáze *My-SQL* a webového serveru *Apache*. Generovaná konfigurace DHCP je navržena pro *ISC DHCP server v3.0+* a konfigurační soubor pro DNS by měl být využíván *ISC BIND server v8* nebo *v9*, přičemž je verze 9 výrazně preferována.

## 3.3 Sauron

Posledním z často používaných OpenSource systémů je *Sauron* [3], vyvíjený od roku 2001 ve výpočetním středisku University of Jyväskylä. Stejně jako předchozí systémy nabízí webové rozhraní pro více uživatelů s možností regulace přístupových práv a vyhledávání záznamů podle různých kritérií. Dále pak umožňuje administrátorovi alternativní správu pomocí příkazové řádky.

Sauron ke svému běhu potřebuje *PERL*, podporu databáze *PostgreSQL* a webový server. Výstupní konfigurační soubory jsou navrženy pro *ISC BIND server v8* a *ISC DHCP server 3.0*.

# 4 Výběr systému

Představené systémy mají mnoho společného a liší se pouze v některých parametrech. Při výběru vhodného systému pro Západočeskou univerzitu v Plzni bylo v první řadě nutné vzít v potaz zejména kompatibilitu s již provozovanými DNS a DHCP servery, jelikož při zavádění nového systému byly žádoucí minimální změny a měly se týkat pouze vytváření konfiguračních souborů, vše ostatní mělo zůstat ve stejném stavu. V našem případě z hlediska kompatibility vede systém *Sauron*, protože je schopen generovat konfigurace pro *ISC BIND server v8* a *ISC DCHP server 3.0*.

Dále byla posuzována uživatelská přívětivost webového rozhraní a možnosti vyhledávání záznamů, jelikož se systémem budou výhledově pracovat desítky uživatelů. Dle subjektivního hodnocení nejlépe vyhovuje systém *Sauron*, za milé pozitivum lze považovat zejména zahrnutí regulárních výrazů do vyhledávacího formuláře.

V neposlední řadě je třeba také zmínit aktuálnost jednotlivých projektů a výhledy na jejich vývoj do budoucna. Ohledně tohoto kritéria lze říci, že systémy *NetReg* a *OSL Maintain* jsou považovány téměř za hotové a opravy chyb jsou zveřejňovány s minimální frekvencí – lze předpokládat, že se tak děje díky absenci chyb a nikoliv neochotě cokoliv opravovat. Systém *Maintain* může být v budoucnu přidáním dalších modulů dále rozšiřován. Naproti tomu se systém *Sauron*, ačkoliv je již praktický nasazen na řadě pracovišť, nachází stále ve vývojové fázi.

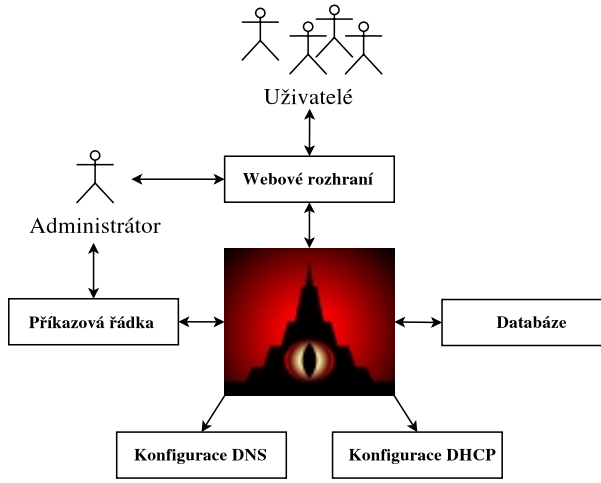
Na základě výše diskutovaných faktů byl vybrán systém *Sauron*, protože generovaná konfigurace plně odpovídá používaným DNS a DHCP serverům a nabízí nejpohodlnější webové prostředí. Skutečnost, že je ještě ve vývojové fázi může být spíše výhodou, protože základní funkce systému je prověřena praktickým používáním a systém může relativně pružně reagovat na nově se vyskytující požadavky.

# 5 Sauron – co to je a proč ho milujeme

V této kapitole se již budeme věnovat detailnímu popisu systému *Sauron*, od celkové koncepce systému přes jeho instalaci až po praktické aspekty používání.

## 5.1 Základní idea systému

O výhodách zavedení delegované správy pomocí více uživatelů jsme již hovořili v úvodu, vzpomeňme na obrázek 1, *Sauron* je typickým představitelem systému pro delegovanou správu. Přistupme nyní blíže k Sauronovi a podívejme se na jeho vztahy s okolím. Celá situace je zachycena na obrázku 2.




Obr. 2 Vztahy Saurona s okolím

Jak to tedy funguje? Sauron ve své databázi uchovává záznamy o hostech a na požádání administrátora je schopen podle nich vygenerovat konfigurační soubory pro DNS a DHCP. Jednotliví uživatelé přes webové rozhraní přistupují do databáze a mění si údaje o hostech, kteří spadají pod jejich správu. Administrátor systému má k dispozici ještě příkazovou řádku a sadu skriptů zpřístupňující další funkce, které nejsou nabízeny ve webovém rozhraní.

Kromě webového rozhraní pro editaci, kde je každý uživatel ověřován heslem, existuje ještě volitelné webové rozhraní určené pouze k prohlížení údajů o hostech, zde jsou informace dostupné pro všechny bez nutnosti přihlašování. Zdá se, že z praktického hlediska není tato komponenta, označovaná jako *DNS prohlížeč*, příliš využívána.

Systém *Sauron* umožňuje spravovat více oddělených konfigurací, tj. v jednom systému může být spravováno několik dvojic DNS a DHCP serverů. V celém systému *Sauron* se pod pojmem *server* rozumí konfigurace pro související DNS a DHCP server. Data každého serveru jsou dále rozdělena do *zón*, a to ve stejném významu jako v DNS konfiguraci. Každý záznam tedy patří právě do jedné zóny a jednoho serveru.

## 5.2 Instalace systému

Předpokládejme, že se vám Sauron zamlouvá a chcete si jej nainstalovat. Podrobný postup jak to provést v systému *Debian GNU/Linux* je náplní několika následujících odstavců.

Ještě než se pustíme do instalace je vhodné zmínit se o výběru stroje na kterém Sauron poběží. Vzhledem k přístupu řady uživatelů přes webové rozhraní je z bezpečnostního hlediska vhodné, aby na stejném stroji neběžely žádné důležité služby. Není tedy žádoucí, aby Sauron byl instalován na stroj kde je situován primární DNS server či DHCP server. Podrobněji se dalším výhodám tohoto přístupu budeme věnovat později v kapitole 5.6 pojednávající o generování konfigurace. Spokojme se zatím s konstatováním, že systém Sauron je třeba nainstalovat na stroj s méně důležitými službami.

### 5.2.1 Požadovaný SW

Nejprve je nutné se přesvědčit, zda-li jsou v systému nainstalovány programy nutné pro chod vlastního systému *Sauron*. Jedná se o následující softwarové produkty:

- *PostgreSQL*
  - požadována je verze 7.x nebo novější
  - v APT balíček `postgresql`
- *PERL*
  - požadována je verze 5.x nebo novější
  - v APT balíček `perl`
  - musí být nainstalováno několik specifických modulů
    - *CGI* – v APT balíček `libcgi-perl`
    - *Digest::MD5* – u vyšších verzí *PERL*u (např. 5.8) je standardně instalován
    - *Net::DNS* – v APT balíček `libnet-dns-perl`
    - *Net::Netmask* – v APT balíček `libnet-netmask-perl`
    - *Pg* – v APT balíček `libpg-perl`
    - *DBD::Pg* – v APT balíček `libdbdpg-perl`, požadováno pro *Sauron* od verze 0.7.x
    - *Crypt::RC5* – není ve standardních balíčcích, nutno stáhnout z *CPAN* [4], požadováno pro *Sauron* od verze 0.7.x
- *WWW server – apache2*
  - v APT balíček `apache2`

### 5.2.2 Systém Sauron

Nyní můžeme přistoupit k instalaci samotného systému *Sauron*. Na příslušných webových stránkách [3] jsou k dispozici ke stažení již důkladně otestované verze (stable) a také nejnovější vydání (testing). Na diskuzním fóru uživatelů *Saurona* jsou k dohledání kladná hodnocení testovacích verzí a tak se nejnovější verze jevila jako ideální volba. V době, kdy bylo na Západočeské univerzitě v Plzni přistoupeno k rozhodnutí přejít na systém *Sauron*, byla k dispozici verze 0.7.2.

Předpokládejme nyní, že instalační soubory poslední verze jsou staženy na vybraný stroj a jsou rozbaleny v nějakém dočasném adresáři. Pro instalaci souborů je třeba jako uživatel `root` postupně spustit příkazy

```
./configure
make
make docs
make install
```

Pro udržení přehlednosti v systému je lepší zkopírovat soubory `sauron.cgi` a `browser.cgi` do standardního adresáře pro *cgi* skripty, tedy z původního adresáře `/usr/local/sauron/cgi` do adresáře `/usr/lib/cgi-bin/sauron`. Alternativně lze také v konfiguraci webového prohlížeče přesměrovat *cgi* adresář do původního adresáře. Nyní se tedy systém Sauron nalézá v těchto adresářích:

- `/usr/local/etc/sauron` – konfigurační soubory,
- `/usr/local/sauron` – vlastní systém Sauron,
- `/usr/lib/cgi-bin/sauron` – *cgi* skripty.

Přistupme nyní k editaci souboru `/usr/local/etc/sauron/config`, ve němž je systém *Sauron* konfigurován. Konfigurační soubor je ve formě perlovského skriptu a jednotlivé volby jsou ve formě `proměnná=hodnota;`. Jelikož konfigurační soubor obsahuje také heslo k databázi, je nutné omezit přístup k souboru, nejlépe pouze uživateli `www-data`. Jednotlivá nastavení můžeme rozdělit na dvě části – povinná a volitelná. Věnujme se nejprve povinné části, která musí být pro chod systému správně nastavena.

#### Obecná nastavení

- `$PROG_DIR = "/usr/local/sauron/";` – cesta k souborům systému *Sauron*, implicitně nastavena
- `$LOG_DIR = "$PROG_DIR/logs/";` – cesta k logovacím souborům, implicitně nastavena, je nutné přidat práva zápisu uživateli `www-data` do tohoto adresáře
- `$SERVER_ID = "mordor.civ.zcu.cz";` – jméno serveru pro www-server

**Přístup k databázi**

- `$DB_DSN = "dbi:Pg:dbname=sauron";` – jméno databáze v *PostgreSQL*, oblíbené jméno je `sauron`
- `$DB_USER = "sauron";` – jméno uživatele, který bude přistupovat do specifikované databáze, vhodné jméno je opět `sauron`
- `$DB_PASSWORD = "sjb.4a";` – heslo výše specifikovaného uživatele

Ve volitelné části je možné částečně přizpůsobit chování systému *Sauron* k obrazu svému. Jednotlivé volby jsou v konfiguračním souboru podrobně komentovány. Je vhodné se jimi zabývat až po bližším seznámením se *Sauronem*, tedy až bude možno si představit co vlastně ovlivňují. Na hlavní funkci systému *Sauron* však nemají vliv.

Konfigurace *DNS prohlížeče* je umístěna v adresáři `/usr/local/etc/sauron`, v souboru `config-browser`. Obsahuje proměnné `$PROG_DIR`, `$DB_DSN`, `$DB_USER` a `$DB_PASSWORD`, jejichž význam je stejný jako v konfiguračním souboru systému *Sauron*. Opět je třeba zakázat právo čtení všem uživatelům kromě `www-data`. Nastavení zón je prováděno následovně:

- `$BROWSER_CONF {zcu=>[‘zcu‘,‘zcu.cz‘], ...}`
  `\texttt{..., pef=>[‘zcu‘,‘pef.zcu.cz‘]};`
  - definice zón, které mají být přístupné
  - formát zápisu zony je patrný z příkladu:
    - `<odkaz>=>[‘<jméno-serveru>‘,‘<jméno-zóny>‘]`
    - interpretace je následující: údaje ze serveru `<jméno-serveru>` a jeho zóny `<jméno-zóny>` budou dostupné na adrese `http://host/cgi-bin/browser.cgi/<odkaz>`
    - jednotlivé zóny jsou od sebe odděleny čárkou
- `$BROWSER_HELP = zcu.cz=>[‘Info‘,‘/manual-wb/wb-help.html‘];`
  - definice odkazu pro nápovědu (není-li pro danou zónu odkaz specifikován, na stránkách se ani nezobrazí)
  - formát zápisu je patrný z příkladu
    `<jméno-zóny>=>[‘<Popisek>‘,‘<relativní/absolutní odkaz>‘]`

Další možnosti konfigurace jsou opět volitelné a dostatečně komentované v konfiguračním souboru. Na primární funkci *DNS prohlížeče* nemají vliv a opět je doporučeno se jimi zabývat až po bližším seznámení se s *DNS prohlížečem*.

### 5.2.3 PostgreSQL

Systém *Sauron* i příslušný *DNS prohlížeč* ke své funkci vyžadují databázi *PostgreSQL*. Ta je již nainstalována, nicméně je ještě třeba vytvořit odpovídající databázi, uživatele a nastavit přístupová práva. Je třeba se přihlásit jako uživatel `postgres` a

- vytvořit uživatele `sauron` příkazem
  `createdb sauron`
- vytvořit databázi `sauron` příkazem
  `createuser --pwprompt --no-createdb --no-adduser sauron`.
  Heslo uživatele musí být stejné jako v konfiguračním souboru *Saurona*.

Dále bude nutné jako uživatel `root` povolit přístup uživateli `sauron` k databázi `sauron`, což se provede v konfiguračním souboru `pg_hba.conf` v adresáři `/etc/postgresql` přidáním řádek

- `local sauron sauron password`
- `host sauron sauron 127.0.0.1 255.255.255.255 password`

před řádek `# All other connections by UNIX sockets`. Na pořadí jednotlivých záznamům záleží, jelikož nejprve jsou definovány méně obecné požadavky. Aby se změny projevily, je potřeba restartovat databázi, což se provede příkazem `/etc/init.d/postgresql restart`.

Dalším krokem je vytvoření prázdných tabulek v databázi `sauron`. V adresáři `/usr/local/sauron` je potřeba spustit skript `./createtables`. O správném vytvoření tabulek se lze přesvědčit spuštěním skriptu `./status`, v následném výpisu bude momentálně hlášení, že se v databázi nenacházejí žádné servery, což je v pořádku, vše bude napraveno při importu aktuální konfigurace. Databázové tabulky jsou tedy a připraveny a je na čase do nich vložit data nezávislá na používané konfiguraci DNS a DHCP.

Prvním krokem je načtení seznamu DNS root-serverů, stále v adresáři *Saurona*, příkazem `./import-roots default named.root`, přičemž aktuální verze souboru `named.root` je ke stažení na stránkách *InterNIC* [5].

Dále pak *Sauron* umožňuje automatické zjišťování výrobce síťové karty. Pro aktivaci této služby je nutné, příkazem `./import-ethers oui.txt`, importovat seznam výrobců a odpovídajících částí hardwarových adres. Aktuální seznam je dostupný na stránkách *IEEE OUI* [6].

### 5.2.4 WWW server – Apache 2

Posledním krokem při instalaci je konfigurace webového serveru. Veškerá nastavení jsou v souboru `httpd.conf` v adresáři `/etc/apache2`. Podrobný popis

konfigurace WWW serveru není náplní tohototo článku, takže v následujícím výpisu z konfiguračního souboru jsou uvedeny pouze důležité části.

```
<VirtualHost mordor.civ.zcu.cz>
        ...
        DocumentRoot /var/www/sauron
        ...
        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/sauron/
        <Directory "/usr/lib/cgi-bin/sauron">
                AllowOverride None
                Options FollowSymLinks ExecCGI
                Order allow,deny
                Allow from all
        </Directory>
        ...
</VirtualHost>
```

Uvedená konfigurace předpokládá, že v adresáři `/var/www/sauron/` se nachází úvodní stránka `index.html` na níž je umístěn odkaz na skript `sauron.cgi` ve tvaru `<a href="cgi-bin/sauron.cgi">`. Řádka `ScriptAlias` definuje adresář obsahující *cgi* skripty. V tomto adresáři musí být z pochopitelných důvodu povoleno spouštění *cgi* skriptů. Nakonec je třeba se ujistit, zda uživatel `www-data` má přístupová práva ke čtení a spouštění *cgi* skriptů `sauron.cgi` a `browser.cgi`.

Nyní by měl být celý systém připraven k použití. Je-li vše v pořádku, na příslušné webové stránce bude vidět přihlašovací formulář. Zatím však není definován žádný uživatel, takže přihlášení zatím není možné.

## 5.3 Správa uživatelů

Veškerá správa uživatelů je prováděna z příkazové řádky bez autentizace, proto je vhodné povolit spouštění skriptů v adresáři `/usr/local/sauron/` pouze správci celého systému *Sauron*. Všechny skripty zmiňované dále jsou umístěny v uvedeném adresáři.

Uživatelé systému *Sauron* mohou být řazeni do skupin, jejichž přístupová práva uživatel zdědí. Kolidují-li práva nastavená ve skupině a práva uživatele patřícího do této skupiny, pak práva uživatele mají vyšší prioritu. Každý uživatel nebo skupina může být členem více skupin. Skupiny jsou vytvářeny příkazem `./addgroup`, interaktivně je zadán název skupiny a její popis. Nastavení jednotlivých přístupových práv a dalších nastavení skupiny je prováděno příkazem `./modgroup`. Přístupová práva mají tyto podoby

- server – právo ke čtení (R), zápisu (W) nebo k rozšířenému zápisu (X)
- zóna – práva RWX

- síť – práva RWX
- hostname – regulární výraz definující povolené záznamy
- IP-maska – výraz definující rozsah IP adres, které smí uživatel zadávat
- maska skupiny – regulární výraz definující povolené skupiny DHCP

Existují i další omezení, která však nejsou důležitá. Z nastavení skupin stojí za zmínku možnost definovat si povinně vyplňované údaje ve formuláři. Při definování podmínek je testována existence serveru, zóny, takže definovat přístupová práva je nutné pro existující konfiguraci. Skupinu je možné opět zrušit příkazem `./delgroup`

Systém *Sauron* principiálně rozlišuje dva typy uživatelů – administrátor, kterému je implicitně zpřístupněno vše, a běžný uživatel, jež implicitně nemá žádná přístupová práva vyjma práv pro nastavení svého účtu. Pro přidání, změnu a smazání uživatele existují obdobné příkazy jako pro skupiny – `./adduser`, `./moduser` a `./deluser`.

## 5.4 Práce v systému Sauron

V této kapitole je krok za krokem představeno vytváření konfigurace pomocí systému *Sauron* a také řešení požadavků, se kterými se lze při běžném provozu setkat. Postupem, kterak do systému dostat stávající konfiguraci, se zabývá až kapitola 5.5, protože je vhodné nejprve nahlédnout jak systém *Sauron* pracuje se záznamy o hostech a jejich konfigurací.

### 5.4.1 Popis prostředí

Většina globálních nastavení je prováděna ve webovém rozhraní. Při přihlášení do systému je možné specifikovat zda stránka má podporovat rámce, či nikoliv. Verze bez rámců je doporučována, protože jsou korektně zobrazeny všechny údaje i pro nízké rozlišení monitoru. Po celou dobu přihlášení je v horní části zobrazena hlavní nabídka definující skupiny možností, po výběru některé z nich se v levé části zobrazí příslušné možnosti. V hlavním okně je pak zobrazen aktuální dialog.

Pro pohodlnou tvorbu základní konfigurace jsou, kromě webového rozhraní, používány také skripty spouštěné z příkazové řádky. Je dobré si ještě jednou připomenout, že skripty jsou pouštěny bez autentizace uživatele, a je tedy nutné povolit přístup pouze administrátorovi systému *Sauron*.

### 5.4.2 Nastavení uživatelského konta

Změny nastavení uživatelského konta jsou prováděny ve skupině možností `Login`. Všichni uživatelé tu mají možnost prohlédnout si jaká mají přístupová práva, změnit si heslo a e-mailovou adresu, zjistit kdo je aktuálně přihlášen do systému a zapnout či vypnout používání rámců. Další užitečnou vlastností je možnost nastavení implicitního serveru a zóny, která se automaticky aktivuje po přihlášení do sytému.

Administrátoři mají zpřístupněny ještě další možnosti – přehled posledních 40 přihlášených uživatelů a tzv. *Session browser*, jenž umožní zjistit co který uživatel prováděl při svém přihlášení do systému. Dále je možné vytvářet zprávy, které se zobrazí všem uživatelům během přihlašovací procedury.

### 5.4.3 Správa serveru

Veškerá správa serverů je skryta ve skupině nastavení `Servers`. Administrátor zde může přidat nový server, vybrat aktuální server, změnit nastavení stávajícího serveru a také kompletně zrušit celý server.

Při vytváření nového serveru je třeba vyplnit tyto údaje:

- `Name` – jméno serveru (např. `zcu-server`), smí obsahovat pouze alfanumerické znaky, podtržítko a pomlčku
- `Hostname` – doménové jméno serveru (např. `eros.zcu.cz`)
- `IP address` – IP adresa DHCP serveru, je využíváno pouze pro aktivovaný `failover`
- `Hostmaster` – údaj pro SOA záznam o DNS zóně
- `Configuration directory` – adresář definující, kde jsou zónové soubory DNS, nutno zadat včetně koncového lomítka
- `Slave for` – nastavení zda je server `slave` jiného serveru, používáno pro `failover`

Editací lze do nastavení serveru doplnit další údaje, konkrétně se jedná o

- implicitní hodnoty pro podřízené DNS zóny – bloky `allow-transfer`, `allow-query`, `allow-recursion`, `logging`, a další uživatelsky definované nastavení
- nastavení globálních parametrů DHCP
- aktivaci `failover`u a jeho nastavení

V rámci popisu serveru je vhodné se zmínit podrobněji o aktivaci `failoveru`. Jelikož se jedná o dva servery, jeden primární a jeden sekundární, je nutné v systému *Sauron* vytvořit dva odpovídající servery. Položka `Slave for` není pro primární server vyplněna a nastavení `Enable failover protocol` je zapnuto. V tomto serveru je možné spravovat hosty a měnit konfiguraci. Sekundární server má v položce `Slave for` nastaven primární server. V sekundárním serveru nelze měnit konfiguraci, neboť je převzata z primárního serveru.

### 5.4.4 Vytvoření zón

Každý server může obsahovat libovolný počet DNS zón a pro jejich správu slouží skupina možností `Zones`. Opět lze zóny přidávat, měnit a rušit. Kromě toho je možné automaticky přidat standardní zóny (tj. zóny související s `localhost`). K dispozici je také seznam o hostů, kteří jsou již zavedeni v systému, ale odpovídající nastavení ještě nebylo vypropagováno na příslušné servery (možnost `Show pending hosts`).

U každé nově přidávané zóny je potřeba vyplnit její název, např. `zcu.cz` či `228.147.in-addr.arpa`, nastavit zda se jedná o `master` nebo `slave` zónu a určit je-li zóna reverzní. V dialogu je také možnost zvolit i jiné typy zón než `IN`, nicméně ve verzi 0.7.2 nejsou zatím použitelné. Každá vytvořená zóna zdědí předdefinovaná nastavení svého serveru, je-li potřeba nastavit každou zónu zvlášť, stačí vyplnit příslušnou kolonku v nastavení zóny a aplikuje se nové nastavení. Dále je možné specifikovat zónové `NS` a `MX` záznamy.

Při zadávání záznamů o hostech je třeba mít vybrán server a konkrétní zónu, oba tyto údaje jsou zobrazeny v levé dolní části obrazovky. Všechny záznamy jsou udržovány pouze v přímých zónách, záznamy pro reverzní zóny jsou vytvářeny automaticky při generování konfiguračního souboru pro DNS.

### 5.4.5 Nastavení sítí a podsítí

Globální parametry DHCP jsou zadávány v rámci serveru, zbývající nastavení je ukryto ve skupinách možností `Nets` a `Groups`.

Možnosti patřící do skupiny `Nets` definují topologii sítě. Blok, který je v konfiguračním souboru DHCP ohraničen tagem `shared-network`, je ve webovém rozhraní označován jako `VLAN`. Každý takový záznam je specifikováno pouze jménem. Všechny dále definované podsítě patří do nějaké `VLAN`, není-li specifikována, jsou implicitně zařazeny do `VLAN` s názvem `CHAOS`.

Webové rozhraní rozlišuje síť (`Net`) a podsíť (`Subnet`). Zatímco síť slouží pouze k definování svěřeného rozsahu, který má být promítnut do konfiguračního souboru, každé podsíti odpovídá v konfiguračním souboru blok `subnet`. Např. pro ZČU tedy existuje jediná síť `147.228.0.0/16`, podsítí je více.

Každá podsíť je specifikována svým jménem, rozsahem ve formě tzv. *CIDR* (Classless Inter-Domain Routing) a příslušností k `VLAN`. Dále může být připojen slovní popis a DHCP záznamy specifické pro danou podsíť. Pro snazší přidávání nových hostů ve webovém rozhraní je také možné nastavit rozsah automaticky přidělovaných pevných IP-adres.

Nabídka `Groups` umožňuje správu skupin v DHCP. Systém *Sauron* rozlišuje skupiny tří typů

- `DHCP class` – skupina ve stejném významu jako `group` v konfiguračním souboru, jednotliví hosté mohou být řazeni do těchto skupin.
- `Custom DHCP class` – skupina umožňující jemnější dělení přístupových práv pro jednotlivé dynamické pooly
- `Dynamic address pool` – každý záznam tohoto typu odpovídá jednomu dynamickému poolu

Každá skupina je definována svým jménem a typem. Dále může obsahovat DHCP záznamy specifické pro danou skupinu, případně daný dynamický pool.

Systém `Sauron` zachází s dynamickými pooly na první pohled velmi nestandardně. Každý dynamický pool je založen vytvořením skupiny typu `Dynamic address pool`. Rozsah poskytovaných IP adres je definován přiřazením hostů s příslušnými IP adresami do této skupiny. Zařazení hosté musí mít vyplněnou pouze IP adresu a doménové jméno, jelikož budou dynamicky přiřazeni k zatím neznámé hardwarové adrese.

Přidávání velkého množství hostů manuálně (viz kapitola 5.4.7) není příliš vhodné, proto je k dispozici skript `./generatehosts` umožňující automatické vytvoření potřebných záznamů. Stačí specifikovat počet generovaných záznamů, počáteční IP adresu, skupinu, do které mají hosté patřit, a šablonu jména. Do šablony je možné zahrnout index generovaného záznamu a jednotlivé části odpovídající IP adresy.

Pro řízení přístupu do dynamických poolů slouží skupiny typu `Custom DHCP class`. Do dynamického poolu s nastavením `deny unknown-clients` smějí pouze členové skupin, pro které má dynamický pool ve svých DHCP záznamech řádku

```
allow members of "<jméno skupiny typu Custom DHCP class>"
```

Každý host může být členem více takovýchto skupin, čímž lze snadno rozlišit do kterých dynamických poolů má mít daný host přístup.

### 5.4.6 Šablony

Správa šablon je ve skupině možností `Templates`. K dispozici jsou šablony pro `MX` záznamy, `WKS` záznamy, `PRN` záznamy a také záznamy typu `HINFO`. Jednotlivé šablony jsou pak dále používány při zadávání informací o hostech.

### 5.4.7 Záznamy o hostech

Správou hostů se, na rozdíl od vytváření globální konfigurace, budou zabývat všichni uživatelé, nejenom administrátoři. Pro práci s hosty je nutné mít vybraný server a zónu, poté je možné využívat možností nabízených ve skupině `hosts`.

Přidání záznamu o hostovi je prováděno pomocí možnosti `Add host`. V zobrazeném dialogu je potřeba vyplnit následující položky

- `Hostname` – doménové jméno, při generování konfigurace je za něj automaticky přidán název příslušné zóny, je hlídána duplicita
- `Subnet` – je-li nastavena nějaká podsíť, je automaticky přidělena IP adresa z definovaného rozsahu příslušné podsítě
- `IP` – nebyla-li nastavena žádná podsíť je nutné zadat přiřazovanou IP adresu, duplicita je hlídána
- `Router` – nenulové číslo definuje hosta jako router pro příslušnou podsíť, je-li v podsíti více takových hostů je vybrán host s nejvyšší prioritou
- `Group` – DHCP skupina, do které host patří
- `Subgroup` – podobně jako `Group`, vhodné pro nastavení přístupu do dynamických poolů (např. pro notebooky)
- `Ethernet address` – hardwarová adresa
- `Expiration date` – datum, do kdy je záznam platný, prázdná kolonka značí neomezenou platnost.

Povinně musí být vyplněno akorát doménové jméno a IP adresa (případně definován subnet pro automatické přiřazení). Dále lze u záznamu pomocí dříve vytvořených šablon vyplnit údaje týkající se `MX`, `WKS`, `HINFO hardware` a `HINFO software`. Pro lepší evidenci uživatelů odpovědných za příslušný stroj je k dispozici blok údajů `Host info`, tyto údaje nemají žádný vliv na konfiguraci.

Kromě přidání záznamů klasických hostů, lze pro danou zónu také přidávat `MX` záznamy, `NS` záznamy, `SRV` záznamy, `PRN` záznamy a `Glue` záznamy.

Zvláštní pozornost je třeba věnovat *alias*ům, jednak jsou k dispozici tzv. volné *aliasy* (možnost `Add alias` ve skupině `Hosts`), kdy je definován libovolný *alias* k libovolnému doménovému jménu. Druhou možností je vyhledat záznam o hostovi a pomocí tlačítka `Alias` přidat *alias* přímo ke konkrétnímu záznamu.

Dále je možné přidat tzv. *rezervaci*, kdy je zablokována určitá IP adresa a doménové jméno, ale záznam není propagován do DHCP. Každý existující záznam lze snadno přepnout na rezervaci a případně opět zpátky na plnohodnotný záznam (tlačítka `Enable` a `Disable`).

Aby bylo možné záznam smazat nebo editovat je nutné jej nejprve vyhledat, k tomuto účelu jsou k dispozici položky

- `Search` – otevře vyhledávací formulář s posledním nastavením
- `Last Search` – zobrazí výsledky pro poslední nastavení
- `New Search` – otevře prázdný vyhledávací formulář

Při vyhledávání je možné filtrovat výsledky podle typu záznamu (host, alias, *MX* záznam, ...), příslušnosti ke skupině, podsítě, CIDR, regulárního výrazu specifikujícího doménové jméno, data vytvoření či poslední modifikace, hardwarové adresy a dalších nepodstatných údajů. Vyhovuje-li podmínkám právě jeden záznam, je přímo zobrazen, jinak se objeví seznam všech záznamů splňující dané podmínky.

## 5.5 Import dat

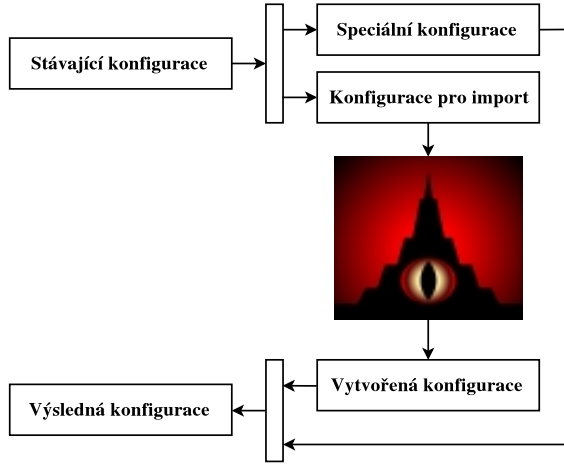
Načtení stávající konfigurace potká téměř každého, neboť valná většina se po systému pro správu DNS a DHCP začne shánět až ve chvíli, kdy začne být spravování konfiguračních souborů nepřehledné. V systému *Sauron* není tato aktivita příliš propracována, naštěstí se však odehraje pouze jednou. Import dat sestává ze dvou fází, nejprve je načtena konfigurace DNS a poté doplněna údaji z DHCP.

### 5.5.1 Import konfigurace DNS

Pro import konfigurace DNS slouží příkaz `./import`, jehož parametry jsou jméno vytvářeného serveru a relativní či absolutní cesta k souboru `named.conf`. Nejsou-li soubory, na něž se hlavní konfigurační soubor odkazuje, na specifikovaném místě, lze určit náhradní adresář, ve kterém budou očekávány.

Jak již bylo naznačováno, importovací skript není dokonalý a s některými parametry si nedokáže poradit. Týká se to zejména bloků `options` a `logging`. Oba bloky lze po vlastním importu ručně dopsat přes webové rozhraní, což při relativně nízkém počtu řádek nevadí. Další problém je s příkazem `$GENERATE`, který je zpravidla používán pro pojmenování dynamicky přidělovaných IP adres. Pro poloautomatickou nápravu je k dispozici skript `./generatehosts`, ale je vhodné jej použít až při importu konfigurace DHCP, protože ovlivňuje také dynamické pooly.

Na některých pracovištích mohou být používány i jiné typy zón než `IN`, např. `CHAOS`, `HESIOD` či `IPv6`, se kterými si systém *Sauron* neumí v současné verzi vůbec poradit. Řešení spočívá v oddělení zón nepodporovaného typu ještě před importem a jejich následného přidání do každé vygenerované konfigurace. Idea je ztvárněna na obrázku 3.




Obr. 3 Zahrnutí zón nepodporovaných typů

Generování konfigurace a její propagaci je bližší pozornost věnována v kapitole 5.6.

Výsledkem importu DNS konfigurace je vytvoření serveru a příslušných zón, do kterých jsou přiřazeny záznamy obsahující IP adresu a doménové jméno. Každé importování DNS vždy vytváří nový server, takže je možné snadno načíst více konfigurací.

### 5.5.2 Import konfigurace DHCP

Konfigurační soubor může být načten, až když jsou importována data z DNS. V podstatě se jedná o vytvoření sítí, doplnění hardwarové adresy k odpovídající IP adrese a přidání dalších nastavení distribuovaných DHCP serverem.

Skript `./import-dhcp` zajišťující načtení konfigurace DHCP je velmi primitivní a očekává jednoduchý konfigurační soubor. Je tedy nutné stávající konfiguraci upravit tak, aby splňovala následující podmínky:

- definice podsítí (`subnet`) v jednotlivých sítích musí být uzavřeny v tagu `shared-network`
- skupiny (`group`) nesmějí být vzájemně vnořovány
- položka `fixed-address` v záznamech o hostech musí obsahovat IP adresu nikoliv doménové jméno

Výše uvedené podmínky jsou nutné, aby se konfigurační soubor vůbec zpracoval. Některé informace obsažené v konfiguraci DHCP jsou však ignorovány. Předpokládejme, že se konfiguračního soubor podařilo načíst a je možné se věnovat doplnění ignorovaných údajů.

Nejprve je nutné přidat doplňková nastavení u jednotlivých hostů, tj. všechno vyjma tagů `fixed-address` a `hardware ethernet`. K tomuto účelu poslouží perlovský skript procházející původní konfigurační soubor, který údaje přidává do databáze pomocí SQL příkazů.

Importovací skript načte nastavení u DHCP skupin, vyjma nastavení uvozených klíčovým slovem `if`. Tento problém se týká pouze malého počtu skupin a chybějící nastavení lze manuálně doplnit.

Jak již bylo zmíněno v kapitole 5.4.5, dynamické pooly jsou generovány na základě DNS záznamů, jenž mají přiřazenou příslušnost k danému poolu. Jednotlivé pooly je nutné vytvořit ve webovém rozhraní a přiřadit do nich příslušné záznamy pomocí skriptu `./modhosts`. Nebyly-li zmiňované záznamy rozepsány, ale zadány v DNS konfiguraci direktivou `$GENERATE`, je vhodné využít skript `./generatehosts`, který umožňuje vytvořené záznamy přímo přiřadit do zvoleného poolu.

Předposledním krokem je aktivita točící se kolem dynamického přidělování IP adres. Systém *Sauron*, jak již bylo zmíněno v kapitole 5.4.5, zachází s dynamicky přidělovanými IP adresami na první pohled nestandardně. Při importu standardní DHCP konfigurace nejsou proto záznamy o příslušných hostech vůbec načteny, jelikož neobsahují tag `fixed-address` a je nutno je dodatečně přidat.

Nejprve je nutno ve webovém rozhraní vytvořit skupinu typu `DHCP class` a v jednotlivých dynamických poolech povolit přístup hostů z této skupiny. Pro automatickou opravu je vytvořen skript, který prochází původní konfigurační soubor, hledá záznamy bez specifikované IP adresy a odpovídajícím statickým záznamům přidá členství v oné skupině typu `DHCP class`.

Nakonec zbývá ještě opravit MX-šablony, jelikož při importu je místo specifikace proměnne `$DOMAIN` načteno cosi ve tvaru `$domain.zona.cz`. Po nahrazení správnou specifikací lze konstatovat, že v systému *Sauron* jsou načteny a připraveny k použití kompletní konfigurace DNS i DHCP.

## 5.6 Generování konfigurace

Tato kapitola je věnována problematice vytvoření konfiguračních souborů z dat uložených v databázi systému *Sauron* a jejich následnou distribucí na příslušné servery.

### 5.6.1 Konfigurace DNS

Vygenerování konfiguračních souborů pro DNS server má na starosti skript `./sauron` umístěný v adresáři `/usr/local/sauron/`, musí být spouštěný s parametry `--bind --updateserial <jméno serveru> <adresář>`. Ve specifikovaném adresáři je vytvořen konfigurační soubor `named.conf` a zónové soubory s příponou `.zone` pro každou přímou a reverzní zónu.

Při importu DNS konfigurace, popsané v kapitole 5.5.1, bylo naznačeno, že s některými typy zón si systém *Sauron* neumí poradit a je nutné je přiřadit do vygenerované konfigurace (obrázek 3). Zóny, kterých se to týká, jsou uchovávány v pomocném adresáři a do vygenerované konfigurace jsou přidány příslušné odkazy. Takto upravená konfigurace je připravena k použití na serveru.

### 5.6.2 Konfigurace DHCP

Konfigurační soubor pro DHCP je vytvářen také skriptem `./sauron`, ale tentokrát s parametry `--dhcp <jméno serveru> <adresář>`. Ve specifikovaném adresáři se objeví soubor `dhcpd.conf`.

Konfigurace vytvořená systémem *Sauron* má jednu drobnou vadu, kterou je nutné odstranit. Definice hostů uvozená klíčovým slovem `subclass` je pouze nutnou, nikoliv však postačující, podmínkou umožňující přístup do dynamického poolu a je tedy nutné pro každý takovýto záznam ještě přidat odpovídající definici hosta bez specifikovaného tagu `fixed-address`. Po vykonání této činnosti externím skriptem je konfigurace připravena k distribuci na server.

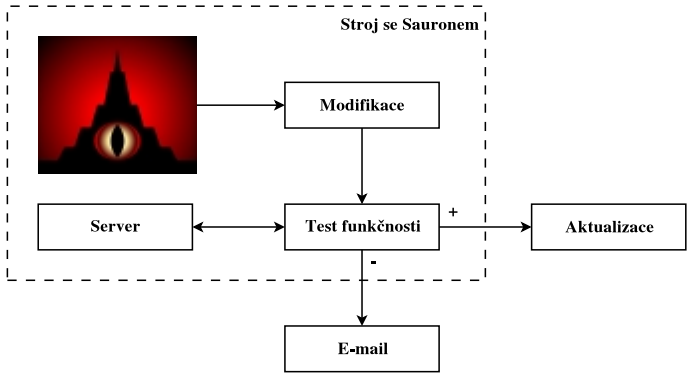
### 5.6.3 Distribuce na servery

Předpodkládejme, že do systému *Sauron* přes webové rozhraní přistupuje řada uživatelů. Není možné, aby každý z nich měl přístup ke generování konfigurace pomocí příkazové řádky a tedy také práva manipulovat s databází. Zasíláním e-mailů s notifikací změn správci systému není také příliš rozumné.

Vhodným řešením je plně automatické generování konfigurace a její následná distribuce na DNS a DHCP server, což je prováděno spouštěním příslušného skriptu v rozumných intervalech za pomocí `cron`u. Vzhledem k tomu, že se uživatelé mohou při vyplňování údajů ve webovém rozhraní dopustit syntaktických chyb – zejména v DHCP konfiguraci – je nutné před distribucí konfigurace na servery otestovat její správnost.

Postup automatické distribuce konfigurace ze systému *Sauron* na server je znázorněn na obrázku 4. Typ serveru není specifikován, neboť stejný postup je aplikován jak pro DNS tak i pro DHCP konfiguraci. Vygenerované konfigurační soubory jsou nejprve upraveny (viz 5.6.1 a 5.6.2) a v dalším kroku otestovány. Na stejném stroji, na němž běží systém *Sauron*, je nainstalován pomocný DNS a DHCP server (stejné verze jako na strojích poskytujících tyto služby). Konfigurační soubor může být poslán do ostrého provozu pouze tehdy, podaří-li se jej bez problému načíst příslušnému pomocnému serveru. V opačném případě je zaslán email správci systému *Sauron*, který je z logu schopen zjistit kde se stala chyba a zajistit nápravu.

Uvedený postup zajistí, aby byla DNS i DHCP služba poskytována v poslední funkční konfiguraci. Pro synchronizaci DNS a DHCP konfigurací je rozumné nechat propagovat změny pouze tehdy, jsou-li obě konfigurace bez chyb.




Obr. 4 Postup při distribuci konfiguračních souborů

# 6 Závěr

V článku byl podrobně představen systém *Sauron*, což je jeden ze zástupců OpenSource systémů pro jednotnou správu DNS a DHCP. Byly popsány praktické úkony od instalace systému až po běžný provoz.

V současné době je *Sauron* nasazen také na Západočeské univerzitě v Plzni a momentálně se rozšiřují počty uživatelů a distribuují příslušná přístupová práva. Během několika týdnů používání si uživatelé, po určitém počátečním šoku z nového a neznámého, systém *Sauron* vcelku oblíbili. Drobné výhrady lze mít k nástrojům dostupných z příkazové řádky, které příliš dbají na striktní oddělování DNS a DHCP. Jisté rezervy jsou také v rozsahu možností poskytovaných nástrojů, ale celkově je zavedení systému *Sauron* zcela jistě přínosem.

# Literatura

[1] http://osuosl.org/projects/maintain

[2] http://acs-wiki.andrew.cmu.edu/twiki/bin/view/Netreg/WebHome

[3] http://sauron.jyu.fi/

[4] http://search.cpan.org/dist/Crypt-RC5/

[5] ftp://ftp.rs.internic.net/domain/

[6] http://standards.ieee.org/regauth/oui/index.shtml

# Detekce spamu

**Jaroslav Šnajdr**

E-mail: jsnajdr@kerio.com

**Abstrakt**

*Spam je dnes nepochybně největším problémem elektronické pošty a do boje proti němu jsou investovány značné intelektuální i finanční zdroje. První část příspěvku přiblíží historii spamu, ekonomické důvody jeho existence a fungování spamové komunity. Dále budou probrány technické prostředky pro detekci spamu. Jednak rozšíření protokolu SMTP pro autorizaci odesílatele (SPF, SRS, Sender ID, DomainKeys), a také statistické metody pro analýzu obsahu zpráv, tzv. Bayesiánské filtry. Bude probrána architektura programu SpamAssassin jakožto komplexního nástroje pro detekci spamu, a dále principy programu DSPAM, který používá pokročilé algoritmy pro statistickou analýzu zpráv.*

Příspěvek nebyl dodán.

# Praktická zkušenost s implementací poštovní brány s antivirovou a antispamovou ochranou založené na svobodném software

**Miloš Wimmer**

E-mail: wimmer@civ.zcu.cz

**Abstrakt**

*Příspěvek je věnován popisu filozofie a provozu produkčního systému elektronické pošty Západočeské univerzity vybaveného nástroji antivirové a antispamové kontroly, který je založen na výhradně svobodném software (linux, spamassassin, clamav, cbl, horde-webmail, ...). Zmíněny jsou i další varianty (MailScanner) použité v jiných sítích.*

Pošta adresovaná zaměstnancům a studentům Západočeské univerzity je doručována na centrální poštovní server (do prostředí Orion). Uživatelé s ní pracují prostřednictvím poštovních klientů podporujících protokoly vzdáleného přístupu se zabezpečením IMAP-SSL nebo POP-SSL anebo prostřednictvím WWW brány WebMail.

## Stručný popis serveru

Centrální poštovní server běží na serveru Dell PowerEdge 1750, který je vybaven dvojicí procesorů Intel Xeon 3.06 GHz, dvojicí gigabitových síťových karet a 2 GB RAM. Datový prostor 600 GB je alokován v systému RAID5 na diskovém poli SAN PowerSTOR FF4000 připojeném přes Fibre Channel.

Operačním systémem serveru je GNU/Linux, distribuce Debian s jádrem 2.6.12. Na místě souborového systému používáme žurnálovací souborový systém XFS.

Distribuci poštovních zpráv zajišťuje sendmail, na pozici IMAP a POP serveru používáme upravený UW IMAP server a lokální doručování provádí upravený program maildrop, který podporuje třídění zpráv podle vlastního jazyka pravidel. Používáme klasický formát složek mbox, kdy jsou všechny zprávy dané

složky uloženy v jednom souboru. Ke kontrole zpráv proti nevyžádané poště používáme antispamový produkt Spamassassin a ke kontrole proti virům antivirový produkt ClamAV.

Na serveru není použito žádné programové vybavení, které by nemělo statut svobodného software.

Konta uživatelů se vytvářejí automaticky podle centrálního registru a jejich celkový objem se pohybuje kolem 15 tisíc. Autentizace uživatelů probíhá proti Kerberos serveru ZČU. Uživatelé se tedy přihlašují svým jménem a heslem, které používají v celém prostředí Orion.

Zaměstnanci mají přidělenu diskovou kvótu o velikosti 50 MB, studenti o velikosti 20 MB. Kvótu si může každý uživatel sám jednorázově navýšit na dvojnásobek. Přidělená disková kvóta se vztahuje na všechny zprávy a složky dohromady – tedy na nově doručené zprávy uložené ve složce „Doručená pošta – Inbox“ i na zprávy uložené v osobních složkách.

Uživatelé nemají ke svému kontu umožněn přímý přístup – nemohou se tedy k němu přihlásit službami telnet, ssh ani ftp. Povolený přístup k serveru mají pouze na úrovni služeb IMAP, POP a WWW se zabezpečením SSL. Službami IMAP a POP pracují se svojí poštou, pomocí brány WWW na adrese http://mail.zcu.cz/ si mohou nastavovat přesměrování pošty, způsob kontroly nově doručovaných zpráv proti spamům a virům i pravidla pro automatické třídění zpráv. Dále mohou získávat informace o zaplnění přiděleného diskového prostoru a mohou si stahovat své celé poštovní složky.

## Adresování

V síti Západočeské univerzity byl zvolen subdoménový systém adresování elektronické pošty, který se velmi osvědčil. Jeho princip spočívá v tom, že e-mailová adresa uživatele je tvořena jeho uživatelským jménem následovaným znakem ‚@‘, subdoménou přidělenou podle pracovního zařazení uživatele a jménem domény Západočeské univerzity ‚zcu.cz‘. U zaměstnanců je subdoména tvořena oficiální zkratkou jejich katedry (tedy např. ‚kiv‘ v případě katedry KIV), u všech studentů byla zvolena pevná subdoména ‚students‘.

Takže např. e-mailová adresa zaměstnance katedry KIV pana Zajíčka má podobu

<zajicek@kiv.zcu.cz>,

zatímco e-mailová adresa studenta ZČU pana Hrocha je

<hroch@students.zcu.cz>.

Tímto způsobem adresování lze garantovat neměnnost e-mailové adresy uživatelů na dlouhou dobu dopředu (leda by katedra projevila o změnu sama zájem), protože se v e-mailových adresách neobjevuje jméno serveru. Navíc lze kdykoli vyjít vstříc případnému požadavku katedry převést doručování pošty

z centrálně spravovaného serveru na její vlastní katedrální server. Také v tomto případě by z pohledu adresování proběhla celá změna transparentně.

Ačkoli je pošta doručována na centrální poštovní server, neznamená to, že všichni uživatelé musí v tomto prostředí se svými zprávami pracovat – mohou si nastavit přesměrování své pošty na jiný server.

## Systém antivirové/antispamové kontroly

Na poštovním serveru provozujeme systém antivirové/antispamové kontroly založený na produktech ClamAV a Spamassassin. Oba produkty jsou vyvíjeny jako svobodný software, takže je lze volně používat.

Systém umožňuje kontrolovat nově doručované zprávy na přítomnost virů a nevyžádaného obsahu (tzv. spamu) pro uživatele, kteří si některou z variant kontrol nastaví. Globálně se tedy žádná kontrola neprovádí a rozhodnutí o jejím uplatnění i následných akcích je ponecháno plně na uvážení a nastavení každého uživatele.

Transportní obálka doručovaných zpráv, v nichž je rozpoznán virus nebo spam, je opatřena značkou, kterou lze následně využít pro vykonání automatické operace s takovými zprávami (např. přesun do určené složky nebo smazání).

Antivirová kontrola pouze značkuje zprávy, ve kterých byl rozpoznán virus – to znamená, že do transportní obálky zprávy přidává řádku

X-Virus-Flag: YES

a dále pak řádku s informací o nalezeném viru:

X-Virus-Status: Yes, found virus Worm.SomeFool.Gen-1

Obsah zprávy se nemodifikuje a nalezený virus se neodstraňuje. Elektronickou poštu považujeme za transportní službu, jejímž úkolem je doručit příjemci v nezměněné podobě to, co mu bylo odesláno. Antivirová kontrola tedy nenahrazuje antivirovou ochranu, kterou poskytuje rezidentní antivirový program běžící na pracovních stanicích s operačním systémem Windows. Přináší uživatelům možnost odstínit je od nežádoucí pošty a zavirované zprávy automaticky přesouvat do určené složky anebo je ihned mazat.

Antispamová kontrola značkuje zprávy, ve kterých byl rozpoznán nevyžádaný obsah/spam – to znamená, že do transportní obálky zprávy přidává řádku

X-Spam-Flag: YES

a dále pak řádky s informací o bodovém ohodnocení spamu:

X-Spam-Level: **********

X-Spam-Status: Yes, hits=18.9 required=5.0 tests=BAYES_99,

Obsah zprávy se nemodifikuje.

Nastavení obou kontrol je pro uživatele velmi jednoduché a provádí se pomocí intuitivního WWW formuláře (viz obrázek 1).


Obr. 1 Formulář pro nastavení kontrol a třídění pošty

Na rozdíl od řady jiných institucí nemažeme zprávy označené jako virus nebo spam rovnou. K takovému postupu máme především silné filozofické důvody (popsané níže). Dále vycházíme z toho, že uživatel, který má o automatickou kontrolu zpráv zájem, si ji může snadno nastavit sám a sám si také může určit, co s označenými zprávami dělat. Navíc v případě nevyžádaných zpráv nemůžeme s úplnou jistotou říci, která zpráva je spam a která není. Antispamová kontrola se to pokouší odhadnout, ale je to jen odhad – i když velmi úspěšný. Oproti jinde používané striktní politice uplatňované na všechny příjemce dáváme přednost více demokratickému způsobu, který přenáší volbu nastavení konkrétního chování kontrol do rukou jednotlivých uživatelů.

## Filozofie antivirové ochrany

Operační systémy Windows jsou díky své vnitřní architektuře snadno zranitelné napadením virů. Se značným rozšířením elektronické pošty se tato služba

stává častým médiem, kterým se viry šíří. Řada institucí zabezpečuje své systémy elektronické pošty různými antivirovými programy, které ve snaze ochránit uživatele před zavlečením infekce odstraňují z doručovaných zpráv nebezpečné přílohy anebo je modifikují. Tento způsob antivirové ochrany však považujeme za nedostatečný a málo účinný a proto preferujeme takové řešení, které zaručuje mnohem větší stupeň ochrany. Tím je ochrana koncových pracovních stanic pracujících s operačním systémem Windows rezidentním antivirovým štítem, který je na každé chráněné stanici nainstalován. Nejprve uvedu hlavní důvody, proč je na centrálním poštovním serveru provozován systém antivirové kontroly ve značkovacím režimu.

## Filozofický pohled

Poštovní systém má zprávu doručit v nezměněném tvaru. Obsah zprávy má být pro transportní systém nedotknutelný a je proto nesprávné, aby v něm docházelo k modifikacím (např. změnou přípony přiloženého souboru, odstraněním zavirovaného souboru v příloze apod.).

Dále jsem hluboce přesvědčen, že uživatelé nepožadují uplatnění antivirového systému na poštovním serveru – uživatelé požadují (nejlépe úplné) ochránění svých pracovních stanic před útokem virů. A z tohoto pohledu je antivirový systém na poštovním serveru nedostatečný.

## Dílčí řešení

Uživatelé mohou získat počítačový virus mnoha způsoby:

- elektronickou poštou
- stažením souboru z WWW, FTP, Network News a jiných aplikačních serverů
- stažením souboru ze sdíleného nebo vyměnitelného disku nebo diskety
- jinak

Z toho je zřejmé, že antivirový systém, který by na poštovním serveru modifikoval obsah zpráv nebo viry odstraňoval, stejně nemůže ochránit pracovní stanici dostatečně, riziko nákazy může pouze snížit.

## Problémy v dohledné budoucnosti

S očekávaným zavedením technologií digitálních podpisů ještě vzrostou požadavky na zachování integrity přenášených zpráv. Zpráva podepsaná digitálním podpisem nemůže být nikterak modifikována, protože její příjemce pak nemůže

ověřit autenticitu odesílatele. Společně s tím začnou uživatelé ve velké míře používat ve svých poštovních klientech i šifrovací technologie, které dovolují dekódovat obsah zprávy jen určenému příjemci. V těchto případech se antivirový systém na poštovním serveru stává zcela neúčinným, protože případný virus nemůže v zašifrované zprávě odhalit. V této souvislosti musím připomenout, že šifrovací technologie se používají již delší dobu. Běžným uživatelům však zatím připadají málo komfortní anebo pro jejich použití nemají silný důvod a proto je používají méně často.

### Zpochybnění kvality služby

Obecně platí, že nasazení jakéhokoli globálně uplatňovaného filtru snižuje důvěru v kvalitu služby. Uživatelé pak totiž málokdy ví, jak je filtr (v tomto případě antivirový systém na poštovním serveru) nastaven a jak se chová a podvědomě mu nedůvěřují.

Za nejúčinnější ochranu pracovních stanic pracujících s operačním systémem Windows považujeme rezidentní antivirový štít v kombinaci s lokální bránou firewallu. Uvědomujeme si, že provozování antivirového systému na pracovních stanicích může klást na některé uživatele zvýšený nárok v podobě potřeby jeho instalace, ale jsme přesvědčeni, že výhody tohoto řešení, jeho filozofická čistota a hlavně jeho vysoká účinnost jej plně vyváží.

## Zkušenosti s provozem

Popsaný centrální poštovní server provozujeme čtyři roky (při průběžné modernizaci hardwarového i softwarového vybavení). V pracovních hodinách bývá zatížen (load) v rozmezí hodnot 2 až 4. Mimo jiného na něm běží kolem 500 procesů imapd a 200 procesů sendmail. Datové přenosy diskového pole se pohybují kolem 30 MB/s pro čtení a 5 MB pro zápis.

Komprimované zálohování dat se provádí každý den na zálohovacího robota a pro operativní přístup k zálohám současně na lokální diskové pole. Zálohy se rotují automaticky.

Některou z variant antivirové/antispamové kontroly má nastaveno 2 500 uživatelů.

S produktem ClamAV máme ty nejlepší zkušenosti. Aktualizace jeho databází známých virů se provádí automaticky každé dvě hodiny a doba rozpoznávání nových virů je stejná jako u obdobných komerčních produktů.

Spamassassin běží v režimu, kdy si buduje vlastní znalostní databázi ze zpráv, které získaly bodové hodnocení vyšší než 10. Původně jsme nechali vytvářet databáze v domovských adresářích jednotlivých uživatelů, kterým takto ohodnocené zprávy došly, ale z důvodu náročnosti na diskový prostor jsme přešli k používání

jediné velké společné databáze. Úspěšnost Spamassassinu je velmi vysoká. Téměř úplného odstínění od spamu je možné dosáhnout souběžným budováním vlastní znalostní databáze spamu v prostředí uživatelova poštovního klienta, který tuto funkci podporuje (např. Mozilla Thunderbird).

Sendmail kontroluje stroje, které na něm otevírají spojení pro doručení nové zprávy, proti databázi strojů považovaných za distributory spamu a virů. Tento systém (CBL – Composite Blocking List) běží na serveru http://cbl.abuseat.org/ a na rozdíl od jiných podobných systémů dovoluje odesílateli, který je o zablokování informován ve vrácené zprávě, odstranit záznam jeho stroje z databáze. Bez nasazení CBL byl poměr kontrolovaných zpráv označených za spam proti zprávám čistým zhruba 7 : 3, při jeho použití se poměr změnil na 3 : 3. Navíc tím šetříme své systémové zdroje. Sendmail denně doručuje více než 200 tisíc zpráv, k tomu na základě CBL odmítá kolem 40 tisíc dalších.

Zprávy, které nemohou být na serveru doručeny z důvodu přeplněné kvóty příjemce, nezůstávají ve frontě, ale jsou s příslušným hlášením odeslány odesílateli okamžitě zpět. Bez tohoto opatření by se počet zpráv ve frontě ze současné hodnoty kolem 500 dramaticky zvýšil.

## WebMail

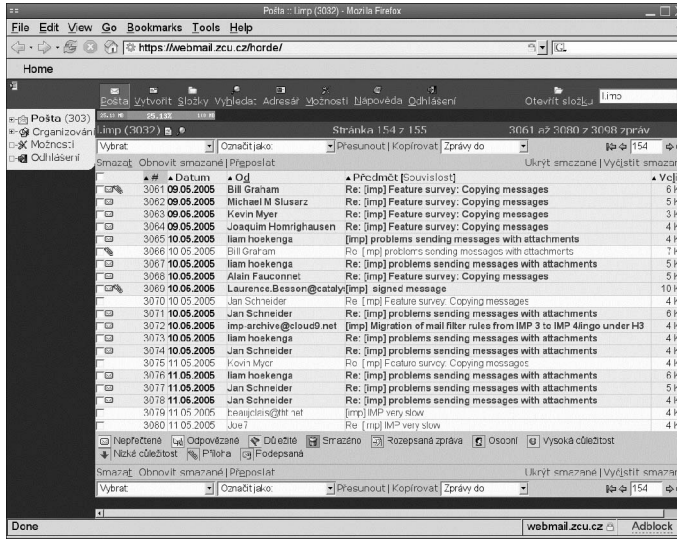
Až překvapivě velké množství uživatelů přistupuje ke svojí poště přes bránu WWW. Používáme produkt Horde-IMP a této službě říkáme WebMail. Horde je aplikační framework napsaný v PHP, nad kterým se vyvíjí řada dalších modulů. Modul pro přístup k poště se jmenuje IMP, modul pro práci s kontakty Turba, apod. Horde-IMP považuji za velmi propracovaný systém, který má velkou škálu vlastností a možností. To na druhé straně vyžaduje jeho rozsáhlejší konfiguraci. Horde je lokalizován do řady národních jazyků.

WebMail provozujeme na jiném serveru, který je kromě běžného připojení do počítačové sítě ještě navíc přímo propojen s centrálním poštovním serverem přes druhou síťovou kartu (back-to-back).

## MailScanner

Pro řadu institucí je řešení, které používáme, příliš liberální a preferují striktní politiku, která na poštovním serveru viry ze zpráv odstraňuje, případně zavirované zprávy maže bez doručení. Nezřídka se můžeme setkat s nastavením filtrů a nástrojů, kterými poštovní server modifikuje obsah zpráv (např. přejmenovává nebo odstraňuje přílohy s příponou .exe, mění prováděcí značky v přiložených HTML souborech, apod.).

Pro tyto případy mohu doporučit volně dostupný produkt MailScanner. Jde o balík skriptů napsaných v perlu, který k detekci virů a spamu využívá jiných




Obr. 2 Ukázka práce v prostředí WebMailu

nástrojů (třeba právě ClamAV a Spamassassin). Filozofie MailScanneru vychází z jedné globální politiky uplatňované na všechny zprávy procházející poštovním serverem. V rozsáhlém konfiguračním souboru se nastavuje reakce a chování systému na nejrůznější případy – rozpoznání viru, spamu, přílohy s exe souborem, nastavení odesílání hlášení odesílateli zachycené zprávy apod.

MailScanner vyžaduje spuštění dvou démonů MTA (např. sendmail nebo postfix) na serveru. První démon jen přijímá nově příchozí zprávy a ukládá je do speciální fronty, odkud je MailScanner vyzvedává ke svému zpracování. Zpracované zprávy pak MailScanner zapisuje do standardní mailové fronty, odkud je vyzvedává druhý démon MTA k následnému obvyklému doručení.

# Otevřená okna

**Zdeněk Šustr**

E-mail: sustr4@civ.zcu.cz

**Klíčová slova:** Windows, AFS, Kerberos, open software

**Abstrakt**

*Centrum informatizace a výpočetní techniky Západočeské univerzity zahájilo přechod k distribuované výpočetní infrastruktuře na bázi otevřených systémů – tzv. prostředí Orion – v roce 1995. V roce 1998 pak začal projekt začlenění osobních počítačů vybavených tradičně „uzavřenými“ systémy firmy Microsoft do této otevřené infrastruktury. První výsledek projektu – distribuce OrioNT – byl již na konferenci EurOpen prezentován v roce 1999. Dnes má ZČU za sebou hromadný přechod k následné verzi OrionXP (více než 600 instalací), která je výsledkem spolupráce CIV s několika zahraničními univerzitami a podporuje využití dalších typů otevřených mechanizmů – (Open)AFS, Kerberos, kx509, apod.*

**Abstract**

*The Centre for Information Technology of the University of West Bohemia has commenced a gradual transformation of its computer infrastracture into a system based purely on open standards – so called Orion – in 1995. In 1998, a new project aimed at the interoperability of traditionally „closed“ systems manufactured by the Microsoft Corporation with the open infrastructure was started. The first-generation result of the project – the OrioNT distribution – has already been presented at the EurOpen Conference in 1999. As of today, the UWB has almost finished its transition to the next-generation distribution called OrionXP (over 600 installations). OrionXP, being a result of the coopration of the CIV with several foregin universities, supports even more open protocols, services and data formats (OpenAFS, kerberos, kx509, etc.) than its predecestor.*

Ve velké organizaci napojené mnoha kanály na státní správu se lze prozatím jen těžko obejít bez nasazení komerčních a „uzavřených“ produktů, především operačních systémů a aplikací. Přesto, má-li instituce zájem, lze čerpat alespoň část výhod poskytovaných otevřeným či „svobodným“ softwarem.

Západočeská univerzita v současnosti provozuje na většině svých administrativních pracovišť komerční operační systém Windows XP. Při tom infrastruktura jejího distribuovaného výpočetního prostředí *Orion* je vystavěna na otevřených systémech a standardech. Při snaze napojit Windows XP na tuto otevřenou infrastrukturu vznikla zvláštní distribuce OrionXP – *otevřená okna*.[1]

# 1 Vytýčení cílů

Windows XP bylo třeba přizpůsobit práci v otevřeném prostředí ZČU v několika ohledech:

- Napojení na infrastrukturu Kerberos pro ověřování uživatelů při přihlášení k systému i k doplňkovým službám.
- Napojení na centrální diskový prostor AFS vč. získání patřičných pověření.
- Autorizovaný přístup k dalším službám distribuovaného prostředí vč. služeb webových.
- Vybavení stanic dalšími „free“ a open-source produkty a zavedení vhodných standardů pro ukládání dat.

# 2 Napojení na infrastrukturu Kerberos

OS Windows používá Kerberos jako standardní mechanizmus ověřování uživatelů od verze 2000. Firma Microsoft sice pojala standard Kerberos „po svém“ a vnesla do své implementace jistou nekompatibilitu, ale součinnosti se standardním prostředím MIT Kerberos lze přesto dosáhnout.

Stanici s Windows 2000 či XP lze snadno zkonfigurovat tak, aby se pokoušela o ověření existujícího uživatele nikoli vůči vlastní autoritě, ale vůči libovolnému zadanému KDC. Toho lze dosáhnout jednoduchým doplněním patřičných odkazů do registru Windows, anebo ještě snáze použitím nástroje `ksetup`, který je součástí balíčku *Support Tools* distribuovaného na instalačních médiích spolu s vlastní instalací Windows:

```
ksetup /AddKdc <Kerberos Realm> <KDC 1>
...
ksetup /AddKdc <Kerberos Realm> <KDC n>
ksetup /AddKpasswd <Kerberos Realm> <kpasswd server>
```

[1] či alespoň okna *pootevřená*...

Příkaz `ksetup` nedělá nic jiného, než že zapisuje do patřičného registrového klíče. Výsledný záznam vypadá takto a lze jej se stejným efektem pořídit i libovolnými dalšími nástroji pro práci s registrem:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos
\Domains\<kerberos realm>]
"KdcNames"=<seznam KDC>
"KpasswdNames"=<KAdmin server>
```

Nastavení cesty k serveru `kpasswd` (tj. `kadmin`) není povinné a ověřování probíhá i bez ní. Ovšem s nastavením tohoto serveru je možné provádět přímo z Windows standardním způsobem např. změny hesel.

Při přihlašování uživatele k lokální stanici se nevyžaduje jen ověření uživatelské identity, ale také ověření identity stanice. Proto je třeba vytvořit databázích Kerbera záznam příslušného počítače. To lze zajistit jednotlivě pro každou stanici, ovšem při velkém počtu zařízení je snazší začlenit všechny stanice do infrastruktury adresářových služeb *Active Directory* (AD) a definovat důvěryhodný vztah mezi danou sférou Kerbera a doménou Active Directory. Tím se zjistí vzájemná důvěra mezi ověřovacími servery a všemi stanicemi zařazenými do domény AD.

## 2.1 Úloha Active Directory

Active Directory je „proprietárním“ produktem, k jehož nasazení má smysl sáhnout pouze v odůvodněných případech. Mají-li se v distribuovaném výpočetním prostředí používat řádově jednotky stanic s OS Windows, a mají-li na nich pracovat řádově jednotky uživatelů, má smysl zajistit níže popsané služby jiným způsobem a použití AD se vyhnout. V ostatních případech se nasazení AD vyplatí, přestože se jím do systému vnáší další zdroj potenciálně nedeterministického chování a další služba, o niž je třeba se starat.

Co lze zavedením Active Directory získat:

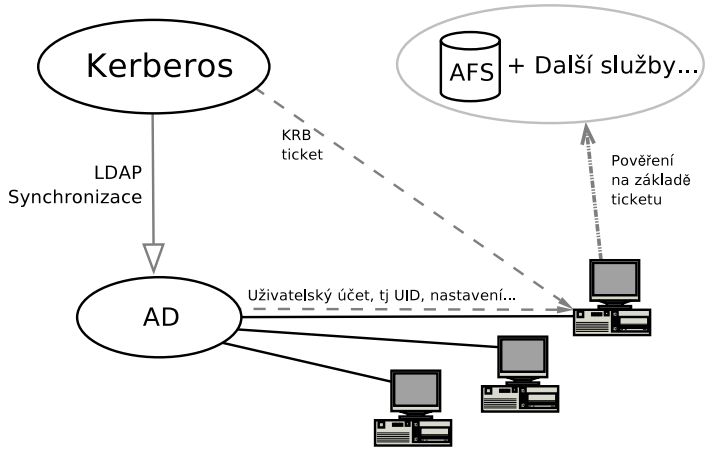
- **Jediný vztah důvěryhodnosti**, který je třeba zavádět mezi stanicemi a sférou Kerberos. Důvěřuje-li Kerberos dané doméně AD, přenáší se tato důvěra na všechny počítače, které jsou její součástí.

- **Centrální místo pro vytvoření uživatelských účtů.** Uživatelé se mohou ověřovat vůči autoritám Kerbera, ale jejich účty (tj. identita a soubor uživatelských nastavení) musejí na koncových stanicích existovat. Bez AD by bylo třeba pro každého uživatele vytvářet samostatně na každé stanici vlastní účet.

- **Centrální místo pro registraci počítačů**, které mají být do infrastruktury zařazeny. Krom sdílení důvěryhodného vztahu s ověřovacími autoritami jsou zde další dílčí výhody.

Active Directory je v zásadě adresářová služba, jíž dokáží stanice s Windows využívat k plnění celé řady funkcí. Zcela zásadní význam pro řešenou problematiku má mechanizmus distribuce uživatelských účtů, které jsou po založení v AD známy všem stanicím zapojeným do příslušné infrastruktury. S Active Direcotry lze navíc pracovat prostřednictvím LDAP rozhraní, takže využívá-li organizace adresářové služby např. právě na bázi LDAP, lze velmi snadno automatizovat vytváření uživatelských účtů v Active Directory.

V prostředí Windows i Active Directory je navíc implementována další významná funkce – mapování uživatelských účtů. Díky tomu lze jednoznačně určit, které identitě v prostředí AD odpovídá která identita (principal) v centrální uživatelské Kerberos-based základně.




Obr. 1 Schéma zapojení koncových stanic do struktury autentizačních služeb

Použití Active Directory přináší – přinejmenším v prostředí ZČU a patrně v prostředí rozsáhlých institucí akademického charakteru obecně – další nespornou výhodu. Jelikož řada pracovišť (kateder) disponuje vlastními prostředky pro zajištění počítačové vybavenosti a má také vlastní představy o jejich využití, vyskytuje se často potřeba napojit na centrální ověřovací infrastrukturu „cizí“ zařízení. Tato externí pracoviště si obvykle zajišťují vlastní správce nejrůznějšího – mnohdy pochybného – původu a kvalifikace, na něž je třeba delegovat práva potřebná k zařazení stanic do ověřovací infrastruktury.

V tomto případě se zdá snazší a méně rizikové zpřístupnit takovým lokálním správcům část stromu Active Directory a zcela je oddělit od infrastruktury Kerbera.

### 2.1.1 Synchronizace uživatelské základny LDAP/AD

Podstatné údaje o uživatelských účtech zavedených v rozsáhlejším výpočetním prostředí lze exportovat do souboru ve formátu LDIF. Informace v této podobě pak lze replikovat v prostředí adresářových služeb AD. Za povšimnutí stojí fakt, že v daném souboru se uživateli nastavují některé proměnné prostředí a také náhodné heslo, což efektivně znemožňuje obejití ověřovacího mechanizmu Kerberos a přihlášení přímo k doméně AD (při takovém přihlášení by uživateli tak či onak nefungovaly některé služby, ale vlastní přihlášení k systému a přístup k místně uloženým datům by si dokázal zajistit).

Příklad záznamu jednoho uživatelského účtu ve formátu LDIF je uveden níže:

```
 dn: cn=<username>, ou=<AD Org. Unit>, <dc=...>
changetype: add
objectclass: user
...
altSecurityIdentities: Kerberos:<username>@<realm>
userPrincipalName: <username>@<AD domain>
ProfilePath: %ROAMINGUSERPROFILE%\<username>
...

dn: cn=<username>, ou=<AD Org. Unit>, <dc=...>
changetype: modify
replace: unicodePwd
unicodePwd:: <random password>

dn: cn=<username>, ou=<AD Org. Unit>, <dc=...>
changetype: modify
replace: userAccountControl
userAccountControl: <No.>
```

V prostředí ZČU/Orion je zcela zřejmý vztah mezi světy LDAP a AD. Centrální adresářová služba LDAP je vždy **zdrojem** informací, AD je vždy **cílem**. Synchronizaci opačným směrem není nikdy třeba řešit.

Vložení dat obsažených v souboru LDIF do aktivní databáze AD lze provést dvěma způsoby. Pro jednotázové akce postačí ruční vložení, které se musí provádět přímo na řadiči patřičné domény AD. Součástí již zmiňovaného balíčku *Support Tools* je nástroj `ldifde`, s jehož pomocí lze provést import záznamů uvedených v souboru do databáze Active Directory.

Pro rutinní použití je vhodnější použít síťové rozhraní LDAP a informace o uživatelské základně synchronizovat vzdáleně v rámci dávkového zpracování souvisejících úloh:

```
ldapmodify -x -c -H ldaps://<Řadič domény> -D
‘ cn=<Účet s dostatečným oprávněním>,cn=<cílová org. jednotka>,
<DC=...>‘  -w <heslo k oprávněnému účtu> < <soubor LDIF>
```

# 3 Napojení na AFS

Souborový systém AFS se začal na ZČU rutinně používat v roce 1995 a jakmile to bylo možné, začali k němu přistupovat i uživatelé Windows.

Prosté připojení stanice s OS Windows k diskovému prostoru AFS je triviální. Použít lze např. klienta OpenAFS, který je volně ke stažení na stránkách projektu.

Jelikož však AFS má být jediným centrálním diskovým prostorem, který stanice využívají, musí splnit i jiné úlohy než prosté ukládání uživatelských dat. Na AFS se ukládají také nástroje pro automatizovanou správu stanic a především cestovní uživatelské profily, tj. datové struktury, do nichž se ukládají osobní nastavení uživatelů.

## 3.1 Problematika cestovních profilů

Systémy firmy Microsoft dodnes neznají pojem „symbolický link“ a vytvoření konfiguračního adresáře, v němž by bylo uloženo uživatelské nastavení použitelné na libovolném z desítek či stovek počítačů, k nimž se uživatel může přihlásit, se díky tomu stává relativně obtížně řešitelným problémem.

V prostředí Windows se tomuto konfiguračnímu adresáři říká „uživatelský profil“. Profil se standardně ukládá na systémový disk Windows (známá cesta `C:\Documents and Settings\`) a i když formálně existuje možnost jeho přemístění na jiný logický svazek či dokonce na jiný disk, v praxi toto řešení obvykle nelze použít. Řada SW produktů totiž počítá s pevným umístěním konfiguračních dat a jejich přesměrování nemusí vůbec brát v úvahu.

Vzhledem k neexistenci symbolických odkazů na jiné souborové systémy zároveň ani není možné systému jinou lokaci jednoduše „podsunout“.

Používání jednotného uživatelského nastavení řeší Microsoft zavedením tzv. cestovního profilu. Celý profilový adresář se vším všudy se při přihlášení uživatele – tj. ještě v rámci přihlašovacího procesu a před zahájením jakékoli práce – zkopíruje ze zadané lokace na místní disk. Při odhlášení uživatele se kopíruje zase zpět.[2]

[2] To může působit značné problémy, pokud uživatel do svého lokálního profilu přesune větší množství dat a překročí diskovou kvótu, která je mu vyhrazena v centrálním diskovém prostoru. Profil se řádně neuloží a při příštím pokusu o přihlášení může být poškozen, mohou chybět některá data, anebo se přihlášení vůbec nemusí podařit. Na ZČU se tento problém zatím nepodařilo vyřešit – platí pouze doporučení, aby si uživatelé do profilu ukládali co nejméně dat.

Při standardním nasazení Windows v rozsáhlejší síti se počítá s tím, že odložené cestovní profily (resp. cestovní kopie profilů) budou uloženy na souborovém serveru s OS Windows, který je začleněn do infrastruktury Active Directory a k řízení přístupů používá jí vydaná pověření. To v případě použití AFS neplatí a chceme-li cestovní kopie profilů ukládat na AFS, musí stanice ještě před pokusem o stažení cestovních dat získat pověření pro přístup k privátnímu adresáři (obvykle části diskového prostoru vyhrazeného konkrétnímu uživateli) na AFS, v němž jsou uložena.

Vzhledem k tomu, jakou formou se ve Windows zpracovávají standardní přihlašovací skripty, není možné zajistit získání potřebných pověření až při jejich vykonání. Uživatelský přihlašovací skript se totiž spouští až po otevření prostředí, tedy i po načtení uživatelského profilu.

### 3.1.1 Včasné získání pověření pro AFS

Řešení používané i na ZČU navrhl Rodney M. Dyer z UNC Charlotte. Spočívá v drobné úpravě klienta OpenAFS, jehož zdrojový kód je volně dostupný. Součástí instalace klienta do prostředí Windows je knihovna `AFSLogon.dll`, jejíž funkce `NPLogonNotify` se volá v okamžiku, kdy interaktivní uživatel zadal své uživatelské jméno a heslo. Primárním účelem této funkce je zajistit tzv. *integrated logon*, tj. získání pověření pro AFS v případě, že lokálně zavedený uživatel používá k přihlášení k lokálnímu systému stejné uživatelské jméno a heslo jako pro přístup k AFS.

Jednoduchou úpravou `AFSLogon.dll` lze zajistit, aby se v okamžiku přihlášení uživatele provedly nejen standardní kroky, ale aby se také spustil další kód – např. externí skript – který včas získá potřebná pověření. V rámci tohoto doplňkového kódu je třeba provést několik důležitých kroků, které dokumentuje níže uvedený výňatek ze skriptu používaného na ZČU:

```
REM ——————————-
REM Inicializace Kerberos Cache
REM ——————————-
set KRB5CCNAME=API:reg_mod HKLM "System\CurrentControlSet\Control\
Session Manager\Environment" "KRB5CCNAME" "%KRB5CCNAME%"
REM ————————————————————
REM Získání uživatelských ticketů od Kerberos serveru
REM ————————————————————
kinit-w -w "%WlMprNotifyPassword%" "%WlMprNotifyUserName%"
REM ————————————————————-
REM Získání AFS tokenu pro uživatelský účet %XUSERNAME%
REM ————————————————————-
```

```
set SetUserToken=True
afsk5log.exe -auth -stlgtk -d
REM ————————————————————
REM Získání AFS tokenu pro systémovy účet SYSTEM
REM ————————————————————
set SetUserToken=
afsk5log.exe -auth
```

Nástroj `kinit-w`, který se ve skriptu používá, je klonem standardního `kinit` z distribuce MIT. Jedinou změnou ve srovnání se standardní verzí je schopnost akceptovat heslo ověřovaného uživatele z příkazového řádku (parametr `-w`).

Nástroj `afsk5log.exe` umí získávat pověření (token) pro AFS nejen pro aktuálního uživatele (AFS klient a s ním i výše uvedený skript běží s identitou uživatele `SYSTEM`), ale také pro uživatelský účet interaktivně přihlášený ke stanici. Oba dva účty budou v průběhu přihlašovacího procesu potřebovat přístup k cestovní kopii uživatelského profilu.

Jistou nevýhodou tohoto postupu je, že při každé změně verze AFS klienta je třeba úpravy knihovny `afslogon` implementovat znovu. Vzhledem k tomu, že jiné řešení se dosud nepodařilo najít, je bohužel třeba se s touto nepříjemností smířit.

### 3.1.2 Souběžná existence několika profilů

Každý uživatel má ve struktuře Active Directory právě jeden záznam a tím pádem také pouze jedinou cestu k cestovní kopii svého uživatelského profilu. Jak již bylo uvedeno výše, ve velkých organizacích typu ZČU existují samostatná oddělení, která si své výpočetní prostředky spravují do značné míry sama a bylo by záhodno, aby uživatelská nastavení vzniklá v jejich prostředí zůstala na toto prostředí omezena a neprojevovala se na jiných pracovištích.[3]

Aby bylo možné zřídit pro jednoho uživatele několik cestovních profilů, nesmí být cesta uvedená v jeho záznamu v AD absolutní. Musí být třeba měnit význam uvedené informace podle toho, v jakém prostředí (na jakém pracovišti) se uživatel přihlašuje.

V zásadě se nabízí dvě možnosti:

- Relativní odkaz na „disk“ (např. `X:`), který bude pak možné na různých počítač mapovat na nejrůznější části lokálního či vzdáleného souborového systému.

[3]Mimo to je v zájmu Centra informatizace a výpočetní techniky nešířit na tato pracoviště mechanizmus upraveného `afslogon.dll`, protože při jeho zpracování si jednotlivé komponenty mezi sebou předávají mimo jiné uživatelská jména a hesla v otevřené podobě a to by mohlo na počítačích mimo přímou kontrolu CIVu představovat větší než přijatelné bezpečnostní riziko.

- Načtení cesty k uživatelskému prodilu z proměnné prostředí definovatelné zvláště na každé stanici – to je také postup, jemuž byla nakonec dána přednost.

V záznamu každého uživatelského účtu je uvedena v zásadě stejná cesta k cestovní kopii uživatelského profilu (viz výše příklad záznamu LDIF):

`ProfilePath: %ROAMINGUSERPROFILE%\`<*username*>

Hodnotu proměnné prostředí `%ROAMINGUSERPROFILE%` pak nastaví správce každé stanice tak, jak uzná za vhodné. Používá-li mechanizmus pro včasné získání pověření k AFS (upravou knihovnu `afslogon`), může jít o cestu k prostoru na AFS. Má-li pracoviště k dispozici souborový server na bázi Windows zařazený do stejné AD infrastruktury, může jej pro ukládání cestovních profilů také použít. A v poslední řadě může použít i místní cestu. I tak bude docházet ke stěhování profilu při přihlášení i odhlášení, ale pouze v rámci lokálního stroje, čímž se proces nejen poněkud zrychlí, ale také se případně zajistí vyšší bezpečnost,[4] sníží se závislost na vnějších zdrojích,[5] a prostředí uživatelů na dané stanici se stane zcela unikátním, což má smysl u počítačů používaných menší množinou lidí ke zcela specifickému účelu.

## 3.2 Průběh interaktivního přihlášení

Po instalaci a aktivaci všech dosud zmíněných mechanizmů by měl systém být připraven přihlásit uživatele a v rámci přihlašovací procedury získat jeho pověření pro přístup k diskovému prostoru i dalším službám a stáhnout na místní disk cestovní kopii uživatelského profilu. Jednotlivé akce probíhají v tomto pořadí:

1. **Autentizace** – Jméno (*principal*) a heslo zadané uživatelem se zasílá KDC. Proběhne-li jejich ověření bez problémů, vydá se TGT (Ticket-Granting Ticket, tj. lístek zaručující vydání dalších lístků) a zašle se zpět stanici. V této chvíli je možné přihlásit uživatele, ovšem ještě se neví, který účet se bude přihlašovat. Zároveň probíhá funkce `NPLogonNotify` a s ní i získání pověření pro čtení uživatelského profilu z AFS.

2. **Nalezení odpovídajícího uživatelského účtu** – Systémem „shora dolů“ se prohledá hierarchická struktura Active Directory a hledá se první použitelný účet – tj. účet mapovaný na daný principal.

3. **Přihlášení uživatele** – Během přihlašování probíhají takové akce jako načítání cestovního uživatelského profilu. Nebýt speciálního kódu vyko-

[4] je-li třeba

[5] AFS nemusí fungovat, tím méně fileserver s Windows...

návaného v prvním kroku, čtení by selhalo, neboť uživateli i systému by chyběla přístupová práva pro čtení cestovní kopie profilu.

4. **Následné operace** – Některé operace lze vykonat až v rámci standardního přihlašovacího skriptu. Ten se spouští až po zavedení prostředí a hodí se pro supuštění doplňkových nástrojů a služeb.

   Jednou z významných akcí, které se vykonávají v této fázi přihlášení, je konverze KRB lístků z formátu MS do formátu MIT. Jak již bylo uvedeno, firma Microsoft si standard Kerberos mírně upravila a lístky, jimiž uživatel disponuje po přihlášení k systému, nejsou plně použitelné dalšími nástroji, byť by byly s Kerberem kompatibilní. Součástí distribuce MIT Kerbera pro platformu Win32 je ovšem nástroj `ms2mit`, s jehož pomocí lze velice snadno provést potřebnout konverzi. Lístky se vyzvednou ze standradního úložiště spravovéného MS CryptoAPI, upraví se a uloží se do Kerberos Ticket Cache, kde je pak již snadno najdou ostatní aplikace.

# 4 SSO přístup k dalším službám výpočetního prostředí

Tím, že uživatel získá zároveň s přihlášením do svého pracovního prostředí i KRB lístky, se nabízí možnost zajistit mu snadný přístup k řadě centrálně poskytovaných služeb. Single-sign-on chování lze zajistit v zásadě dvěma mechanizmy:

- SASL/GSSAPI
- kx509

## 4.1 SASL/GSSAPI

SASL (Simple Authentication and Security Layer) je způsob zajištění autentizace uživatelů při používání stavových (tj. *connection-based*) síťových protokolů. GSSAPI (Generic Security Application Programming Interface) je obecný nástroj pro zajištění client-server autentizace. Řada klientů běžných síťových služeb (e-mail, ssh apod.) podporuje ověřování tímto způsobem a zjednodušuje tak uživateli přístup k příslušným zdrojům.

Na ZČU zatím neproběhl pokus o vytvoření vlastní aplikace používající GSSAPI k ověření uživatele. Dobrým zdrojem takových aplikací je však open source/free software komunita. Na stanice s Windows se tak rutinně instaluje např. poštovní klient `Pine`, nebo SSH klient `Putty` – v obou případech právě s možností SSO chování díky implementaci GSSAPI.

### 4.2 kx509

kx509 je nástroj, s jehož pomocí může uživatel disponující platným KRB lístkem získat krátkodobý certifikát X.509. S takovým ceritifátem pak může přistupovat k webovým aplikacím či dalším prostředkům podporujícím PKI.

Provozuje-li organizace kx509 infrastrukturu, je napojení stanic s Windows na její služby poměrně triviální. Klient kx509 dokáže získané certifikáty přímo vkládat do standardního úložiště certifikátů používaného produkty Microsoftu.

Ve výpočetním prostředí ZČU má již použití xk509 pro přístup k webovým aplikacím svou historii. V současnosti se ovšem rutinně nepoužívá. Plánuje se však integrace s ověřovacím mechanizmem *WebAuth*, který zajišťuje ověřování uživatelů u velké části webových služeb poskytovaných uživatelům počítačové sítě ZČU a rozšiřuje se i na služby další.

## 5 Další užitečné aplikace

V rámci přizpůsobování Windows nekomerčnímu či lépe *otevřenému* světu lze využít řadu volně dostupných aplikací, které vyplní mezery a přivedou svobodný software do podvědomí běžného uživatele. Některé SW produkty z této kategorie instalované standardně na koncových stanicích ZČU souvisí těstně s využívanou infrastrukturou (např. integrační rozhraní `WAKE`). Jiné jsou na ní nezávislé a dají se využít na všech stanicích s OS Windows k plnění běžných služeb.

### 5.1 Integrace služeb a uživatelské rozhraní

Vynikajícím nástrojem pro integrovanou práci s Kerberem a AFS je program `WAKE` (Windows AFS and Kerberos Enabler), jehož producentem je Rose-Hulman Institute of Technology.

Jde o grafické uživatelské rozhranní, které umožňuje méně znalému uživateli provádění operací běžně dostupných z příkazového řádku – např. získání či konverzi lístků, vyžádání pověření pro AFS, mapování částí AFS prostoru apod.

### 5.2 Freewarové nástroje

Na základě zkušeností z projektu lze k širokému nasazení na koncových stanicích doporučit několik freewarových nástrojů:

- `PDF Creator` – virtuální zařízení pro výstup (tisk) do PDF.
  http://sourceforge.net/projects/pdfcreator/

- `Mozilla Thunderbird` (Díky nasazení tohoto produktu se podařilo na administrativních pracovištích a v učebnách ZČU prakticky vymýtit Out-

look/Outlook Express.)
http://www.czilla.cz/produkty/thunderbird/

- `Mozilla Firefox` – postupně se prosazující náhrada za MS Internet Explorer.
http://www.czilla.cz/produkty/firefox/

### 5.3 Otevřené datové formáty

Vzhledem k poměrně špatným znalostem a jednostranné orientaci většiny uživatelů je prosazení myšlenky na používání otevřených datových formátů poměrně obtížné.

Jako schůdný krok se jeví změna standardních formátů pro ukládání klasických *office-type* dokumentů – např. ukládání výstupů z MS Wordu do formátu RTF. Problémem je, že podobný standrad jako RTF lze jen těžko hledat u dalších běžných typů dat – např. tabulek,[6] prezentací, lokálně uložených složek elektronické pošty, apod.

V nejbližší době jistě bude třeba v tomto směru podniknout další kroky a opustit proprietární formáty, prozatím je však celá věc jen ve stádiu úvah.

## 6 Odkazy

- Projekt Orion http://support.zcu.cz/orion.html
- Projekt OrionXP – začlenění Windows do otevřené infrastruktury
http://support.zcu.cz/prostredi/OrionXP.html
- The Open AFS Project http://www.openafs.org
- Kerberos – the Network Authentication Protocol
http://web.mit.edu/kerberos/www/
- WAKE – the Windows-AFS-Kerberos Enabler
http://www.rose-hulman.edu/TSC/software/wake/
- Kerberos Leveraged PKI http://www.citi.umich.edu/projects/kerb_pki/
- The Integration of Kerberos V5, AFS, and Windows XP using the AFSLogonShell http://www.coe.uncc/edu/řmdyer/krblogon.htm

[6] odhlédneme-li od možnosti ukládat je ve formátu CSV, který by většina uživatelů zřejmě nedokázala pochopit, natož akceptovat.

# Wireless security

**Petr Cahyna**

E-mail: caha@sunsec.cz

**Abstrakt**

*Pracovní název přednášky je „Wireless security“ a obsahuje bezpečnostní infrastrukturu 802.11 sítě (wifi), její slabiny, jakými prostředky lze riziko minimalizovat, (ne)vhodnost použití tohoto typu sítě, typy možných ataků a jejich následky.*

Příspěvek nebyl dodán.

# WiFi, zkušenosti z projektu pokrytí areálu ZČU a projektu Eduroam

**Jaroslav Čížek**

E-mail: cizek@civ.zcu.cz

**Abstrakt**

*Příspěvek se zaměřuje na praktické zkušenosti s instalací a provozem rozsáhlé bezdrátové sítě Západočeské univerzity v Plzni. Od návrhu topologie a s tím souvisejícím výběrem přístupových bodů a antén se přes různé autentizační mechanizmy dostaneme například až k logování přístupu uživatelů nebo dohledávání útoků vedených z a do bezdrátové sítě. Zmíníme se o i projektu Eduroam, který zajišťuje IP mobilitu a roaming v rámci české sítě národního výzkumu a vzdělávání.*

## 1 Úvod

Vzrůstající počet notebooků a množící se požadavky na mobilitu po univerzitním areálu nás donutily přemýšlet o výstavbě rozsáhlejší bezdrátové WiFi sítě. Za přispění Fondu rozvoje CESNETu a účelové dotace MŠMT se naše úvahy staly skutečností a v těchto dnech dokončujeme síť čítající cca 70 přístupových bodů (Access Point – AP) pro koncové uživatele a 8 serverů zajišťujících autentizaci a ukončení VPN tunelů. Vzhledem k rozmanitosti a principům univerzitního prostředí jsme byli nuceni hledat ne vždy běžné způsoby řešení a hlavně používat v co největší míře open source nástroje.

## 2 Fyzická infrastruktura

### Použitá zařízení

Stavebními prvky nově budované bezdrátové sítě byly přístupové body firmy Cisco Systems, typů Aironet 1231G, 1232AG a 1131AG pracující v pásmu 2,4 resp. 2,4 a 5 GHz a podporující standardy 802.11 (a)/b/g. K řadě 1200 je možné připojit externí anténu – v našem případě to v závislosti na tvaru pokrytého prostoru byly různé typy všesměrových nebo sektorových antén Cisco Aironet

Obr. 1 Přístupové body Cisco Aironet

se ziskem 5,2 až 9 dBi. Radia samozřejmě mají pro každé pásmo (2,4/5 GHz) vyvedeny anténní konektory zvlášť a nutností je i použití odlišných antén.

Kromě pro Cisco typických důležitých vlastností, jako je například možnost správy přes TELNET/SSH a TFTP, je velkou výhodou těchto AP podpora VLAN i v bezdrátové části sítě. To znamená, že jste schopni jedním AP obsloužit různé druhy uživatelů – při připojování si lze vybrat z obecných SSID *eduroam*/*eduroam-simple* a nebo uživatel může zvolit přímo síť svojí katedry nebo útvaru. Z bezpečnostních důvodů jsou v celé síti avizovány pouze SSID *eduroam* a *eduroam-simple*, ostatní SSID jsou neviditelné.

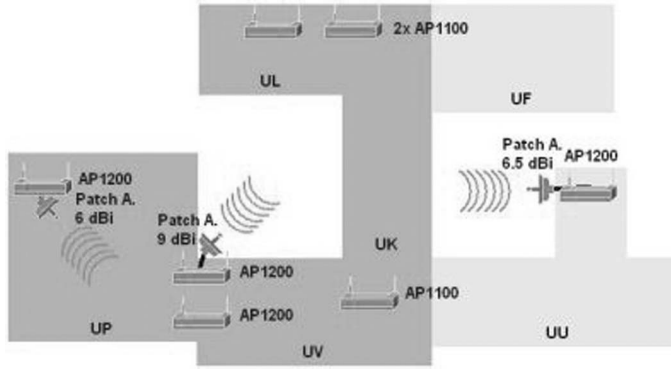
V případě, že AP umístíte na volně přístupných místech, například na chodbách nebo přímo v učebnách, zjistíte, že se neobejdete bez fyzického zabezpečení. Příjemným překvapením při dodávce prvního AP od Cisca proto byl kovový úchyt na stěnu s očkem pro malý visací zámek. Kromě zcizení zařízení je tímto způsobem zabráněno i vytažení ethernetového či napájecího kabelu. V některých speciálních případech jsme navíc využili i elektronického zabezpečení magnetickým senzorem.

### Topologie

Vzhledem k rozsahu požadovaného pokrytí signálem a stavebnímu řešení univerzitních budov v hlavním areálu na Borech (železobeton, kovové podhledy), jsme se od začátku zabývali myšlenkou využít externích všesměrových nebo sektorových antén, které by nám umožnily zvýšit dosah a ozářit budovy kateder a jejich laboratoří „zvenku“, tzn. signál by přicházel okny budovy. Realizace se povedla pouze částečně. Budovy UK,UL a UU, kde sídlí technicky zaměřená Fakulta aplikovaných věd a Fakulta strojní, jsme tímto způsobem dokázali pokrýt bez sebemenších problémů. Naopak nově dokončená budova Fakulty elektrotechnické je pro tento způsob naprosto nevhodná, neboť její okna signál odrážejí.

V současné době se již význam externího pokrytí částečně snížil, protože z důvodu zatížení byly doinstalovány další interní AP. Přesto je však stále na vnějších AP několik desítek přístupů denně, neboť je využívají uživatelé pro připojení z okrajových částí budov a navazujících chodeb, kam signál interních AP nedosáhne.

Pro pokrytí velkých přednáškových sálů je stejně jako v případě externího pokrytí použito AP řady 1200 v kombinaci s externími anténami – v některých případech sektorovou, jindy všesměrovou tyčovou anténou. Přístupové body řady 1100 jsou pak použity na pokrytí zbývajících prostor, kde se ve větší míře studenti nebo zaměstnanci pohybují – ostatní katedry, laboratoře, chodby, prostory kolem bufetů, odpočinkové místnosti atd.




Obr. 2 Příklad externího pokrytí areálu ZČU

# 3 Autentizace, autorizace a účtování přístupů klientů

**Autentizace**

Pro ověření uživatelů sítě WEBnet je na ZČU používán Kerberos v5. Využití tohoto druhu autentizace ale není v případě 802.1x/EAP možné, a bylo tedy nutné hledat jiné způsoby. Po dlouhé diskuzi jsme se shodli na následujícím schématu:

- PEAP – ověření jménem a heslem, určeno pro všechny studenty a zaměstnance, v případě ZČU je použito „síťového hesla“, tzn. neověřuje se proti Kerberos databázi. Doba platnosti tohoto hesla je 1 měsíc.
- EAP-TLS – tento způsob bezpečné autentizace, kde se k ověření používají certifikáty, je určen pro ty zaměstnance univerzity, jejichž notebooky jsou v naší správě a je třeba je předem připravit na delší dobu funkčnosti.
- NoCat – www ověření uživatele na bázi https, které je na ZČU už delší dobu v provozu. Určeno pro uživatele, kteří si nedokáží nakonfigurovat bezpeč-

nější metodu PEAP a nebo jejich HW nebo OS nepodporuje 802.1x/EAP autentizaci. K ověření se opět použije jméno a „síťové heslo“. Vyžaduje prohlížeč s povolenými pop-up okny.

Přihlášení jednou z prvních dvou metod sebou kromě bezpečného ověření uživatele přináší i šifrovaný přenos dat. Kromě původního zabezpečení dynamickým WEP klíčem je k dispozici také WPA a WPA2, tzn. změna klíče s každým paketem a kontrola integrity přenesených dat. Způsob, jak je uvedenou variabilitu připojení možné nastavit na Cisco AP, najdete v kapitole Příklady konfigurací na konci příspěvku.

Naše prvotní představy se však od výše uvedeného schématu částečně lišily – více než rok (březen 2004 až srpen 2005) jsme například pro ověřování studentů testovali z bezpečnostního hlediska velice zajímavý dvoufaktorový mechanizmus autentizace – v případě, že se student chtěl připojit do počítačové sítě se svým notebookem (přes WiFi nebo pevným ethernetovým kabelem), musel nejdříve prokázat svoji identitu přiložením průkazu JIS (Jednotný Identifikační Systém) studenta ZČU ke snímači těchto karet a až poté měl cca 5 minut na přihlášení přes NoCat. Důvodem, proč jsme toto řešení nakonec opustili, je změna naší strategie – z původně plánovaných několika hot-spotů jsme se dostali k plošnému pokrytí celé univerzity a tím pádem problému s rozmístěním a počtem JIS snímačů.

### Autorizace

Jak již bylo uvedeno, kromě bezdrátových sítí s SSID *eduroam* a *eduroam-simple*, do kterých mají přístup všichni ověření uživatelé, provozujeme i sítě kateder a univerzitních útvarů, například síť Laboratoře počítačových systémů s SSID *wlan-lps-1*. Přístup do těchto speciálních sítí je povolen nebo odepřen na základě autorizačních údajů, které přístupovému bodu poskytne prostřednictvím radius serveru MySQL databáze. Přístupovým bodům firmy Cisco je totiž možné zaslat seznam pro uživatele povolených SSID.

Další možností, kterou můžeme ve fázi autorizace udělat, je přiřadit uživatele do jedné konkrétní VLAN, a to nezávisle na použitém SSID. Tento způsob je však v nejnovějších verzích IOSu (operační systém Cisco AP) trochu komplikovaný a klade nároky i na očíslování VLAN v celé metropolitní síti, a proto se mu zatím raději vyhýbáme.

### Účtování

V univerzitní síti samozřejmě neprovádíme účtování ve smyslu výběru peněžních prostředků od připojených studentů, ale pouze ukládáme základní informace, které jsou nutné ke zpětnému dohledání viníka v případě stížností nebo jiných

problémů. K uložení záznamů o startu a ukončení připojení používáme již zmíněnou MySQL databázi, pro ukládání ostatních logů z AP a ostatních služeb (NoCat, DHCP, firewally atd.) používáme syslog server.

Dohledání útoku vedeného z nebo do bezdrátové sítě je i přes všechny uložené informace poměrně problematické a ne vždy průkazné. Vazbou mezi start/stop záznamy z AP a logy z DHCP, které obsahují IP adresy, je MAC adresa. Při ověření přes NoCat, který na své bráně vytváří pravidla jak pro IP, tak i MAC adresu, je riziko zneužití téměř nulové, ale v případě ověření přes 802.1X/EAP může uživatel za jistých okolností vazbu IP adresa – MAC adresa porušit. Tomu je sice možné předejít například zapnutím speciálních funkcí Cisco přepínačů (DHCP snooping a IP source guard), ale my je zatím nevyužíváme, protože většina útoků je nechtěná (např. způsobená viry) a původce lze tedy jednoduše dohledat.

### Technické řešení

Základem serverové části naší AAA infrastruktury je radius daemon distribuovaný pod GNU GPL licencí – freeRADIUS [3] doplněný o MySQL databázi, která obsahuje všechny potřebné autentizační i autorizační údaje. Kromě primárního serveru radius.zcu.cz, který běží na vyhrazeném zařízení, provozujeme ještě identickou horkou zálohu radius2.zcu.cz, což je virtuální stroj na univerzitním XEN serveru.

Přístup s ověřením přes www zajišťuje několik NoCat [4] bran s jedním nadřazeným NoCat autentizačním serverem, který brány řídí. NoCat brána funguje jako směrovač s filtrovacími pravidly. Pokud se připojí nový uživatel a otevře www prohlížeč, je NoCat bránou přesměrován na http přihlašovací formulář na autentizačním serveru. Po úspěšném ověření je na NoCat bráně vytvořeno dočasné pravidlo, které propouští pakety/rámce na základě IP a MAC adresy. Pravidlo je platné do chvíle, kdy se uživatel odhlásí nebo zavře periodicky obnovované https pop-up okno www prohlížeče.

### Administrace uživatelů

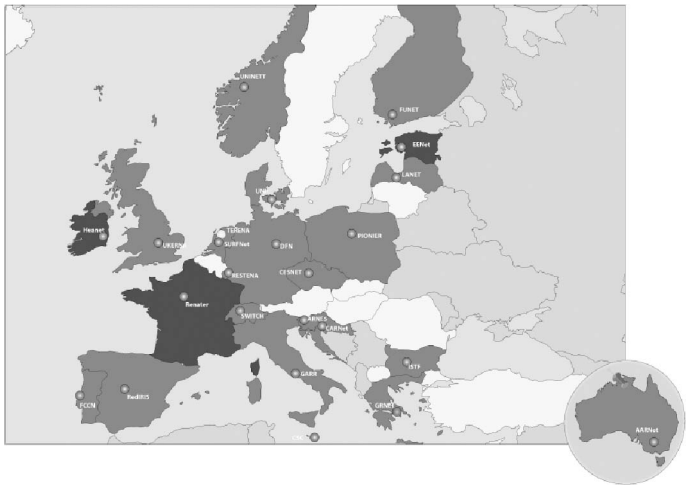
Jak již bylo uvedeno, všechny údaje potřebné pro ověření klientů jsou uloženy v MySQL databázi. Seznam uživatelů, včetně jejich příslušnosti ke katedře nebo útvaru, je jednou za 24 hodin synchronizován proti univerzitním LDAP serverům. Každé dvě hodiny se z důvodu redundance také exportuje obsah celé MySQL databáze z primárního serveru na sekundární záložní server.

Pro administrátory sítě jsme vytvořili jednoduchou aplikaci pro dohled nad AAI infrastrukturou. Jednoduše lze získat údaje o aktuálně připojených uživatelích, zobrazit historii přihlašování nebo, pokud jsou porušena pravidla, zakázat konkrétnímu uživateli přihlášení.

Má-li uživatel v úmyslu připojit se do sítě, musí pouze aktivovat svůj účet tím, že si přes www rozhraní nastaví síťové přístupové heslo. Na rozdíl od univerzitních přihlašovacích údajů v Kerberos databázi, které jsou platné po dobu celého školního roku, je nutné účet pro mobilní připojení obnovovat každý měsíc. Při tomto procesu je uživateli vypsán seznam jeho připojení v minulém měsíci a uživatel je musí odsouhlasit. V případě, že z předložených údajů zjistí, že se v danou dobu nepřipojoval, je navíc povinen změnit si heslo.

# 4 Projekt Eduroam

V roce 2003 vznikla pod hlavičkou asociace TERENA [6] skupina „Task Force on Mobility“, jejímž cílem bylo zmapovat v té době existující způsoby zabezpečení bezdrátových sítí a navrhnout mechanizmus, který by uživatelům akademických a výzkumných sítí umožnil IP mobilitu a roaming jejich mobilních zařízení v rámci zúčastněných sítí. Navržené řešení bylo otestováno a postupně se rozšířilo po téměř celé Evropě (cca 350 sítí v 18 zemích) a dokonce i do Austrálie. Vzniklá AA(A) infrastruktura byla pojmenována Eduroam [7].




Obr. 3 Účastníci projektu Eduroam (září 2005)

**Eduroam v ČR**

Již od samého vzniku aktivity Eduroam se na všech pracích podíleli také zástupci sdružení CESNET, z. s. p. o., které provozuje českou síť národního výzkumu.

CESNET na sebe vzal i roli koordinátora a provozovatele kořenových RADIUS serverů pro ČR. V současnosti je do projektu zapojeno 7 nejvýznamnějších vysokých škol, kromě ZČU, která se do hierarchie Eduroam připojila v zimě 2004, je to například Karlova univerzita, ČVUT nebo Technická univerzita v Liberci.

### Scénáře připojení

Základní motivací pro vznik projektu byla myšlenka umožnit transparentní přihlášení mobilního uživatele kdekoli v síti Eduroam pomocí vlastní identity ze své domovské organizace, tzn. bez nutnosti žádat o lokální přihlašovací údaje v navštívené organizaci. Pro tento účel byly navrženy tři obecné typy připojení:

- www ověření pomocí http protokolu – uživatel je při spuštění www prohlížeče přesměrován na formulář, kde vyplní přihlašovací údaje (jednoduché, ale značně nebezpečné)
- 802.1X autentizace – o přihlášení se stará speciální klientský software (tzv. suplikant), kromě přihlášení jménem a heslem je možné použít např. certifikáty (velmi bezpečné, trochu složitější)
- VPN autentizace/přístup do domácí sítě – uživatel se tunelem připojí do své domácí sítě a z ní dostane také přidělenu IP adresu, pod kterou pak v Internetu vystupuje.

Pro předávání uživatelské identity pro první dva způsoby (www a 802.1X autentizace) byla vybudována hierarchická autentizační a autorizační infrastrukturu (AAI), kterou v síti Eduroamu tvoří vzájemně propojené RADIUS servery. Příslušnost ke své domovské organizaci dá uživatel roamingu na vědomí připojením realmu (identifikátor/doména organizace, např.: `cizek@ZCU.CZ`), který pak zajistí správné směrování požadavku ověření v AAI.

V případě VPN přístupu není AAI třeba, neboť se uživatel ověřuje přímo proti své VPN bráně. Problém je však s tím, jak konektivitu omezit pouze na důvěryhodné VPN servery v sítích projektu a zamezit tak připojení libovolného uživatele, který má konto na nějakém VPN serveru v Internetu. V ČR jsme tuto situaci dočasně vyřešili tak, že zúčastněným organizacím byly pro účely VPN přiděleny IP subsítě z rozsahu 195.113.214.0/24.

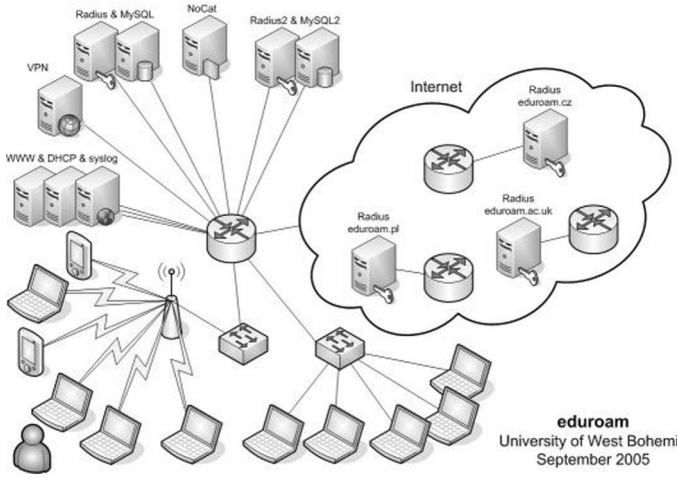
### Politika připojení, účtování

Pokud má cizí uživatel využívat hostitelskou síť, a nebo dokonce vystupovat pod její IP adresou, je třeba definovat striktní pravidla nejen pro samotné uživatele, ale i práva a povinnosti provozovatelů domácí a hostující sítě, způsoby řešení bezpečnostních incidentů nebo sankce, které lze použít proti provinilcům. Příklad roamingové politiky vydané sdružením CESNET viz [8].

# 5 Přístupová síť pro mobilní uživatele

Na ZČU provozovanou přístupovou síť pro mobilní zařízení tvoří kromě cca 70 AP a 5 přepínačů Cisco Catalyst 2950T také celkem 8 serverů, všechny s OS Linux (distribuce Debian):

- 2× freeRadius + MySQL
- 1× NoCat autentizační server
- 3× NoCat gateway
- 1× OpenVPN
- 1× www, dhcp, syslog (fyzicky různé univerzitní servery)




Obr. 4 Přístupová síť pro mobilní uživatele

Podmínkou připojení na straně mobilního zařízení je WiFi certifikovaná bezdrátová karta podporující alespoň jeden ze standardů 802.11 a/b/g a buď www prohlížeč s povolenými pop-up okny, nebo klient s možností 802.1x/EAP ověření. Na stránkách uživatelské podpory a projektu eduroam na ZČU [9] jsou k dispozici návody na připojení z MS Windows XP, Linuxu a nově i popis připojení pro PDA zařízení s Windows Mobile 2003 SE (pouze PEAP a EAP-TLS).

Kromě bezdrátového WiFi přístupu mají uživatelé nejen z řad studentů a zaměstnanců, ale i externistů z ostatních eduroam sítí k dispozici pevné připojení na vyhrazených pracovištích – v současnosti je to cca 30 přípojných míst ve 4 lokalitách (budovy IC, EP, HJ a KL). Pro přístup je použit stejný princip jako v případě bezdrátové sítě, tzn. pokud se uživatel nedokáže ověřit přes EAP, může se autentizovat pomocí www nebo spustit VPN tunel do domovské organizace. Ukázku pro tento účel použité konfigurace dvou VLAN na jednom portu najdete v poslední kapitole.

Pro vzdálený přístup z Internetu, ať už se uživatel nachází v některé ze sítí eduroam nebo kdekoliv jinde, mohou všichni zaměstnanci a studenti naší univerzity využívat OpenVPN server. K dispozici je opět buď obecný přístup do společné sítě, zabezpečený univerzitním jménem a heslem, nebo speciální přístup do katedrálních sítí, kde se pro přihlášení uplatní osobní certifikáty.

# 6 Zkušenosti z provozu

I když je bezdrátová infrastruktura ZČU stále ve zkušebním provozu, využívají jí už nyní desítky uživatelů. Po předání do ostrého provozu (plánováno začátkem října) a spuštění propagační kampaně se dle našich předpokladů vytíženost tohoto „pohodlného“ způsobu připojení ještě výrazně zvýší.

Celkově je spolehlivost nově postavené sítě poměrně vysoká a problémy, které nejčastěji řešíme na naší straně, se většinou týkají pouze specifických konfiguračních detailů, které jsme při instalaci nových AP opomenuli. Na straně uživatelů jsou naopak běžné případy špatně nainstalovaných a nakonfigurovaných ovladačů, nepredikovatelně se chovajících levných WiFi zařízení atd., které lze jen obtížně řešit.

Mezi další nepříjemnosti, s kterými se občas potýkáme, patří:

- studenti používají převážně www ověření, velká část z nich dokonce nemá ani naimportovaný certifikát certifikační autority, čímž se vystavuje nebezpečí nežádoucího odposlechu hesla (částečně řeší nastavení portů do stavu *switchport protected*, kdy je zakázána komunikace mezi připojenými uživateli)

- při konfiguraci protokolu PEAP v případě nativního klienta MS Windows XP zapomínají uživatelé vypnout automatické použití přihlašovacího jména do domény

- studentské notebooky jsou často zavirované, což nás donutilo zakázat komunikaci na nejvíce zneužívaných udp a tcp portech a omezit smtp

- při použití v rámci sítě eduroam je problém s radius realmem, protože ho při konfiguraci uživatelé většinou nezadají a pak už je oprava komplikovaná.

# 7 Závěr

Shrnu-li naše poznatky a výsledky práce, povedlo se nám vytvořit funkční síť s rozumně nastavenou mírou bezpečnosti, která umožní plošné bezdrátové připojení

všem oprávněným uživatelům nejen ze ZČU, ale i z ostatních sítí sdružených v projektu Eduroam.

Příspěvek se sice zabýval konkrétní bezdrátovou sítí se všemi jejími specifiky a problémy, přesto si ale myslíme, že může posloužit jako vodítko pro stavbu libovolné jiné velké bezdrátové sítě, ať už v komerční nebo akademické sféře.

# 8 Příklady konfigurací

Ukázka části konfigurace několika SSID na přístupovém bodu Cisco Aironet 1100, IOS verze 12.3(4):

```
aaa authentication login eap_methods group rad_eap
dot11 mbssid
dot11 ssid eduroam
   vlan 181
   authentication open eap eap_methods
   authentication network-eap eap_methods
   authentication key-management wpa optional
   accounting acct_methods
   mbssid guest-mode
!
dot11 ssid eduroam-simple
   vlan 180
   authentication open
   accounting acct_methods
   mbssid guest-mode

dot11 ssid wlan-civ-101
   vlan 101
   authentication open eap eap_methods
   authentication network-eap eap_methods
   authentication key-management wpa optional
   accounting acct_methods

interface Dot11Radio0
   encryption vlan 181 mode ciphers aes-ccm tkip wep128
   encryption vlan 101 mode ciphers aes-ccm tkip wep128
   ssid eduroam
   ssid eduroam-simple
   ssid wlan-civ-101
```

Ukázka části konfigurace 802.1X/guest vlan portu na přepínači Cisco Cat 2950T, IOS 12.1(22)EA5:

```
aaa authentication dot1x default group rad_eap
aaa authorization network default group rad_eap
dot1x system-auth-control
dot1x guest-vlan supplicant

interface FastEthernet0/1
 description dot1x eduroam/nocat (vlan 181/180)
 switchport access vlan 181
 switchport mode access
 switchport protected
 dot1x port-control auto
 dot1x timeout quiet-period 10
 dot1x timeout tx-period 5
 dot1x guest-vlan 180
 dot1x reauthentication
 spanning-tree portfast
```

# Literatura

[1] *IEEE 802 LAN/MAN Standards Committee*, Port Based Network Access Control, http://www.ieee802.org/1/pages/802.1x.html

[2] *PPP Extensible Authentication Protocol (EAP)*, http://www.ietf.org/rfc/rfc2284.txt

[3] *freeRADIUS, GNU GPL server radius*, http://www.freeradius.org/

[4] *www ověření NoCat*, http://nocat.net/

[5] *WPA klient pro Linux*, http://hostap.epitest.fi/wpa_supplicant/

[6] *TERENA – Trans-European Research and Education Networking Association*, TF-Mobility, http://www.terena.nl/tech/index_mobility.html

[7] *Eduroam – Education Roaming*, http://www.eduroam.org, http://www.eduroam.cz

[8] *Roamingová politika projektu Eduroam*, http://www.eduroam.cz/cz/main/politika.html

[9] *Uživatelská podpora a Eduroam na ZČU*, http://support.zcu.cz, http://eduroam.zcu.cz

# Single Sign-On řešení pro webové aplikace

**Petr Grolmus, Michal Švamberg**

E-mail: grolmus@civ.zcu.cz, svamberg@civ.zcu.cz

**Klíčová slova:** Single Sign-on, SSO, WebAuth, PubCookie, CoSign, autentizovaná webová proxy, identity federation, service provider, identity provider

**Abstrakt**

*Cílem tohoto příspěvku je seznámit čtenáře s možným řešením zavedení jednotné autentizace ve webovém prostředí tak, aby uživatel byl ověřen pouze při přístupu k první aplikaci. Jeho identita je pak předána všem ostatním aplikacím, které uživatel v průběhu času použije. Tento způsob ověření uživatele a sdílení jeho identity množinou webových aplikací je běžně nazýváno jako Single Sign-On (SSO).*

**Abstract**

*This paper presents a possible solution to the problem of web-based authentication that would require the user to enter his/her credentials just once – ie. when accessing any of the pool of applications for the first time. The user's identity is then automatically communicated to other applications the user tries to use. This authentication method is usually referred to as Single Sign-On (SSO).*

## 1 Elektronická identita

Každý jednotlivec žijící v moderní společnosti je identifikovatelný velkým množstvím atributů, jakým mohou být například rodné číslo, číslo občanského průkazu, cestovního pasu, řidičského průkazu, číslo kreditní karty, bankovních účtů a mnohými dalšími. Elektronickou identitu jednotlivce lze však chápat v mnohem větším měřítku. K dříve uvedeným je možné přidat e-mailové adresy, jména a hesla k aplikacím a systémům, diář plánovaných schůzek, seznam kontaktů,

charakteristické profily zájmů, ale také i elektronicky objednané obědy z firemního jídelníčku. Přidáme-li k tomuto výčtu spojeného nejčastěji s výkonem zaměstnání další uživatelovy „otisky“ při putování internetem (účty veřejných poskytovatelů elektronické pošty, účty v rámci elektronických obchodů, registrace při stažení demoverzí softwarových produktů, identitu v diskuzních fórech, ...), dostaneme poměrně pěknou džungli plnou jmen, hesel a různých kódů.

Naštěstí doba, kdy každá nová aplikace si s sebou přinášela vlastní (samozřejmě nejlepší :-) způsob ověření uživatelů a řízení přístupových práv, se pomalu stává minulostí. Stále více se daří přesvědčovat dodavatele nových systémů, aby využívali k ověření uživatelů autentizačních prostředků již ve firmách zavedených. V rámci společností se nyní směřuje k tomu, aby uživatel byl uvnitř rámci organizace identifikován ve výpočetním systému, pokud možno, pouze jediným účtem (jménem/heslem, osobním certifikátem, apod.). Nejinak tomu je i na Západočeské univerzitě v Plzni (ZČU), kde si již před mnoha lety naši vizionáři uvědomili tento trend. Jako krok správným směrem se zpětně jeví nasazení systému Kerberos pro ověřování uživatelů a zdrojů ve výpočetním prostředí ZČU.

## 2 Prostředí webu na ZČU v minulosti

V případě webových aplikací zde panovala velká nejednotnost. Velká množina aplikací používala vlastní způsoby ověření uživatelů, nejčastěji proti nějakému vnitřnímu seznamu uživatelů. Není snad ani třeba zdůrazňovat potíže spojené s aktualizací těchto seznamů. Poměrně homogenní prostředí použité pro WWW servery (kombinace Linuxu a WWW serveru Apache) umožnilo v pozdější době navázání nových webových aplikací na autentizační modul *mod_auth_kerb*. Toto řešení s sebou přineslo několik výhod i nevýhod. Neoddiskutovatelnou výhodou je používání jediného jména/hesla shodného s ověřením uživatele ve výpočetním prostředí *Orion*[1]. Velkou nevýhodou tohoto řešení však bylo, že uživatel musel toto heslo vyplňovat do každé webové aplikace zvlášť. To je nejenom znepříjemňující fakt, ale také se zvyšuje pravděpodobnost odezírnutí hesla někým dalším, při jeho častém zadávání.

Se stále rostoucím počtem webových aplikací se dosud používaný ověřovací modul *mod_auth_kerb* postupně stával nedostačujícím. Nastal čas porozhlédnout se po novém řešení, které bude umět předávat identitu již jednou ověřeného uživatele na další weby bez nutnosti opakovaného zadávání přístupového hesla. Možných řešení bylo hned několik. Do detailnějšího zkoumání postoupily tři produkty: WebAuth[2], CoSign a PubCookie. Všechna řešení mají velmi podobný způsob nakládání s ověřením uživatele a přesměrováváním požadavků mezi apli-

[1] *Orion* je název distribuovaného výpočetního prostředí na ZČU.

[2] Pozor! Existují dvě řešení webového SSO stejného jména z různých zdrojů. Zde jde o verzi vyvíjenou na univerzitě ve Stanfordu.

kačními servery (stroje poskytující webové aplikace) a tzv. login-serverem (stroj, na kterém je ověřena identita přistupujícího uživatele).

Jako první z výběru vypadl PubCookie – řešení, které není primárně závislé na žádné autentizační autoritě. Z tohoto pohledu je poměrně modulární a zcela jistě by mohlo vyhovovat pro velkou množinu případů. Jeho návrh je poměrně jednoduchý, ale neumožňoval vzájemné ověření aplikačního serveru a ověřovacího serveru. Také neexistovala možnost, aby v případě použití systému Kerberos jako autentizační autority, získal aplikační server identitu přistupujícího uživatele (tj. zaslat uživatelský krb ticket z login serveru na aplikační server).

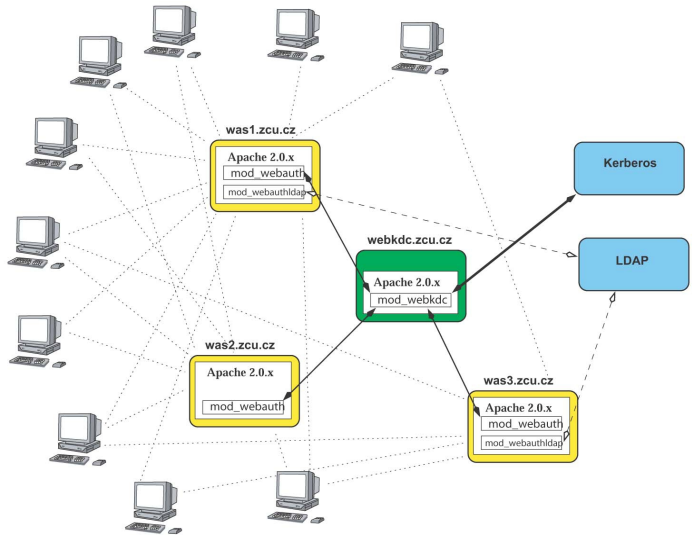
Další řešení CoSign je oproti PubCookie robustnější. Obsahuje seznam spravovaných aplikací, které smí požádat o identitu uživatele, je připravené na ověření proti systému Kerberos, umí předávat krb ticket uživatele na aplikační server. Toto řešení má ale i své nevýhody, z nichž patrně největší je neustálé ověřování na login-serveru při každém přístupu k nějaké stránce webové aplikace, zda je uživatel přihlášen. V případě několika hojně využívaných aplikací se může tento fakt negativně podepsat na odezvách login-serveru. Navíc, v době výběru vhodného SSO řešení pro naše potřeby, bylo nutné pro použití systému CoSign provádět netriviální úpravy webových aplikací.

Poslední z užšího výběru je řešení s názvem WebAuth, které je vyvíjené na univerzitě ve Stanfordu. Již na první pohled je přímo šité na míru autentizačnímu systému Kerberos. Výměna a potvrzení informací, nebo chcete-li protokol, mezi jednotlivými subjekty (uživatelský browser, aplikační server a login-server) ne náhodou připomíná samotný protokol systému Kerberos. Každý z aplikačních serverů je v systému identifikován vlastním krb principalem, vyměňované informace mezi login-serverem a aplikačním serverem jsou v cookie kryptována společným klíčem relace. Informace v cookie jsou platné vždy maximálně na dobu platnosti uživatelského krb ticketu (získaného login-serverem při ověření uživatele) a není tedy třeba neustálého ověřování platnosti přihlášení. WebAuth podporuje také server-pool přihlašovacích webů, kdy dochází k load-balancingu zátěže na ověřovací login-servery a v případě výpadku jednoho z nich přejímají jeho zátěž servery zbývající. Kromě jiného také umí aplikačnímu serveru zaslat uživatelský krb ticket, pro případ, že by aplikace vyžadovala pro nějaký úkon přímo identitu daného uživatele.

# 3 SSO řešení WebAuth

Jak bylo již dříve zmíněno, je WebAuth založen na autentizačním systému Kerberos. Celý systém WebAuth je pak tvořen třemi spolupracujícími moduly do WWW serveru Apache[3]: *mod_webauth*, *mod_webauthldap* a *mod_webkdc*.

[3]Moduly jsou psány pro verzi Apache 2.0.43 a vyšší.




Obr. 1 Prostředí SSO sytému WebAuth

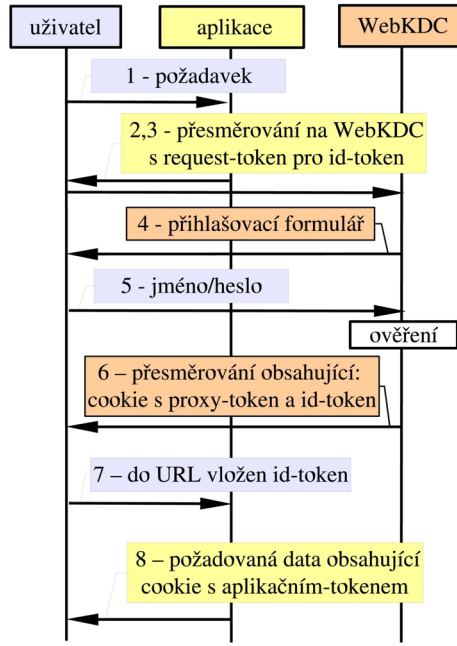
Srdcem celého systému je login-server běžně nazývaný WebKDC (KDC je převzato z názvosloví systému Kerberos, kde KDC je zkratka pro Key Distribution Center – centrum výdeje krb ticketů). Na WebKDC a pouze na něm běží jeden z výše uvedených modulů: *mod_webkdc*. Jeho úkolem je přejímat požadavky od aplikačních serverů, zpracovávat je a ověřovat identitu přistupujícího uživatele u autentizační autority.

Zbylé dva moduly *mod_webauth* a *mod_webauthldap* jsou umístěny na aplikačním serveru – WWW serveru Apache, který poskytuje stránky chráněné systémem WebAuth. První z modulů zajišťuje ověření dosud neznámého uživatele přesměrováváním na WebKDC server a přejímá identitu uživatele z jím zaslaného cookie, do kterého si aplikační server dříve tuto identitu již ověřeného uživatele uložil. Je tedy možné říci, že se stará resp. vyžaduje autentizaci uživatele.

Druhý modul, *mod_webauthldap*, provádí autorizační službu. Pomocí něj je možné definovat seznam povolených skupin uživatelů, které jsou spravovány adresářovou službou LDAP. Modul *mod_webauth* musí být na aplikačním serveru vždy, pomocí něj je možné v konfiguračním souboru WWW serveru Apache povolit všechny ověřené uživatele nebo jejich jmenovitý výčet. Modul *mod_webauthldap* je pro aplikační server volitelný. Prostředí založené na webovém SSO řešení WebAuth může tedy vypadat například tak, jak je znázorněno na obrázku 1 na straně 90.

Zpracování přístupu dosud neověřeného uživatele v SSO systému se účastní uživatel (resp. jeho WWW prohlížeč), aplikační server (webová aplikace), login-

server (WebKDC) a autentizační autorita (např. Kerberos). Je nutné předeslat, že pro správnou funkci je vyžadované kryptované (https) spojení jak mezi uživatelem a aplikačním serverem, tak i mezi uživatelem a login-serverem. Časová souslednost jednotlivých akcí nutných pro úspěšné ověření uživatele do webové aplikace je patrná z obrázku 2 na straně 91.




Obr. 2 Prvotní přihlášení uživatele k SSO

1. Neověřený uživatel přistupuje k webové aplikaci chráněné WebAuthem.

2. *mod_webauth* detekuje, že uživatel dosud nevlastní aplikační token (neobdrží od něj aplikační cookie) a vytvoří tzv. request-token pro id-token. Request-token obsahuje informace jako jsou návratové (resp. původně dotazované) URL, požadovaný typ tokenu, atp. Request-token je zakryptován použitím AES session-klíčem sdíleným mezi aplikačním serverem a WebKDC (login-server) získaným z webkdc-service-tokenu. *mod_webauth* pak vytvoří redirekt na WebKDC, jenž obsahuje request-token v parametrech URL.

3. Redirekt způsobí přesměrování uživatelova prohlížeče na WebKDC spolu s vygenerovaným request-tokenem. Žádné cookie není zasláno na WebKDC (zatím žádné uživatel nemá).

4. WebKDC následně rozkryptuje request-token. Zkontroluje čas vytvoření, za účelem ověření, zda je dostatečně „čerstvý“ a pošle zpět uživatelskému prohlížeči přihlašovací formulář. Request-token je uložen ve skryté položce tohoto formuláře.

5. Uživatel zadá své přihlašovací jméno a heslo a odešle data formuláře zpět ke zpracování na WebKDC.

6. WebKDC ověří zadané jméno a heslo a také skutečnost, zda aplikační server, který požaduje ověření uživatele má povolení vyžadovat id-token. Předpokládejme, že přihlašovací jméno a heslo jsou správná, pak WebKDC vytvoří cookie, do kterého uloží proxy-token a id-token (obsah cookie je kryptován privátním AES-klíčem WebKDC). Stránka s potvrzením, že ověření proběhlo v pořádku, je následně zaslána do uživatelského prohlížeče obsahující odkaz na původně požadovanou stránku.

7. Uživatelský prohlížeč znovu přistoupí na původně požadovanou stránku a v URL parametrech je předán také id-token (identita) uživatele.

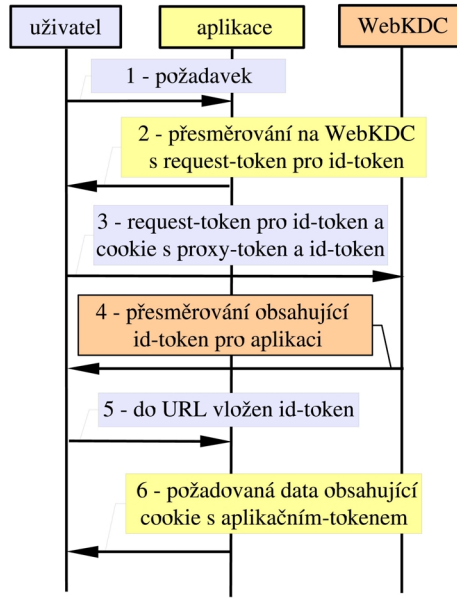
8. *mod_webauth* si z požadavku převezme id-token a následně zkontroluje, zda je „čerstvý“. Pokud je vše v pořádku, pak přepíše id-token na aplikační-token a uloží jej do cookie pro další použití. Nakonec je token odstraněn z URL (již není zapotřebí – aplikace věří předkládanému cookie, které je kryptované jejím privátním AES klíčem).

Pokud je uživatel již úspěšně ověřen pomocí WebKDC, pak přístup ke každé další aplikaci probíhá zjednodušeným způsobem – viz obrázek 3.

1. Uživatel přistupuje k webové aplikaci chráněné WebAuthem.

2. *mod_webauth* detekuje, že uživatel dosud nevlastní aplikační token (neobdrží od něj aplikační cookie) a vytvoří tzv. request-token pro id-token. Request-token obsahuje informace jako jsou návratové (resp. původně dotazované) URL, požadovaný typ tokenu, atp. Request-token je zakryptován použitím AES session-klíčem sdíleným mezi aplikačním serverem a WebKDC (login-server) získaným z webkdc-service-tokenu. *mod_webauth* pak vytvoří přesměrování na WebKDC (obsahující request-token v parametrech URL).

3. Redirekt způsobí přesměrování uživatelova prohlížeče na WebKDC spolu s vygenerovaným request-tokenem. Zároveň je na WebKDC server zaslané cookie obsahující proxy-token a id-token, které WebKDC server uživateli vystavil při jeho prvotním ověření.




Obr. 3 Single Sign-On

4. WebKDC server detekuje ze zaslaného cookie, že uživatel má platný proxy-token a použije jej pro vytvoření nového id-tokenu pro aplikační server. WebKDC následně vygeneruje návratové URL, které obsahuje response-token pro aplikační server.

5. Uživatelský prohlížeč znovu požádá o původně zadanou chráněnou stránku a v URL parametru předá aplikačnímu serveru id_token již dříve přihlášeného uživatele.

6. *mod_webauth* si z požadavku převezme id-token a následně zkontroluje, zda je „čerstvý“. Pokud je vše v pořádku, pak přepíše id-token na aplikační-token a uloží jej do cookie pro další použití. Nakonec je token odstraněn z URL (již není zapotřebí – aplikace věří předkládanému cookie, které je kryptované jejím privátním AES klíčem).

## 3.1 Konfigurace WWW serveru Apache na WebKDC

Pro správnou funkci je samozřejmě nutné korektně nakonfigurovat jednotlivé moduly WWW serveru, jak na straně WebKDC, tak na straně jednotlivých aplikačních serverů. V době, kdy jsme začínali experimentovat se systémem WebAuth, bylo v podstatě nutné moduly na jednotlivých strojích ručně překládat. V současné době již vývojáři poskytují kompilované balíky pro linuxovou distribuci

Debian, čímž výrazným způsobem zjednodušili jak používání jejich systému, tak i jeho další šíření.

Jak WebKDC, tak i všechny aplikační servery mají svůj vlastní principal v Kerberos systému. Tento fakt pomáhá zvyšovat bezpečnost celého systému, neboť je možné definovat seznam „povolených“ aplikačních serverů, které smí požádat o identitu přistupujícího uživatele.

Na straně WebKDC je samozřejmě nutné zapnout modul *mod_webkdc*, nainstalovat skripty a šablony, které se účastní ověření uživatele (přihlašovací formulář a jeho zpracování). V neposlední řadě je nutné provést alespoň minimální konfiguraci modulu. Toto nastavení může v konfiguračních souborech WWW serveru Apache vypadat například takto:

```
## WebKDC nastavení
WebKdcServiceTokenLifetime 30d
WebKdcKeyring /etc/apache2/conf/webkdc/keyring
WebKdcKeytab /etc/apache2/conf/webkdc/keytab
WebKdcTokenAcl /etc/apache2/conf/webkdc/token.acl
```

kde soubor `keytab` obsahuje krb klíč služby webkdc – půjde například o klíč krb principalu „*webkdc/webkdc.zcu.cz@ZCU.CZ*“. Soubor `keyring` pak obsahuje soukromý AES klíč webkdc služby. Tento soubor je načten při každém startu nového procesu Apache. V případě, že dosud tento soubor neexistuje, je modulem *mod_webkdc* vytvořen pro další použití. Direktiva `WebKdcServiceTokenLifetime` určuje, s jakou periodou si bude modul z bezpečnostních důvodů měnit tento privátní klíč. Poslední z uvedených souborů `token.acl` obsahuje seznam krb principalů, které se mohou serveru WebKDC dotazovat na identitu uživatele a také seznam tokenů, které mohou vyžadovat. Soubor tedy může mít například tento obsah:

```
krb5:webauth/*.zcu.cz@ZCU.CZ id
```

kde tato řádka definuje, že kterýkoliv aplikační server identifikovaný krb principalem s instancí „webauth“ v doméně *zcu.cz* (`webauth/*.zcu.cz`) smí vyžadovat identifikační token uživatele (`id`). Pokud máte kontrolu nad vytvářením krb principalů pro jednotlivé aplikační servery, pak je tento zápis dostačující. V opačném případě by bylo vhodné vyjmenovávat všechny tyto principaly pro případ jejich možného pozdějšího vyřazení ze seznamu aplikací, které se smějí WebKDC dotazovat na uživatelovu identitu.

## 3.2 Konfigurace WebAuthu na aplikačních serverech

Konfigurace modulů na straně aplikačního serveru je rozdělena na dvě části. První provádí obecné nastavení, jako je definování adresy URL pro ověření dosud neznámého uživatele na WebKDC serveru, soubor s krb klíčem principalu, apod. Definice nastavení pro modul *mod_webauth* může vypadat tedy například takto:

```
## WebAuth nastaveni
WebAuthLoginURL "https://webkdc.zcu.cz/login.fcgi"
WebAuthWebKdcURL "https://webkdc.zcu.cz/webkdc-service/"
WebAuthWebKdcPrincipal webkdc/webkdc
WebAuthKeyring /etc/apache2/keyring
WebAuthKeyringAutoUpdate on
WebAuthKeyringKeyLifetime 30d
WebauthKeytab /etc/apache2/keytab
WebAuthServiceTokenCache /etc/apache2/service_token.cache
```

Většina konfigurace tohoto modulu lze odvodit z názvu jednotlivých direktiv, ale pro pořádek je rozepíšeme. První URL odkazuje na přihlašovací formulář; druhý URL odkaz směřuje na adresu, kam aplikační servery přímo posílají kryptované požadavky; další řádka obsahuje název krb principalu služby WebKDC. Další tři direktivy souvisejí s lokálně uloženým privátním AES klíčem aplikačního serveru. `WebAuthKeyring` sděluje modulu, kde daný klíč leží; direktiva `WebAuthKeyringAutoUpdate` povoluje modulu automatickou změnu tohoto klíče a `WebAuthKeyringKeyLifetime` určuje, jak dlouho může být jednou vygenerovaný klíč používán před další výměnou. Direktiva `WebauthKeytab` sděluje, kde na disku je možné najít klíč krb principalu aplikačního serveru. Poslední direktiva slouží k uložení service-tokenu, který se tak stává použitelným pro všechny nově vytvářené procesy WWW serveru Apache.

V případě, že je na aplikačním serveru použit i modul *mod_webauthldap*, pak další konfigurace se týká tohoto modulu:

```
## WebAuthLDAP nastaveni
WebAuthLdapHost ldap.zcu.cz
WebAuthLdapBase ou=rfc2307,o=zcu,c=cz
WebAuthLdapAuthorizationAttribute cn
WebAuthLdapKeytab /etc/apache2/ldapkeytab webauth/sso.zcu.cz
WebAuthLdapFilter memberUid=USER
WebAuthLdapTktCache /tmp/webauthldap.tkt
```

První direktiva určuje, na kterém serveru bude hledat informace pro autorizaci podle skupin. Další direktiva `WebAuthLdapBase` definuje konkrétní úložiště s informacemi o skupinách. `WebAuthLdapAuthorizationAttribute` určuje, který atribut koresponduje s položkou v LDAPu obsahující skupiny uživatelů. `WebAuthLdapKeytab` obsahuje odkaz na soubor s klíčem krb principalu použitého pro dotaz na LDAP server. `WebAuthLdapFilter` definuje filtr do LDAPu, který určuje, na kterou z položek je mapován login uživatelů. Poslední direktiva určuje Kerberos cache, která pokud obsahuje validní krb ticket, tak je použit, jinak je použit dříve definovaný keytab k získání nového krb ticketu.

Další část konfigurace už se týká přímo chránění přístupu pro definovaný adresář pomocí WebAuth SSO řešení.

```
<Directory /usr/local/data/www/>
    AllowOverride None
    AuthType WebAuth
    Require privgroup lps civ
    Require user indy bike dolf sustr4
</Directory>
```

Důležité jsou poslední tři řádky uvnitř tagu `Directory`. Direktiva `AuthType` definuje, že přístup pro daný adresář je používána služba WebAuth. Další řádek určuje seznam povolených skupin uživatelů (skupiny *lps* a *civ*). Poslední řádek obsahuje explicitní seznam povolených uživatelů, kteří budou autorizováni bez ohledu na příslušnost v některé ze skupin. V případě, že aplikační server neobsahuje modul *mod_webauthldap*, pak není možné přístup omezovat na seznam skupin uživatelů. Je možné povolit přístup jen pro vyjmenované uživatele, případně lze povolit všechny uživatele pomocí direktivy `Require valid-user`.

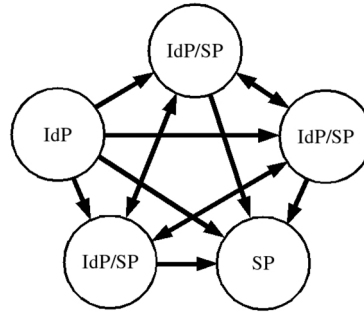
# 4 Identity Federation

V současné době se většina firem otevírá více jak svým zákazníkům, tak i dodavatelům a dává na svém webu prostor pro výměnu informací s těmito subjekty. Vznikají tak v poslední době často diskutované a prezentované systémy: v případě zákazníků *Business-to-Customer* (B2C) a v případě obchodních partnerů *Business-to-Bussiness* (B2B).

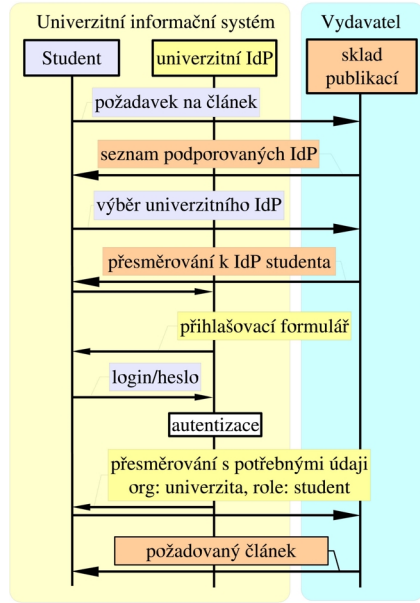
Právě u B2B a B2C, kde dochází k jistému překrytí elektronických systémů dvou (nebo více) různých subjektů, se otevírá prostor pro sdílení elektronických identit svých uživatelů (studentů, vědců, zaměstnanců, ...) mezi spolupracujícími organizacemi v rámci projektů nebo obchodu. Dochází tedy ke sdružování identit – identity federation (IF). Na celou problematiku IF lze v podstatě nahlížet jako na rozkrývání vazeb spolupráce mezi organizacemi s cílem umožnit uživatelům jedné organizace přistupovat ke zdrojům jiné organizace pod „domácí“ identitou uživatele.

Vazbu mezi dvěma organizacemi umožňuje fakt, že jedna z nich nabízí své datové zdroje – je tedy poskytovatelem služby (service provider – SP). Doplňkem popisované vazby je konzument této služby. Konzumentem může být například uživatel jiné organizace, který je ověřen u svého lokálního IM systému – poskytovatele identity (identity provider – IdP). Poměrně častým případem je stav, kdy jedna organizace fungující jako jeden IdP pro své uživatele poskytuje navenek i několik desítek služeb SP. Na obrázku 4 jsou patrné tyto vazby mezi několika různými poskytovateli a konzumenty služeb.

Typickým příkladem identity federation systému může být spolupráce univerzitní knihovny s externími poskytovateli literatury (např. vydavatelé odborné




Obr. 4 Vazby v Identity federation



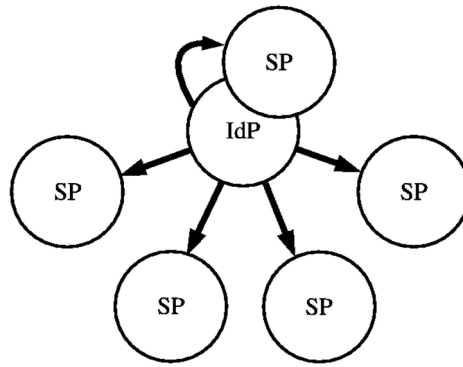

Obr. 5 Typický příklad Identity federation

literatury nebo sborníků konferencí). Za poměrně nemalé finanční prostředky je nakoupen přístup k jejich elektronickým zdrojům. Nyní nastává problém, jak zajistit, aby těchto informačních zdrojů mohly využívat pouze oprávněné osoby. Do nedávné doby byl tento způsob řešen tak, že byl definován výčet „povolených“ IP adres – nejčastěji šlo o IP rozsah celé organizace.

V poslední době je však běžné, že oprávnění uživatelé často potřebují přistupovat k těmto zdrojů nejen z povoleného rozsahu. Jistě, je možné jisté procento případů řešit pomocí VPN (Virtual Private Network), které obsáhnou i domácí nebo mobilní počítače několika zaměstnanců, ale zcela určitě není možné toto řešení využít u tisíců studentů.

V tuto chvíli přichází ke slovu identity federation, které by v oficiální nebo spíše správné formě mělo vypadat tak, jak je patrné z obrázku 5.

Tento speciální případ identity federation, kdy služby poskytujeme výhradně vlastním uživatelům a subjekty mimo organizaci jsou pro nás jen poskytovatelé služeb, lze řešit poměrně snadno pomocí dříve diskutovaného SSO řešení WebAuth. Jak vypadá graf tohoto speciálního případu identity federation je vidět na obrázku 6.



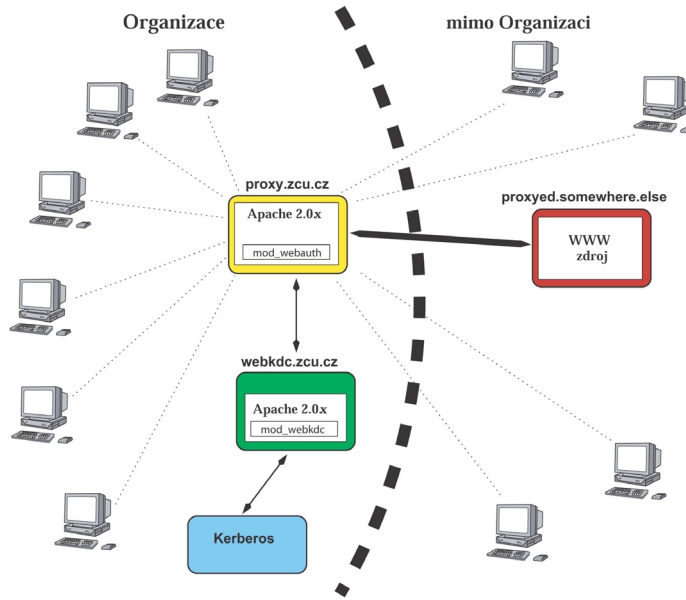

Obr. 6 Organizace je IdP; SP jen pro vlastní účely

Toto řešení je možné nazvat jako webový proxy server chráněný WebAuthem (nebo též webauth proxy-server). Celý trik spočívá v relativně jednoduché konfiguraci WWW serveru uvnitř organizace tak, aby se choval jako WWW proxy pro WWW server poskytovatele[4]. Pak již stačí pro lokální server vyžadovat autentizaci přistupujícího uživatele. Pokud je to vhodné nebo žádoucí je také možné login uživatele posílat na vzdálený server v HTTP hlavičkách požadavků. Princip celého řešení je patrný z obrázku 7.

Celá konfigurace by pak vypadala následovně:

```
ProxyRequests on
<Proxy http://proquest.umi.com/*>
  order allow,deny
  allow from all
  AuthType WebAuth
  Require valid-user
  RequestHeader set Webauthproxy ":%{WEBAUTH\_USER}e:"
  WebAuthExtraRedirect on
</Proxy>
```

[4]Postup byl v podstatě převzat přímo od vývojářů sytému WebAuth a stejným způsobem je i používán na Lealand Stanford Junior University v Kalifornii.




Obr. 7 Autentizovaná webová proxy

```
<VirtualHost 147.228.4.46:80>
  ServerName www.proquest.zcu.cz
  ServerAlias proquest.zcu.cz
  DocumentRoot /afs/zcu.cz/project/www/
  ServerAdmin indy@civ.zcu.cz
  ErrorLog /var/log/apache2/www.proquest-error_log
  TransferLog /var/log/apache2/www.proquest-access_log
  Redirect / https://www.proquest.zcu.cz/
</VirtualHost>

<VirtualHost 147.228.4.46:443>
  ServerName www.proquest.zcu.cz
  ServerAlias proquest.zcu.cz
  DocumentRoot /afs/zcu.cz/project/www/
  ServerAdmin indy@civ.zcu.cz
  LogFormat  "%h %l %u %t %v \"%r\" %>s %b \"%{Referer}i\" \
     \" %{User-Agent}i\"" virtual
  ErrorLog /var/log/apache2/ssl-proquest-error_log
  CustomLog /var/log/apache2/ssl-proquest-access_log virtual

  SSLEngine on
```

```
  SSLCipherSuite \
    ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
  SSLCertificateFile \
    /etc/apache2/ssl/www.proquest.zcu.cz/server.crt
  SSLCertificateKeyFile \
    /etc/apache2/ssl/www.proquest.zcu.cz/server.key

  <Location / >
     ProxyPass http://proquest.umi.com/
     ProxyPassReverse http://proquest.umi.com/
  </Location>
</VirtualHost>
```

Výše uvedený případ WWW proxy serveru chráněného WebAuthem není vykonstruovaný, ale je rutinně používaný pro přístup k elektronickým zdrojům Proquest. Tento informační zdroj nakoupila knihovna při ZČU pro potřeby organizace.

Tento příspěvek vznikl za podpory grantu 087/2004 při Fondu rozvoje CESNET.

# Literatura

[1] Grolmus, P., Švamberg, M.: Identity federation nejen v univerzitním prostředí. Vyšlo v *Data security management*, 2005, roč. 9, č. 1, s. 14–17, ISSN 1211-8737.

[2] Sborník příspěvků XXIV. konference EurOpen, obsahující prezentace několika různých IF systémů; ISBN 80-86583-06-6; http://www.europen.cz/.

[3] http://webauthv3.stanford.edu/ – domovská stránka SSO systému WebAuth vyvíjeném na Lealand Stanford Junior University, California, USA.

[4] http://www.umich.edu/ umweb/software/cosign/ – webové SSO řešení z University of Michigan, Michigan, USA.

[5] http://www.pubcookie.org/ – oficiální stránka webového SSO PubCookie.

[6] http://shibboleth.internet2.edu/ – akademický projekt IF systému Shibboleth.

[7] http://www.projectliberty.org/ – komerční projekt IF systému Liberty Aliance.

# Řízení SW projektů – příručka pro přežití

**Václav Pergl**

E-mail: vpergl@kerio.com

**Abstrakt**

*V příspěvku jsou ukázány obecně platné zásady projektové řízení i některá specifika SW projektů. Hlavní důraz je kladen na klíčové procesy a hlavní role SW projektu. Dodržování předloženého (podobného) postupu, podstatným způsobem zvyšuje pravděpodobnost úspěšného dokončení SW projektu a tím i šanci na přežití jeho projektového manažera.*

## 1 Cíle

### Vytvářet správný produkt

Nutnou podmínkou pro přežití manažera softvarového projektu je, že vytváří produkt, který uspokojí potřeby zákazníků. Aby tyto potřeby mohl uspokojit musí je nejen dobře znát, ale i mít pod kontrolou jejich změny v průběhu celého projektu.

### Vytvářet produkt správně

Na základě formulovaných, správně zaznamenaných a odsouhlasených potřeb je třeba vytvořit příslušné specifikace (modely – konceptuální, logický, fyzický, . . . ). Při vývoji produktu je nutno zajistit shodu mezi vytvářeným kódem programu a odsouhlasenou specifikací.

### Respektovat omezení projektu

Další nutnou podmínkou pro přežití je respektování projektových omezení – v projektovém trojúhelníku *Cíl* – *Zdroje* – *Termín*, lze zvolit maximálně dva vrcholy.

# 2 Životní cyklus vývojového SW projektu

## Definice

Hlavním výstupem definiční fáze SW projektu je dokument nazývaný Vision Statement (Vision/Scope Document, Software Requirements Specification Document, apod). Smyslem tohoto dokumentu je vytvořit definici rozsahu připravovaného projektu. Pro zde uvedené vlastnosti budou vytvořeny podrobné spefikace a na jejich základě bude připraven plán projektu.

Velmi důležitou součástí fáze je formování projektového týmu a určení rolí jednotlivých členů týmu.

## Plánování

Na základě vytvořených specifikací připraví projektový manažer ve spolupráci s členy týmu, hlavní plán projektu, který obsahuje alespoň:

1. Časový plán projektu,
2. Plán vytížení zdrojů,
3. Rozpočet projektu,
4. Plán testů.

## Vývoj

V průběhu této fáze vývojáři vytvářejí zdrojový kód aplikace a sestavují spustitelnou aplikaci. Velmi důležité je denní automatické vytváření spustitelné podoby vyvíjené aplikace a podrobení tohoto buildu rozsáhlým (nočním) automatickým testům.

## Stabilizace

Stabilizační fáze je zahájena absolutním zmražením požadavků na funkcionalitu vytvářené verze produktu, zahájení intezivních a rozsáhlých testů vyvíjené aplikace a odstraňováním nalezených chyb.

V závěru fáze je produkt poskytnut skupině beta testerů a následně je vyvořena RC (Release Candidate) verze.

## Vydání

Po úspěšném ukončení testů RC verze lze přistoupit k uvolnění finální verze pro trh.

# 3 Role

## Produktový manažer

Smysl práce produktového manažera lze poměrně jednoduše shrnout do pouhých dvou slov: „Uspokojení zákazníků“. Jeho činnost lze tedy charakterizovat, jako prosazování potřeb zákazníků v projektové týmu.

## Projektový manažer

Projektový manažer je primárně odpovědný za dodání kvalitního softwarového produktu s danými vlastnostmi v naplánovaném termínu s alokovanými zdroji.

## Vedoucí vývojového týmu

Je zodpovědný za vytvoření specifikace produktu dle požadavků zákazníků a softwarovou realizaci této specifikace.

## QA manažer

Je zodpovědný za řádné otestování produktu dle vytvořené specifikace testů a nahlášení všech nalezených problému v produktu.

## Release manažer

Odpovídá za úplnost finálního produktu a jeho vydání pro zákazníky.

# 4 Procesy

## Řízení změn

Vzhledem k tomu, že je nerozumné (někdy i nemožné) změny na SW projektu zakázat, dobrý projektový manažer musí dostat změny pod kontrolu. Toho lze nejlépe dosáhnout definováním procesu řízení změn a jmenováním Komise pro změny (Change Board).

## Řízení požadavků

Protože uspokojení potřeb zákazníků je primární podmínkou pro přežití projektového manažera, je proces řízení požadavků zákazníků významným procesem SW projektu.

## Odstraňování chyb

V průběhu projektu je třeba provádět testování a odstraňovat nalezené chyby. Formalizací tohoto procesu zajistíme, že každá nalezená chyba je přidělena konkrétní osobě odpovídající za její odstranění a ověření, že problém byl skutečně vyřešen.

# ICT projekty v 21st. – nic složitějšího

**Antonín Bulín**

E-mail: bulin@rek.zcu.cz

**Abstrakt**

*Příspěvek bude věnován problematice přípravy a řízení ICT projektů a rizik s nimi spojených (zejména v předkontraktační fázi), v návaznosti na strategii a komerční cíle společnosti, v době tvrdého konkurenčního boje. Autor se pokusí položit otázky vztahu IT a strategie firem, odvozených od tvrzení typu: „IT je příliš důležité, aby bylo svěřeno IT manažerům.“*

Příspěvek nebyl dodán.

# Uplatnění teorie omezení (TOC) v projektech IS/ICT

**Josef Basl**

E-mail: basl@vse.cz

**Klíčová slova:** podnikové informační systémy, projekty IS/ICT, teorie omezení, constraint management

**Abstrakt**

*Příspěvek poskytuje základní charakteristiku principů TOC, přípravu projektu, oblasti nasazení TOC v podnikové praxi, specifika projektů řízených podle TOC, zásady správné komunikace a eliminace překážek nástroji včetně uplatnění metrik TOC při optimalizace přípravy a řízení projektu.*

## 1 Úvod

Většina knih o Teorii omezení svým obsahem podporuje zažité povědomí o teorii omezení (Theory of Constraint – TOC) především jako velmi účinné metodiky plánování a řízení výroby. Je tomu pravděpodobně z toho důvodu, že první publikovanou aplikací byla Drum-Buffer-Rope popsaná formou eseje v knize Cíl (v originále The Goal) v roce 1984. Ta následně sloužila jednak jako praktické ověření myšlenek TOC a jednak se dala následně vzniknout dalším nástrojům, které dnes lze zahrnujeme pod teorii omezení.

Připomeňme, že přístup TOC nabízí vlastní řešení, jak se orientovat ve složitosti podnikové reality. Tento přístup je odlišný od řady v praxi aplikovaných přístupů, jako je například známý Paretův principu. Dle tohoto principu, často označovaného jako pravidlo 80 : 20, se doporučuje soustředit se na 20 % hlavních činností, které na druhé straně zajišťují vytvoření 80 % efektu. Tento postup je sice rovněž vhodný při rozhodování v podmínkách složitého systému, ale předpokladem je určitá opakovanost těchto činností či jevů. Uplatnitelnost Paretova principu se snižuje s klesající opakovaností sledovaných jevů, což je stále častější

průvodní jev pro současné podnikové prostředí. Proto je v podnicích důležité hledání a využívání nových metod, mezi které bezesporu TOC patří, přičemž její použití je poměrně široké a využitelné ve třech základních oblastech:

- podpora rozhodování pro hlavní podnikové činnosti – vedle již zmíněné výroby lze dále uvést distribuci, marketing, prodej a řízení projektů

- průtoková analýza – finanční aplikace TOC, která může pomoci při změně rozhodování od zohlednění zejména nákladů k procesu trvalého zlepšování, při kterém klíčovými elementy je hlavní ukazatel TOC – průtok systému, dále omezení systému a statisticky stanovená ochrana kapacit a kritických bodů

- logická analýza v TOC (tzv. thinking process) – představuje třetí oblast všeobecně použitelných nástrojů k identifikaci a řešení různých problémů v organizaci. Logika TOC je aplikována k identifikaci, které faktory jsou v organizaci limitující k dosažení cílů k vytvoření návrhu řešení problému a vtažení pracovníků i do procesu nalézání předpokladů řešení.

TOC napomáhá při řešení a následné zlepšování obvyklých problémů v podniku, jako jsou např.:

- vizualizaci a zlepšování procesů,

- pomoc při hledání nových strategií a přístupů s podporou jejich následné realizace.

Díky pevné logice, vzájemné kauzalitě jevů a vizualizaci pomáhá řešit problémy založené na verbalizaci pocitů, emocí a intuice. Nástroje TOC pomáhají zlepšit porozumění současnému světu a dosažení cíle orientovaného na trvalé zlepšování.

Tím napomáhá i zlepšení:

- komunikace,

- hledání řešení,

- řešení konfliktů,

- učení,

- vývoje a implementaci informačního systému.

## 2 Hlavní principy Constraint managementu

Teorie omezení, kterou vypracoval Dr. E. M. Goldratt, používá systémový přístup, dívá se tedy na výrobní systém z globálního pohledu. Nezajímá ji, jak fungují jednotlivé části celku, ale jak funguje celek. Jednotlivé části systému se musí podřídit cíli, který si daný systém určil. Tomuto globálnímu pohledu odpovídá jak metrika, tak metody řešení problémů včetně jejich nástrojů.

Tento přístup předpokládá, že:

- každý systém je součástí většího systému,
- systém má cíl, který chce dosáhnout,
- systém jako celek je více než pouhý součet jeho částí,
- úsilí systému je omezeno jednou proměnou (nebo velmi málo proměnnými).

Každý systém je součástí většího systému, tzn. že na žádný systém nemůžeme pohlížet odděleně, ale jako na součást většího celku. Změna jednoho atributu systému se může odrazit i v dalších systémech/subsystémech.

Cílové chování systému je jednou z nezbytných podmínek chování každého systému, který chce být z dlouhodobého hlediska životaschopný. Od vytyčeného cíle se odvíjí strategie a způsob jejího naplnění.

Je samozřejmé, že celý systém bude fungovat lépe, jestliže jeho jednotlivé části budou podřízeny úsilí celku k dosažení vytyčeného cíle, než kdyby každé části celku sledovaly a plnily své částečné (lokální) cíle. Plnění těchto lokálních cílů neznamená ve výsledku plnění cíle globálního.

Každý systém je limitován omezením, protože jinak by svých cílů dosahoval neomezenou rychlostí a v neomezeném čase. Těchto omezení je vždy jen několik. Zde je nutno poznamenat, že je nezbytné každý systém důkladně analyzovat a jednoznačně určit, co je skutečné omezení (a tudíž naše priorita při přeměně systému) a co není.

Obecně omezení (úzké místo) je takové místo, které brání systému v dosažení jeho cíle. Základní typy podnikových omezení:

- zdrojová a kapacitní,
- časová,
- hodnocení a měření,
- prodejní,
- organizační,
- komunikační,
- kulturní.

TOC navrhuje pěti bodový postup pro trvalé zlepšování podniku a řízení úzkých míst. Protože jestliže nebudeme řídit úzká místa, úzká místa budou řídit nás:

- identifikace úzkého místa,
- využití úzkého místa na 100 %,
- podřízení podniku tomuto úzkému místu,
- rozšíření tohoto úzkého místa,
- vše znovu od začátku.

Přístup TOC obecně ovlivňuje nárůst efektivity řízení a fungování podniku v dané oblasti. Jeho řešení v obecné rovině však žádným způsobem nepopisuje, nevyžaduje ani nezohledňuje roli informační systému podniku. A přitom informační systémy a zejména nástroje ERP sehrávají v podnicích od počátku 90. let klíčovou úlohu a spojením s TOC se zvyšuje jejich potenciál.

# 3 Důvody pro použití TOC v podnikových informačních systémech

Je zajímavé, že oblasti podnikových informačních systémů reprezentovaná řešeními typu ERP prožívala svým způsobem v 90. letech minulého století svůj „zlatý věk“. Rostly počty implementací a zvyšovala se jejich funkcionalita umocněná navíc potenciálem internetu. Ale přesto i v těchto dobách se objevily kritické názory upozorňující na malou respektive problematickou návratnost těchto projektů v praxi. Hledání příčin začalo již v první polovině 90. let, na něž navázalo hledání účinných „léků“ řešících problém. Tím nejznámějším je patrně procesní reengineering (BPR) Hammera a Champyho. Nejeden český podnik a organizace tuto proceduru absolvovalo a nutno dodat, že ne vždy s potřebným efektem. Příčin je jistě více a jednou z odpovědí na takovéto snahy a projekty bylo využít jednak BSC (Balanced Scorecard) a nebo vhodné metriky. Ten měl zajistit větší preciznost cílů a měření progresu na projektu. Tuto snahu využily i tendence k outsourcing služeb spojených s podnikovými informačními systémy.

Celkově se dá říci, že hlavní neefektivnosti spojené s projekty ERP se souhrnně týkají:

- nevhodného systému ERP,
- neadekvátních podnikových procesů,

- nesprávně fungujících vztahů mezi dodavatelem a uživatelem, vč. metrik a smluv.

Všechny tyto snahy postupně směřovaly k nalezení a určité vizualizaci aspektů projektů ERP pro všechny zúčastněné:

- obě strany projektu (dodavatele a uživatele),
- všechny úrovně uživatelů (od vrcholového managementu až po uživatele zadávající základní transakcí údaje z financí, logistiky a prodeje).

Snad proto, že je využívali a podporovali technici a zejména informatici, zůstávali na úrovni technologie. Přitom technologie typu ERP vyžaduje důraznou součinnost obou stran – jak dodavatele tak uživatele. Tu ale zároveň ovlivňují podmínky implementací, včetně například způsoby placení vycházející ve většině případů z placení konzultantů v člověkohodinách.

Dosud není až tak obvyklé uzavírat při prodeji ERP smlouvu formou garance přínosů z vlastní implementace.

V praxi tak je ERP na jedné straně skvělou technologií, ale současně standardní postup ERP implementace se může výrazně lišit od potřeb daných situací podniku. Navíc po zmíněných „zlatých časech“ ERP, které jakoby symbolicky ukončil rok 2000, se objevila po období extenze snaha po intenzifikaci. Se zajímavým názorem v této souvislosti přišel E. Goldratt – autor TOC. ERP z jeho pohledu není efektivní, protože neřeší změnu pravidel pro používání této špičkové technologie.

# 4 Využitelnost TOC v projektech IS/ICT a podnikové informatice

Jedna ze základních otázek tohoto přístupu je, proč implementace ERP systémů nevykazují návratnost, která se obvykle ukazuje na prodejních prezentacích a představuje jeden z hlavních argumentů dodavatelů pro nákup jejich ERP systému? Proč je obtížné hledat odpověď při inovaci ERP? Odpověď je třeba hledat na trhu samotných ERP systémů a analýze chování jejich dodavatelů. Ukazuje se, že ačkoli je informační systém (většinou nějaký komerčně dostupný ERP systém) v podniku instalován a je naplněn daty, přesto nepřináší očekávaný užitek. Důvodem je fakt, že neposkytuje takové informace, podle kterých by bylo možné podnik řídit lépe, než je tomu doposud (tj. před implementací ERP).

Takže na jedné straně existuje technologie ERP, která podstatným způsobem může překonávat současná omezení systému, ale je nutné současně podstatným způsobem změnit pravidla vytvořená pro „žití“ (tzn. zejména rozhodovaní)

v podniku s omezením před implementací ERP. Využití TOC je v dané oblasti ale mnohem širší, protože v souvislosti s aplikacemi IS/ICT ji lze efektivně nasadit především při:

- přípravě a tvorbě IS/ICT strategie, vč. účelných metrik podporujících růst příjmů podniku – tzv. průtoku,
- přípravě projektů změn v podnikových aplikacích IS/ICT,
- analýze současného stavu,
- specifikaci klíčového omezení a nalezení způsobu provedení požadované změny vedoucí k odstranění omezení, a to včetně limitujících procesů,
- popisu přechodu na požadovaný stav pomocí očekávaných dílčích cílů a jevů, vč. stanovení směru postupu realizaci změny,
- určení způsobu provedení změn,
- podpoře realizace projektů IS/ICTvyužitím buffer managementu a metody kritického řetězce (Critical Chain),
- podpoře mediativních technik řešení konfliktů a vyjednávání v průběhu implementace IS/ICT, např. ERP,
- podpoře trvalého zlepšování (on-going process improvement) – 5 kroků zlepšování,
- podpoře stanovení vhodných metrik na bázi průtokového účetnictví,
- podpoře inovace ERP,
- podpoře inovace podnikových procesů a jejich trvalého zlepšování.

Podívejme se nyní na základní koncepci naplnění těchto bodu trochu podrobněji.

# 5 Koncept zlepšení v podniku podle TOC

Základem analýzy jsou nástroje tzv. thinking procesu, který modeluje stávající a budoucí realitu v přesné kauzalitě.

Základem jsou následující stromy:

- při mapování současného stavu a identifikaci hlavního problému – strom současné reality (Current Reality Tree – CRT),

- při zachycení požadovaného stavu a hlavních žádoucích efektů – strom budoucí reality (Future Reality Tree – FRT),

- při specifikaci možných překážek navrhovaného zlepšení a jejich řešení – strom předpokladů (Prerequisite Tree – PrT),

- při specifikace dílčích kroků řešení s určením nutných podmínek i očekávaných výsledků – strom přechodu (Transition Tree – TrT).

Kromě diagramů stromů se při projektech často uplatňuje a tuto kauzální logiku rovněž využívá diagram konfliktu (někdy též označované jako mizející mrak (evaporating cloud).

Obecně se diagramů stromů využívá při hledání dopovědí na tři základní otázky:

1. Co změnit (What to change) – cílem je odhalení současného omezení s použitím techniky stromu současné reality.

2. Na co to změnit (What to change to) – cílem je s použitím technik stromu budoucí reality a diagramu konfliktu popsat budoucí cílový stav.

3. Jak změnu provést (How to change) – cílem je propracovat implementační plán změny ze současného stavu do stavu budoucího. Využity přitom jsou techniky stromu překážek a přechodu.

Proces vytvoření těchto diagramů je velmi účinný pro efektivní průběh návrhu strategie a projektu změny, i když si jejich zpracování vyžaduje určitý čas řešitelského týmu. Výhodně tyto diagramy mohou doplnit výsledky zpracované v rámci SWOT analýzy podniku a jeho podnikové informatiky.

# 6 Řešení efektivnosti implementace IS/ICT dle TOC

Jak již bylo řečeno, TOC předpokládá, že ERP řešení jsou neefektivní proto, že neřeší v rámci implementace změnu pravidel uživatelů pro používání této špičkové technologie. Hlavní myšlenka TOC vychází z teze, že jakákoliv technologie může přinést efekty pouze a jenom tehdy, když zmenší vliv existujících omezení. Předpokládá, že před tím, než je nová technologie dostupná, si uživatelé vytvořily formy chování, měřítka a postupy, které jim v podniku pomohli přizpůsobit se existujícím omezením. Primárně se tedy úsilí TOC soustřeďuje na změnu základních pravidel fungování podniku – na změnu chování uživatelů. Tímto způsobem implementace je podporováno zlepšení návratnosti ERP řešení.

Důvodem je fakt, že dříve než je nová technologie dostupná si uživatelé vytvořili formy chování, měřítka a postupy, které jim pomohly přizpůsobit se existujícím omezením. Zásadní potom je, jaké efekty budeme mít z jakékoliv nové technologie, pokud existující pravidla nebudou změněna. Hlavní argumentace se odvíjí od uvědomění si skutečnosti, v čem spočívá síla nové informační technologie.

Pokud přijmeme fakt, že hlavní schopností ERP systému je zcela jednoznačně jejich klíčová schopnost shromažďovat, uchovávat, vyhledávat, třídit a prezentovat obrovská množství údajů ze všech možných oblastí řízení podniku, pak také platí, že ERP je technologie jako podmínka nutná. Jak však z ní lze zároveň udělat podmínku i postačující? Odpověď pomohou sestavit následující otázky:

- Jaká pravidla se musí změnit?
- Jaká pravidla máme používat místo stávajících?
- Vyžadují nová pravidla změny ve způsobu využívání technologie?
- Jak prakticky způsobit změnu?

V každém případě jsou důležité měřitelné výstupy každého projektu změny na bázi IS/ICT vyjádřitelné v korunách. Tato minimální dosažitelná zlepšení by měla být společným motivátorem projektu jak pro implementační tým podniku, tak zejména pro dodavatele (např. dodavatele ERP).

## Literatura

[1] Basl, J., Majer, P., Šmíra, M.: *Teorie omezení v podnikové praxi.* Gradapublishing, Praha 2003.

[2] Basl, J.: *Podnikové informační systémy.* Gradapublishing, Praha 2002.

# Neurovědy a IT

**Milan Šárek, Pavel Voral**

E-mail: milan.sarek@cesnet.cz, pavel.voral@uvn.cz

**Abstrakt**

*Díky rozvoji optických sítí se otevřela možnost vysokorychlostního napojení medicínských pracovišť v ČR. Jedním z prvních zájemců o využití této technologie je špičkové pracoviště neurověd v Ústřední vojenské nemocnici v Praze. Ve sdělení bychom rádi informovali o aktuálním stavu a možnostech, jak se vyvíjí napojení tohoto pracoviště. Mimo vlastní projekty ÚVN a dalších spolupracujících nemocnic v ČR, bychom rádi dokumentovali možnosti využití vysokorychlostních sítí na projektu BIRN. Biomedical Informatics Research Network (BIRN) v současné době zajišťuje propojení výzkumných týmů z 30 univerzit a 21 dalších organizací s koordinačním centrem v UCSD (University of California, San Diego). Jeho zaměření je přednostně na výzkumu v oblasti neurověd.*

## 1 Úvod

Předpokládáme, že pojem IT bude přítomným kolegům alespoň trochu povědomý a proto se více zaměřím na pojem neurovědy. Neurovědy se zabývají studiem struktury, funkcí, vývoje, genetiky, biochemie, fyziologie, farmakologie a patologie nervového systému. Studium chování a učení je také předmětem neurověd [1]. Typickými představiteli tohoto oboru (silně zjednodušeně) jsou neurochirurgové, neurologové a psychiatři.

Je zajímavé, že již od samého počátku využívání výpočetní techniky ve zdravotnictví, byli mezi průkopníky lékaři z oboru neurověd, zejména psychiatři. Důvodů může být řada. Mimo první, který může napadnout, že se tito lékaři cítili mezi programátory jako mezi svými pacienty. Po studiu příslušné literatury [2], jsme však přišli k závěru, že pojem normality je natolik pružný, že z odborného hlediska to nemusel být hlavní důvod. Zejména v akademickém prostředí jsou některé odchylky od běžného chování přijímány s větší tolerancí a vcelku nevybočují z takto nastaveného rozmezí normality.

Jiný přijatelný důvod může vycházet z potřeby psychiatrie a neurologie analyzovat poměrně složité projevy vyšší nervové činnosti. Názorným příkladem může

být náročná analýza elektrického signálu činnosti mozku EEG (elektroencefalogram). V podstatě se jedná o nalepení obvykle 16 až 25 elektrod (vyskytují se i specializované systémy až s 64 elektrodami) na skalp. Náročnost spočívá v tom, že v tomto případě nejsou až tak důležité lehce měřitelné hodnoty signálu (např. amplituda, střední hodnota napětí), ale výpovědní hodnotu má harmonická analýza signálu. Výstižně popisuje problém M. Novák [3]: „Paralelou k EEG je situace, kdy bychom ze záznamu několika mikrofonů umístěných v různých místech nad hlavami ohromného davu hlučících lidí činili soud o celkové náladě a stavu tohoto davu.“ Naštěstí problém se zjednodušuje, protože lékaře hlavně zajímá, která frekvence v signálu převažuje. Základní dělení je na:

- frekvence alfa (8 až 13 Hz) – zjednodušeně odpovídá stavu relaxace
- beta (14 až 30 Hz) – přibližně odpovídají aktivní fázi
- théta (4 až 8 Hz) a delta (méně než 4 Hz) jsou obvyklé pro patologické stavy.

V kombinaci s dalšími údaji (zda měl pacient zavřené oči, zda se jedná o evokované potenciály (následek určitého dráždění – elektricky, mechanicky, farmakologicky)) může dát vyšetření EEG základní informaci o typu onemocnění, případně příčině nebo oblasti poškození mozku. Bylo až nepředstavitelné, že kvalifikovaný neurolog byl schopen pomocí speciálních měřítek provádět analýzu ručně na papírovém záznamu EEG a po přibližně desetileté praxi vcelku kvalitně. V každém případě, nasazení programů využívajících principů rychlé Fourierovy transformace přinesla možnost nezvykle rychlé analýzy, která navíc mohla být oproštěna od subjektivního faktoru lidského hodnotitele.

Možná to byl první krok. Byl však úspěšný, protože přinesl lékařům úsporu práce a času. Tím se možná tento vstup odlišoval od některých dalších alternativ, kdy s počítači přicházela jen další forma administrativy, která jistým způsobem lékaře od pacienta vzdalovala. Vznikal virtuální pan Vomáčka v počítači, který byl to podstatné pro tok peněz od zdravotní pojišťovny a vytváření této virtuální postavy odvádělo lékaře od reálného pana Vomáčky, který aby přežil, potřeboval přítomnost lékaře u lůžka, nikoli u počítače.

## 2 Připravovaný rozvoj

Medicínská oblast představuje zdroj zajímavých aplikací, které často vyžadují kvalitní infrastrukturu s velkou propustností. Sdružení CESNET má dlouhodobě o tuto oblast zájem. V druhé polovině devadesátých let proběhla řada medicínských videokonferencí, infrastruktura sdružení CESNET byla rovněž využívána pro přenos medicínských obrazových dat. Protože zájem o medicínské projekty postupně rostl, došlo v druhém pololetí roku 2004 k ustavení samostatné aktivity

Medicínské aplikace. Záměrem této nové aktivity je vyhledávat a dále rozvíjet další projekty z oblasti medicíny.

Jedním z prvních subjektů, který vyjádřil svůj zájem se zapojit do nových aplikací, byla Ústřední vojenská nemocnice, která je mimo jiné špičkovým pracovištěm v oblasti neurověd. První impuls, tak jak to odpovídá úvodní stati, přišel z psychiatrického oddělení, ale plánované akce jsou v mnohem větším rozsahu, která přesahuje rámec tohoto jednoho oddělení.

Druhým klíčovým subjektem je Fakultní Thomayerova nemocnice v Praze Krči, která má zájem a potenciál se stát komunikačním centrem pro zpracování obrazových dat v pražské oblasti a navázat tak na původní aktivity brněnského Medimedu, které je zmiňován na tomto semináři v samostatné přednášce.

Situace dalšího propojování se zkomplikovala s ohledem na změny v pražském zdravotnictví. Původní záměr propojit Všeobecnou fakultní nemocnici na Karlově náměstí s Fakultní nemocnicí na Bulovce a Fakultní Thomayerovu nemocnici v Krči do jednoho „virtuálního“ celku platil asi jeden rok. Během této doby byly nastartovány aktivity, které měly umožnit vytvořit mimo jiné adekvátní dostatečně propustnou infrastrukturu těchto propojených nemocnic. K realizaci bohužel nedošlo a v současné době se jednotlivé nemocnice vrací k modelu samostatného fungování.

Další klíčovým subjektem je v současné době Masarykova nemocnice v Ústí nad Labem (MNÚL). Jedná se o nemocnici vybavenou systémem bezfilmového zpracování obrazové nemocnice, která se snaží sledovat nejmodernější trendy. V současné době je například zapojena do grantu Akademie věd MediGRID [4].

# 3 Systém pro neurologické konzultace

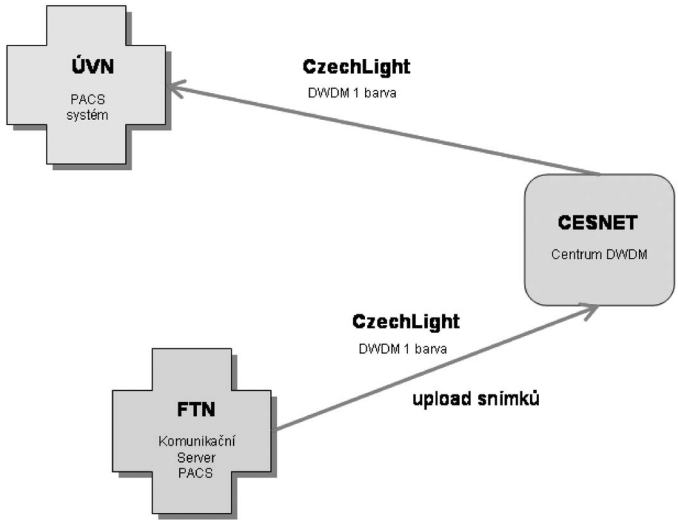
Jak jsme se již zmínili v předcházející kapitole, špičkovým pracovištěm v oblasti neurověd je Ústřední vojenská nemocnice v Praze Střešovicích (ÚVN). O konzultace na takovém pracovišti má zájem například Fakultní Thomayerova nemocnice v Praze Krči (FTN). Protože obě nemocnice mají vybavení pro bezfilmové zpracování PACS (Picture Archiving and Communication System), nabízí se možnost propojení přes počítačové sítě. Takovéto propojení musí řešit řadu problémů.

Jedním z požadavků týmu lékařů ve Střešovicích je zpracování těchto konzultací v prostředí systému jim známém, což znamená upload snímku ze systému FTN do systému ÚVN.

Další otázka je zajištění dostatečných přenosových kapacit a vyřešení zabezpečení přenosu.

V souvislosti s budováním infrastruktury pro podporu rozvoje komunikačního serveru PACS byl vytvořen gigabitový spoj sítě CzechLight s využitím technologie DWDM (Dense Wavelength Division Multiplexing). Obdobně v souvislosti

s řešením přípravy napojení na projekt BIRN byl vybudován další spoj Czech-Light také do ÚVN. Oba spoje jsou vedeny z centra sdružení CESNET v Praze Dejvicích. Díky tomu je možné vyřešit v rámci pilotního provozu první problém, kterým bylo nedostatečná rychlost připojení nemocnic. Zejména přenos sekvencí z počítačového tomografu (CT), což je pro vyšetření v oblasti neurologie a neurochirurgie typické, je schopen způsobit problémy, pokud je celý provoz nemocnice veden pouze přes linku 100 Mb/s. Přenos musí být rychlý i spolehlivý z toho důvodu, protože se jedná o život ohrožující stavy a snímek se v těchto případech běžně posílá sanitkou rychlé záchranné služby.

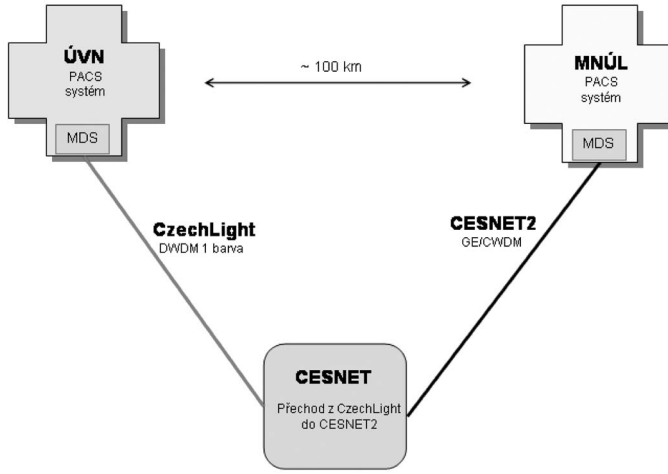



Obr. 1 Systém pro neurologické konzultace

Druhým aspektem je zabezpečení přenášených dat. Tady je opět otevřena nová možnost řešení. K přenosu bude použito v celé trase systému DWDM. Bude možné zajistit bezpečnost spoje na základní fyzické úrovni a to díky frekvenčního dělení kanálů. Zjednodušeně se jedná o využití pro přenos informací mezi nemocnicemi jedné velmi úzké části barevného spektra optického kabelu. Protože současné technologie umožňují rozčlenění až na 160 kanálů, je s ohledem na technologické a ekonomické limity nemožné se do takto vytvořeného systému nelegálně napojit. Pro další rozvoj takových to systémů bude vhodným vodítkem doporučení k vytváření Optical VPN (Virtual Private Network) [5].

# 4 Systém SAN

Případ propojení lokalit ÚVN ve Střešovicích a MNÚL v Ústí nad Labem bude potřeba řešit trochu odlišným způsobem. Na rozdíl od ÚVN je MNÚL napojena prostřednictvím sítě CESNET2. Na celé trase tedy nelze použít výhodu zabezpečení na fyzické úrovni pomocí DWDM technologie. Také účel propojení je odlišný než v případě propojení ÚVN a FTN.




Obr. 2 Struktura SAN řešení

V tomto případě je zájem vytvořit společný SAN (Storage Area Network), který bude sloužit pro kontinuální vytváření záložní kopie pacientských dat. Zálohování bude probíhat recipročně. Kopie dat z ÚVN bude ukládána v Ústí a naopak.

Předpokladem vytvoření tohoto systému je vybavení obou pracovišť hardware podporou FCoIP (Fibre Channel over IP), například zařízení typu MDS řady 9000, které mimo uvedenou konverzi protokolů dále zajistí jejich kódování z důvodu zabezpečení dat a management celého systému přenosu dat.

Uvedený systém zálohování jednak zvýší spolehlivost ukládání dat. Navíc v případě výpadku nebo odstavení datového skladu v jedné lokalitě (servis, profylaxe), umožní práci s daty v druhé lokalitě s akceptovatelnou dobou odezvy.

Nezanedbatelnou výhodou je také geografická vzdálenost datových úložišť přibližně 100 km od sebe, což vylučuje poškození obou systému současně i v případě katastrofických scénářů.

# 5 Zapojení do projektu BIRN

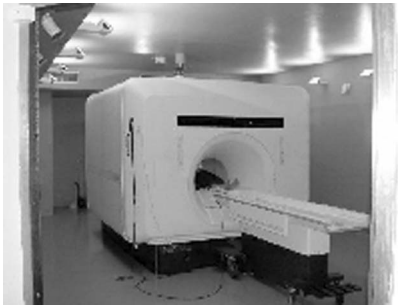
Jedním z velkých projektů, které podporují výzkum v oblasti neurověd je projekt Biomedical Informatics Research Network (BIRN), který v současné době zajišťuje propojení výzkumných týmů z 30 univerzit a 21 dalších organizací.

Základní členění tohoto projektu je do tří oblastí:

1. prostorové modely lidského mozku a mozku pokusných zvířat,
2. specializované databáze v oblasti neurověd a
3. možnost vzdáleného přístupu ke specialním zařízením.

Klíčovým problémem komunikace mezi jednotlivými pracovišti je vysoký nárok na objem přenášených dat. Proto se na koordinační centrum (BIRN-CC) v UCSD (University of California, San Diego) nejprve připojovala americká pracoviště.

Během úvodního jednání, které proběhlo mezi zástupci BIRN-CC a sdružení CESNET během konference HealthGrid 2005 v dubnu letošního rok, přislíbil Prof. Ellisman, ředitel BIRN-CC, možnost zapojení kvalifikovaného českého pracoviště do sítě BIRN.



Obr. 3 Scanner MRI se supravodivým magnetem [6]

Faktory, které vedly k výběru pracoviště:

- špičkové pracoviště v oboru neurověd
- kompletní profil oddělení
- diagnostické vybavení (MRI (Magnetic Resonance Imaging), 2× CT (Computer Tomography), angiografický komplex s digitální substrakční angiografií, kompletní radiologické a ultrazvukové vybavení)

- mezinárodní akreditace Joint Commission International (garance kvality péče na světové úrovni)
- rutinní zapojení do mezinárodní spolupráce (NATO)
- formulace svého zájmu zapojit se projektu BIRN

Na základě tohoto hodnocení byly zahájeny přípravně práce, které v současné době umožnily napojit ÚVN pomocí gigabitového DWDM spoje sítě CzechLight. Toto napojení je vyhrazeno pro pilotní provoz výzkumných aplikací.

Výzkumná síť BIRN se skládá z průběžně vyvíjeného software, hardware a metod sdílení zdrojů, které umožňují efektivní a bezpečné sdílení dat a vzájemnou spolupráci. BIRN software je navrženo na principu *interoperable network accessible services*.

Servisní komponenty obsahují:

- identifikaci uživatele (certifikaci)
- autorizaci uživatele
- digitální podpis dokumentů
- lokalizaci dat
- sdílení dat
- přístup k datům
- workflow engine
- doménově specifické aplikace

Jednou definované služby musí být zdokumentovány, připraveny k instalaci, konfigurovatelné a aktualizovatelné pro různé sestavy hardware.

Úlohou BIRN-CC (koordinačního centra) je definování, integrace, implementace aktualizace kompletní BIRN kyber struktury. Předpokladem efektivní správy tohoto komplexního systému je formalizace a rozšiřování procesu integrace, testování, využívání a údržby softwarového řešení. Dvakrát ročně (v dubnu a říjnu) vydává BIRN-CC v závislosti na vývoji hardware nové verze systému. Tento upgrade obsahuje:

- middleware pro zabezpečení, data a výpočty, sdílení dat a jejich integraci, BIRN portál
- BIRN bioinformatický software pro vědecké aplikace, které byly vyvinuty na BIRN aplikačních testovacích pracovištích (například LONI Pipelíne [8], 3D/Slicer [9], AFNI [10], AIR [11] a další).

Biomedicíncké aplikace jsou distribuovány ve formátu Red Hat Package Manager (RPM).

K zajištění optimálního využití software modulů v celé síti BIRN, je využíván v BIRN-CC NPACI Rock Cluster Toolkit [12], což je sada open source modulů, pro vytváření a správu Linux based clusterů.

# 6 Závěr

Zmíněná řešení jsou jen malou ukázkou možností rozvoje eHealth technologií. Investičně náročné technologie v konečném důsledku přináší provozní úspory a navíc se zvyšuje kvalitu ošetření pacientů. Zejména oblast neurověd je z tohoto hlediska velmi důležitá, protože se jedná o velmi složitý a náročný obor, který v řadě případů má pro pacienta velmi rozhodující úlohu ve stanovení léčby a další perspektivy kvality jeho budoucího života.

# Literatura

[1] URL: http://www.answers.com

[2] Syřišťová, E. a kol.: *Normalita osobnosti.* Avicenum, Praha 1972.

[3] Novák, M. a kol.: *Neuronové sítě a informační systémy živých organismů.* Grada, Praha 1992.

[4] URL: http://www.medigrid.cz/cs/oprojektu/index.html

[5] URL: http://www.ntt.co.jp/saiyo/e/rd/review/pdf/nw01.pdf

[6] URL: http://www.uvn.cz/html/CZ/oddeleni/rdg/pracoviste.php

[7] Grethe, J. S. at al.: *Biomedical Informatics Research Network: Building a national collaboratory to haste the derivation of new understanding and treatment of disease.* proc. of HealthGrid 2005, IOS Press, Oxford 2005.

[8] Rex, D. E. at al.: *The LONI Popelíne Processing Environment*, NeuroImage 19 (3), Elsevier 2003.

[9] Gering, D. at al.: *An Integrated Visualization systém for surgical planning and guidance usány image vision and interventional imaging.* In Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Cambridge Press, England 1999.

[10] Cox, R. W.: *AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages.* Computer and Biomedical Research 29, Elsevier 1996.

[11] Woods, R. P. at al.:*Automated image registration.* Journal of Computer Assisted Tomography (JCAT), 22, Lippincott Williams & Wilkins, 1998.

[12] URL: http://www.rocksclusters.org/Rocks/, leden 2004.

[13] Šárek, M.: *Telemedicínské aplikace a sdružení CESNET.* sborník konference MedSoft 2005, Benešov 2005.

[14] Dostál, O., Javornik, M., Slavíček, K., Petrenko, M.: *MEDIMED-Regional Centre for Archiving and Interhospital Exchange of Medicine Multimedia Data.* Proc. of the Second IASTED International Conference on Communications, Internet, and Information Technology, International Association of Science and Technology for Development – IASTED, Scottsdale, Arizona, USA 2003.

[15] Dostál, O., Slavíček, K., Petrenko, M.: *Použití vysokorychlostních sítí pro medicínské aplikace.* sborník Širokopásmové síte a jejich aplikace, Olomouc 2003.

[16] Javorník, M., Dostál, O., Andres, P.: *Výukový PACS na LF MU v Brně.* sborník Brněnské onkologicke dny. Edukační sborník, Brno 2005.

[17] Šárek, M.: *Komunikační infrastruktura pro eHealth.* II. mezinárodní konference Informační systémy a telekomunikační služby ve zdravotnictví, Brno 2005.

[18] Dostál, O.: *Vybrané právní aspekty využití informačních technologií v medicíně.* II. mezinárodní konference Informační systémy a telekomunikační služby ve zdravotnictví, Brno 2005.

[19] Javorník, M.: *MeDiMed.* elektronický sborník mezinárodní konference eHealth'05, URL: http://www.ehealth2005.no, Tromso, Norsko 2005.

[20] Reding, V.: *Introduction to the Ministerial Round Table.* elektronický sborník mezinárodní konference eHealth'05, URL: http://www.ehealth2005.no, Tromso, Norsko 2005.

# Regionální řešení zpracovávání medicínských obrazových informací

**Otto Dostál, Michal Javorník**

E-mail: otto@ics.muni.cz, javor@ics.muni.cz

**Klíčová slova:** PACS (Picture Archiving and Communication System), DICOM (Digital Image Communication in Medicine), MeDiMed (Metropolitan Digital Imaging System in Medicine)

**Abstrakt**

*Projekt MeDiMed (Metropolitan Digital Imaging System in Medicine) řeší problematiku sběru, zpracování a dlouhodobé archivace medicínských obrazových informací. Byl vybudován systém, který není pouze standardní PACS-ovou implementací, ale který od samotného počátku směřuje k sofistikovanějšímu řešení metropolitnímu a regionálnímu.*

*Od roku 1999 spolupracuje Masarykova univerzita s brněnskými nemocnicemi při zavádění informačních a komunikačních technologií v oblasti pořizování, přenosu, archivace a zobrazování digitálních medicínských obrazových informací. Tato spolupráce v rámci regionu představuje řadu koordinovaných aktivit a projektů směřujících k vybudování metropolitního archívu medicínských obrazových informací získávaných z nemocničních modalit, diagnostických zařízení, jako je ultrazvuk (US), digitální mamograf (DMG), magnetická rezonance (MR) a další, a jeho zpřístupnění prostřednictvím počítačové sítě. Cílem je využít možností současných ICT technologií a lékařské informatiky jak pro zvýšení kvality medicínské operativy a obecné lékařské péče, tak i zlepšení podmínek pro medicínský výzkum a výuku studentů.*

*V článku se zabýváme současným stavem, nabízenými možnostmi stávajícího řešení i dlouhodobými cíli, které si klade řešitelský kolektiv. Nejedná se pouze o technickou realizaci, ale i o snahu na základě zkušeností získaných při provozu tohoto rozsáhlého systému dosáhnout takových legislativních změn, které jsou nezbytnou podmínkou pro další rozvoj v této oblasti.*

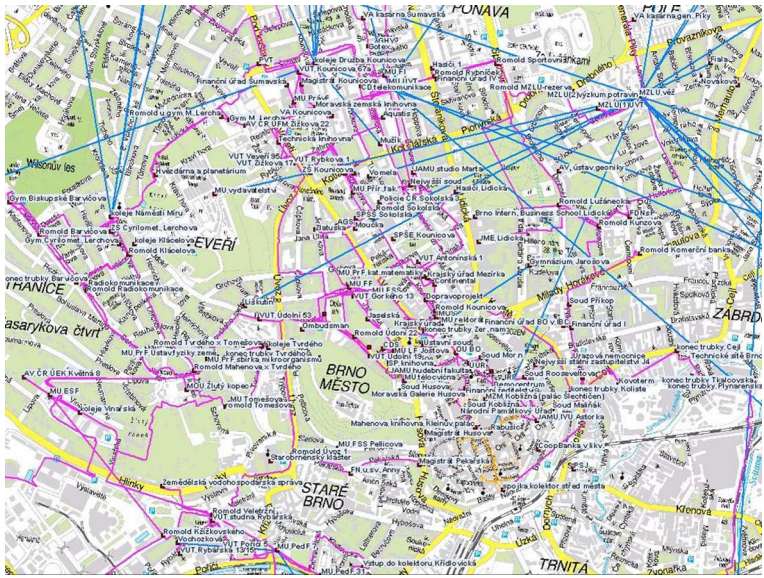
*Velmi významný je vývoj technologií pro řešení podpory výuky a výzkumu, od kterého se očekává významné zlepšení úrovně znalostí pregraduálních i postgraduálních studentů medicíny a začínajících radiologů ve zdravotnických zařízeních.*

*Technické řešení v maximální míře využívá dostupných technických prostředků doplněných o speciálně upravené komponenty a je koncipováno tak, aby současně splňovalo všechny požadavky kompatibility, které jsou kladeny na systémy pracující v reálném provozu.*

# 1 Projekt MeDiMed

Vybudování optické metropolitní akademické počítačové sítě bylo velmi výhodnou počáteční investicí pro rozvoj aktivit v oblasti zpracování medicínských dat. Sít v současné době zahrnuje více než 90 významných uzlů a celková délka optických kabelů dnes činí více než 100 km.

Počítačová síť propojuje nejen univerzity, pracoviště akademie věd, soudy, finanční úřady, ale i nemocnice a další zdravotnická zařízení. Optická vlákna jako přenosové médium podporují vysokorychlostní datové přenosy. Vyhrazená vlákna navíc umožňují i vysoce účinnou ochranu přenášených informací. Metropolitní akademická síť skýtá jedinečný prostor pro další rozvoj těchto aktivit [1].




Obr. 1

Projekt MeDiMed navazuje na výsledky předchozích aktivit a projektů:

- První projekt „Využití vysokorychlostních přenosů k integraci heterogenních multimediálních medicínských dat“ byl podán již v roce 1995. Vzhledem k finanční náročnosti nebyl přijat.
- V roce 1997 byl podán projekt s názvem „Rozvoj brněnské akademické sítě pro potřeby vědy a výzkumu“. Tento projekt byl schválen jako tříletý projekt 1998–2000. Řešil otázku spojení lokalit i vlastní realizaci řešení PACS. Řešení bylo dodáno a zprovozněno v roce 1999.

Interdisciplinární projekt MeDiMed řeší otázky medicínské využitelnosti, právní aspekty a problematiku vlastního technického zabezpečení. Hlavním cílem projektu je získávání informací o nutných přenosových kapacitách, objemech ukládaných dat, možnostech zpracování obrazových informací, možnostech obrazových informačních systémů, možnostech a omezeních v připojování jednotlivých vstupních informačních zdrojů (ultrazvuk, počítačová tomografie, magnetické rezonance, atd.).

Kromě archivace obrazových dat toto řešení zahrnuje i podporu přenosů obrazových informací mezi jednotlivými pracovišti (nemocnicemi), která pacient v průběhu léčby navštíví, s možností konzultací vzdálených specialistů. Výsledkem je usnadnění a urychlení formulace správné diagnózy, vyloučení opakovaných vyšetření, úspora času pacienta i lékaře a tím i finančních prostředků.

Mezi hlavní priority projektu patří získávání medicínských obrazových informací pro potřeby výzkumu a výuky.

## 2 Podpora výuky a výzkumu

Velmi významný je vývoj technologií pro podporu výukového a výzkumného podsystému, od kterého se očekává významné zlepšení úrovně výuky pregraduálních i postgraduálních studentů medicíny a začínajících radiologů ve zdravotnických zařízeních. Výukový systém je určen pro studenty lékařských oborů a pro začínající radiology nemocnic. Řešení je koncipováno tak, aby splňovalo požadavky kompatibility se systémy pracujícími v reálném provozu s cílem vytvořit pro uživatele prostředí, které se v zásadních aspektech neliší od reálných provozních systémů.

Před zařazením obrazové studie vhodné pro potřeby výuky a výzkumu do výukového systému jsou odstraněny (modifikovány) všechny informace, které by v budoucnu mohly vést k odhalení identity pacienta s ohledem na zachování maximální vypovídací schopnosti. U pacienta jehož obrazová informace byla zařazena do výukového systému z různých zdravotnických zařízení lze sledovat průběh onemocnění (léčby). Každá obrazová studie zařazena do výukové databáze musí být opatřena standardním popisem obsahujícím vlastní popis nálezu

případně vyšetřovacího postupu a sadou klíčových slov, které umožní pozdější snadné vyhledávání.

Databáze obrazových dat jednotlivých zdravotnických zařízení již existují, avšak pro výuku a výzkum se v současnosti využívají v minimálním rozsahu. Odstraněním bariéry, kterou v současnosti představují identifikační údaje pacienta a přidáním možnosti kontinuálního sledování nemoci pacienta i v případě léčby v různých zdravotnických zařízeních, tyto databáze skýtají obrovský potenciál využití pro výuku a výzkum.

# 3 Hlavní aktivity a směry dalšího rozvoje projektu

- Výukový a výzkumný systém je vyvíjen s cílem vytvořit pro uživatele prostředí, které se v zásadních aspektech neliší od reálných provozních systémů, s nimiž se mohou setkat na radiologických pracovištích zdravotnických zařízení. V roce 2004 byl dokončen vývoj anonymizačního modulu, který umožnil zahájit přenos vybraných anonymizovaných studií do databáze výukového a výzkumného systému.

- Zvyšování úrovně dostupnosti a spolehlivosti systému. Z tohoto důvodu zvyšujeme kvalitu centrálních uzlů v obou lokalitách:

  - centrální pracoviště pro podporu rutinního provozu v lokalitě Botanická 68a,
  - záložní pracoviště pro podporu rutinního provozu a výzkumné a výukové centrum v lokalitě Komenského náměstí.

- Ve spolupráci s dodavateli systémů PACS jsou realizovány další práce na úpravách systému přístupových práv komunikačního modulu, umožňující přesnější směrování obrazové studie na konkrétní pracoviště nemocnice případně na konkrétní diagnostickou stanici. Tyto úpravy přispívají ke zvýšení úrovně zabezpečení citlivých dat v systému [2].

- Zvyšování úrovně bezpečnosti. Jedná zejména o zabezpečení medicínských informací před možným zneužitím. Jde o velmi citlivou oblast, která je z hlediska nemocnic velmi sledovanou a požadavky na zabezpečení dat, v místech jejich vzniku i při přenosu a archivaci, jsou velmi striktní.

- Pokračují aktivity umožňující připojování vzdálených pracovišť (optická vlákna, rádiová připojení, využití satelitů, atp.). Předpokládáme další nárůst počtu připojených modalit a prohlížecích stanic, propojení s dalšími PACS systémy spolupracujících nemocnic regionu. Pro připojení nových

lokalit je určující zejména cena a technické možnosti potenciálních přenosových tras a připravenost zdravotnických zařízení.

- Vytvoření podpůrného legislativního rámce, který spolupráci zdravotnických subjektů při výměně (nejen) digitální obrazové informace bude podporovat a nikoliv blokovat. V roce 2004 podstoupili řešitelé řadu jednání o souvisejících legislativních otázkách, včetně vystoupení na půdě parlamentu ČR s předsedou komise pro zdravotnictví a pozdějším ministrem zdravotnictví a jednání s ministrem informatiky o potřebných změnách v legislativě. Pro Ministerstvo informatiky byl zpracován materiál, který upozorňoval na přetrvávající problémy nejasnosti v této oblasti. Na základě dosavadních jednání byli řešitelé vyzváni ke zpracování dalších podkladů. Budeme se i nadále snažit dostávat na půdu parlamentu požadavky na změny a úpravy relevantních zákonů.

- Informace ze zahraničí budeme i nadále získávat jednáním na významných mezinárodních akcích [3, 4]. Významnou akcí roku 2005 byla prezentace projektu MeDiMed na konferenci eHealth za účasti ministrů zdravotnictví států EU.

- Hledání finančních zdrojů pro rozvoj systému (jak v rozpočtu jednotlivých nemocnic, tak i v tuzemských a zahraničních grantových programech). Současně však také hledání optimální rovnováhy mezi požadavky, technickými možnostmi a cenou řešení v jednotlivých oblastech nasazení. K tomu je nezbytná úzká spolupráce se specialisty v příslušných oblastech medicíny, neboť jen ti jsou například schopni určit, jaké zobrazovací jednotky jsou ještě vhodné a dostatečné pro daný účel a danou modalitu.

Současné aktivity jsou hrazeny z následujících projektů:

a) „*Autentizovaný přístup k službám metropolitního archivu medicínské obrazové informace*“, CESNET z.s.p.o (2004–2005),

b) „Rozvoj výuky klinických oborů moderními informačními technologiemi“, Lékařská fakulta Masarykovy univerzity v Brně (2005),

c) „*Optická síť národního výzkumu a její aplikace*“ v aplikační oblasti *Telemedicína*, výzkumný záměr CESNET z.s.p.o. (od roku 2004),

d) „*Efektivní zpracování medicínských obrazových dat*“, v rámci programu *Informační společnost* Akademie věd ČR (2005–2008),

e) „*Healthware*“ v rámci IST programů 6. rámcového programu EU (2005–08).

# 4 Závěr

Smyslem projektu je využít možností současných IS/ICT k efektivnímu zpřístupnění multimediálních dat pro potřeby zdravotnické operativy i pro výzkumné instituce a univerzity.

V oblasti medicínské operativy se jedná především o podporu přenosů informací mezi jednotlivými pracovišti (nemocnicemi), která pacient v průběhu léčby navštíví, včetně možnosti on-line konzultací vzdálených specialistů a rychlý přístup ke sdíleným databázím obsahujícím kritická data.

V oblasti podpory výuky a výzkumu pak vytvořit pro uživatele prostředí, které se v zásadních aspektech neliší od provozních systémů na radiologických pracovištích a strukturované databáze anonymizovaných obrazových studií pokrývající jednotlivé oblasti medicíny.

# Literatura

[1] Dostal, O., Filka, M., Petrenko, M.: *University computer network and its application for multimedia transmissin in medicine.* WSEAS Int. Conf. on Information Security, Harware/Software Codesign, ECommerce and Computer Networks, Rio de Janeiro, Brasil. WSEAS 2002, 1 961–1 964.

[2] Dostal, O., Javornik, M., Slavicek, K.: *MEDIMED – Regional Centre for Archiving and Interhospital Exchange of Medicine Multimedia Data.* Proceedings of the Second IASTED International Conferee on Communications, Internet and Information Technology. Scottsdale, Arizona, USA: International Association of Science and Technology for Development – IASTED, 2003, p. 609–614. ISBN 0-88986-398-9

[3] Schmidt, M., Dostal, O., Javornik, M.: *MEDIMED – Regional PACS Centre in Brno, Czech Republic.* Proceedings of the 22th International Conference of EuroPACS & MIR (Managenment in Radiology) Conference, 16–18 September, Trieste, Italy.

[4] Dostal, O., Javornik, M.: *MEDIMED – Regional educational and research centrte for processing of medical image information Centre in Brno, Czech Republic.* Computer Assisted Radiology and Surgery, Elsevier publ., Berlin, Germany 2005. ISSN 0531-5131

# Algorithms in medicine, their value, limitations and perspectives

**Jan Vejvalka**

E-mail: jan.vejvalka@lfmotol.cuni.cz

**Abstract**

*To standardize and improve quality, various types of algorithms are being introduced into health care. Ranging from general clinical guidelines that define the strategy of care in typical situations to very detailed formulae for calculations of e.g. drug dosage adjusted to kidney functions, use of algorithms is positively accepted and becomes a standard. The aim of this article is to demonstrate to the more technical audience some of the specific aspects of use of algorithms in medicine.*

## 1 Medicine as an information-processing activity

Placed somewhere on the border of art, craft and science, medicine has always been based on information processing. With the aim to help patients' health, doctors have since ever been observing their patients, listening to their complaints, searching for significant facts in their previous histories and in their environments and then comparing these bits of specific information with the sum of general knowledge of medicine, as it is in each case represented in doctors' individual knowledge and experience. The task of medical science in this endeavour is to assure (and explain) reproducibility of processes and results. From this perspective, clinical medicine as applied science is based on development and application of operational algorithms, be they called clinical guidelines, standards of care or clinical protocols.

## 2 Lege artis medicinae

Traditionally, medical schools have been the place where medical knowledge was institutionally taught and graduated doctors were obliged to adhere to this knowledge. Medical chambers are still keeping their governance over doctors following approved standards of care. Contents and representation of these approved standards vary in place and time, and they can be seen as the scientific base for medical practice. Lege artis medicinae, according to the rules of the art of medicine, is the way doctors work. Non lege artis is a nightmare: overlooking or mis-interpreting an important sign, making an error in diagnosis or treatment, or, more recently, not having read about the current procedures – all this can be under some conditions seen as professional misconduct and can have severe consequences. The rule of the profession says that treatment should be based on contemporary scientific knowledge. Similarly to judges who in their judgements apply the general rules of law to specific cases of law suits, doctors apply the general knowledge of medical science to specific cases of their patients' health conditions. Unlike law, where sources of general rules are defined and stable (at least in Czech law), medical science has much broader base of general knowledge, that also develops much more rapidly. The number of published articles is beyond comprehension: 1 500–3 500 articles daily are indexed by Medline, the reference bibliographic database maintained by the U.S. National Library of Medicine [1].

## 3 Evidence-based medicine

Traditionally based on experience and observation, with some reserve towards experiments, medical science must take into account the complex and stochastic nature of processes it deals with, the uncertainty of any observations and, consequently, a degree of uncertainty inherent to devised knowledge. To handle this uncertainty, to be at least able to give it a measure, evidence-based medicine classifies the background that underlies specific bits of medical knowledge. Knowledge backed by sound statistical evaluation e.g. of a double-blinded clinical trial (as in drug testing: neither the patient nor the doctor knows which substance the patient gets) is considered more relevant than consensus of experts [2]. Clearly, there are not too many pieces of knowledge supported by double-blinded trials.

## 4 Guidelines, algorithms and ambiguity

In order to make operational knowledge in certain areas of medicine more comprehensive, various bodies and institutions have over the past decade published

recommendations in forms of clinical guidelines. These guidelines can be seen as algorithms to be performed by doctors when their patients meet some input conditions. The relevance of the respective recommendations varies, as vary the local conditions: availability of various investigations and procedures is a typical factor that affects portability of guidelines between systems of health care and between institutions. Guidelines are being published at various scopes (national, regional, hospital), supported usually by consensus of relevant experts, and usually cover only those conditions where consensus is possible and justified by sufficient evidence. The scope of guidelines affects the level of detail which they provide. The aim of clinical practice guidelines is to assure certain quality of the process of health care while containing its costs, and publication of guidelines, as an expression of accepted responsibility, has often a political dimension.

An example of a clinical guideline represented as a flow chart is presented in Fig. 1.

The flow chart is accompanied by several pages of comments that describe the various conditions and procedures and by a various number of references to supporting medical literature. Individual elements of the algorithm (decisions and procedures) are described at a very high level of generalization, leaving up to the doctor e.g. to choose the right drugs for recommended therapies, to choose (based on calculations) appropriate dosage and, most importantly, to reflect all other conditions that could affect patient's state or responsiveness to the recommended therapy.

Despite all the ambiguity and uncertainty inherent to these recommendations, they represent a step towards standardization of quality and containment of costs in health care. Guidelines recommended by authoritative bodies are often adopted by others and adapted to local conditions. To facilitate these portations, and also to create ways to cmputer-aided execution of the high-level algorithms contained in guidelines, projects in health informatics have been studying guideline structure, developing notations designed to describe various situations in medicine (e.g. Arden Syntax), formats for representation of guidelines (GLIF, Guideline Interchange Format) etc. [3, 4, 5, 6].

# 5 Computational Algorithms

Computational algorithms usually represent a different level of medical knowledge; sometimes they can be seen as building blocks for the more complex algorithms of clinical care. Indeed, if clinical guidelines can be compared to top-down programming, computational algorithms represent a bottom-up approach, taking bits of specific knowledge about specific clinical situations and transforming them into more complex assessments that can either serve as input to guideline-based decisions (e.g. assessment of patient renal functions based on values of

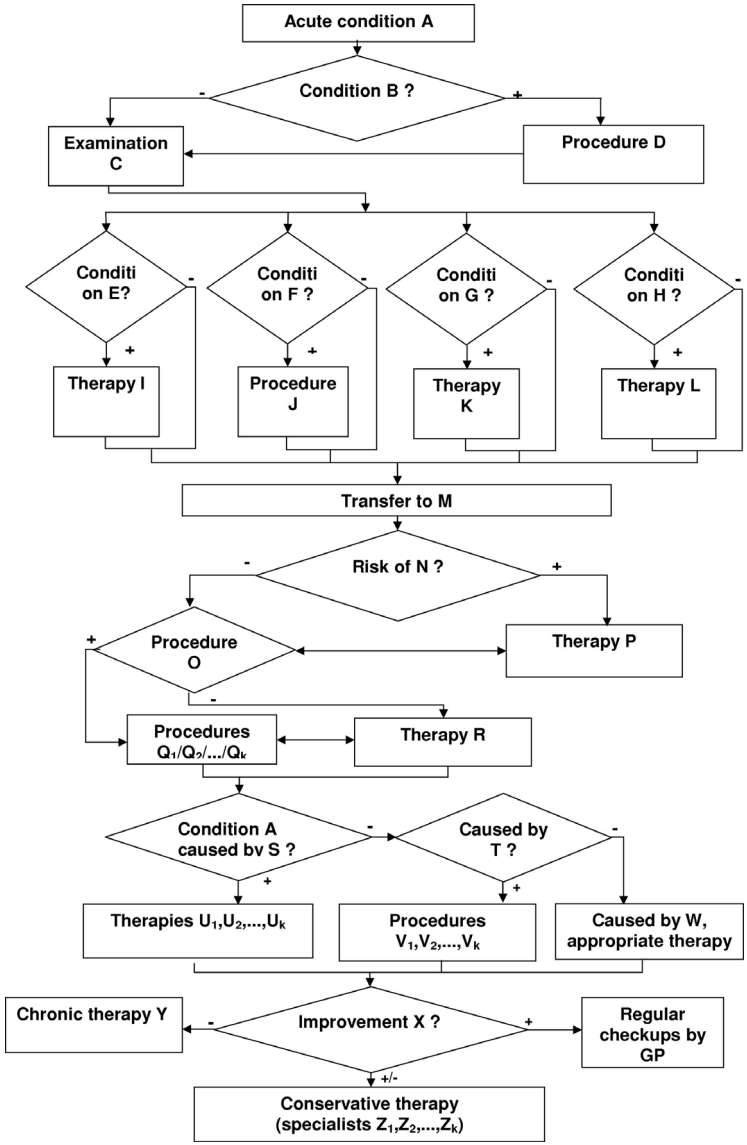



Figure 1 Flow chart of a clinical guideline

various substances in blood serum and in urine) or as a more detailed recommendation about specific procedures to be followed (as e.g. drug dosage based on patient body surface and further parameters). These algorithms have a longer tradition than clinical guidelines and are less dependant on local conditions. A number of simple or more complex calculators that execute these algorithms has been created since the first introduction of computing into medicine, and these calculators have usually a good acceptance by users who see them as a help to their routine work [7, 8]. Evidence that supports computational algorithms is more variable than that supporting clinical guidelines: tasks that are aided by these algorithms (drug dosage, e.g.) must be performed irrespectively of the value of underlying evidence. In this specific example, the authoritative information for drug dosage is usually the manufacturer's recommendation. The formula for calculation of body area is based on a number of measurements which were then correlated to patient weight and height – parameters that are easily accessible.

### 5.1 Scoring systems

A specific class of computational algorithms are scoring systems. Developed for classification of severity of patients' conditions, early scoring systems were based on observations – as e.g. the most famous Apgar score for evaluation of newborn infants.

More than 50 years after its introduction, Apgar score (a simple sum of simple scores that on a 0–1–2 scale assess heart rate, respiratory efforts, color, muscle tone and reflex irritability) is still used to report newborn status [9].

Recent scoring systems are constructed in a more sophisticated way: based on large statistics they classify patients according to expected outcome of care or costs of care. These scoring systems can then be used to measure effectivity of applicable clinical protocols (that may later be acknowledged as guidelines) and also to compare outcomes or costs of care across various institutions and care systems. A well known example of these is the Apache score (used routinely in intensive care units) that uses a still relatively simple procedure to determine from 34 variables in 8 groups the probability that the patient will die in the hospital within the current episode of care [10].

## 6 Ambiguities and contradictions of medical algorithms

Where uncertainty is one of the basic characteristics of all processes, be it caused by influence of many (yet) unknown factors or by the limited means of their observation, ambiguous and sometimes contradictory pieces of knowledge are no

surprize. Contradictory views of different experts are as legitimate as contradictory local guidelines governing the procedures of care in different institutions. More importantly, any recommendations reflect the state of medical science at a given time, and guidelines therefore have to be maintained and updated – which is often a more demanding task than their formulation.

A similar situation of ambiguity exists in computational algorithms: as an example, at least 5 different formulae exist to calculate the body area depending on body height and weight, the oldest of them dating back to 1910's. These formulae are based on statistics of different numbers of measurements, and they result in slightly different estimates. All of them neglect possible influence of other factors (than weight and height), e.g. factors that would reflect body composition. Just as it is the responsibility of the doctor to assess applicability of recommended clinical algorithms to a specific case, it is the responsibility of the user of a clinical calculator/formula to assess its possible limitations. Medical literature is accepted as a relevant source of information; therefore a proper search in published literature should be done prior to introduction of any algorithm to justify its use for any specific clinical situation. As an example, after some discussions, the Apache score mentioned above has undergone two major revisions since its original release in and several mionor modifications have been published [11, 12].

# 7 Data entering medical algorithms

Besides the basic judgement if a specific algorithm is suitable for a specific clinical situation, the second step before an algorithm can be applied is to ensure that relevant and valid data is available – if the available data meets the requirements of the algorithm. With stand-alone clinical calculators this is the task of the user, and the process of algorithm and data selection and validation is fully under his/her control. With calculators that should be used routinely and automatically, data selection is not that simple.

## 7.1 Computing at the source of data

Decisions whether a specific algorithm is suitable for specific data can be relatively easy at the source of data. Biochemical analyzers, therefore, usually come with software that besides controling the analyzer offers certain calculations on the measured data. Similarly, image processing algorithms are being implemented in medical imaging devices. Manufacturers' choice of algorithms reflects their expertise in the respective field and is documented in device manuals. Still it is up to the experienced user to assess the individual clinical situation and accept or reject the suggested post-processing of measured data.

## 7.2 Computing on data in electronic health records

Electronic health records are a natural source of relevant data that can be used for furher calculations. Compared to calculations embedded in medical devices, it is a more difficult task to assure that data on input of a medical algorithm are consistent, i.e. that it makes sense to combine them for a calculation. Further, medical algorithms usually represent a piece of highly specialized knowledge that is quite far from the main field of activities of hospital information systems' makers. It may be for these two reasons that hospital and GP information systems vendors have traditionally been rather reluctant in offering routine processing of data with specialized algorithms as part of their systems. Only recently specialized modules (developed together with specialists in e.g. cardiology, diabetology etc.) are being introduced in clinical information systems where expert users are likely to benefit from them – and also to assess their possible limitations. Similarly to clinical guidelines, also these modules have to be maintained and updated to reflect the state of clinical science in the respective fields.

Variability of formats is one of the characteristics of health data. While laboratory results and measurements are generally stored in a structured way, clinical observations are usually stored as free text descriptions. Of the 5 parameters that constitute Apgar score, e.g., only one is a measurement (and could be taken from health record directly). One of them, moreover, requires a specific action to be taken in order to observe the neonate's reaction. Implications for inclusion of such algorithms into routine information systems are clear:

- for data kept in a structured way, data definitions (and formats) must be taken care of to fit (database/algorithm alignment)
- for data recorded as free texts, either some text mining must be used or new structured data elements must be defined
- for data specific to certain algorithms, data entry prior to algorithm execution must be assured.

# 8 Repository of medical algorithms – project MEDAL

The aim of MEDAL, a project developed currently at the Institute for Algorithmic Medicine in Texas, USA, is to make a representative collection of medical algorithms readily available in a practical format to clinicians, educators and researchers [ww].

Specifically, the goals of MEDAL are to enhance the use of medical algorithms by providing:

1. A comprehensive collection of accurate and reliable algorithms
2. Adequate documentation with references to the original sources
3. Standardization of data elements, to enable automation of input and output
4. Indexing and linking for quick access and retrieval

At present, MEDAL contains more than 7 000 algorithms organized into 45 chapters that reflect the various application fields. Algorithms are represented in a structured way that covers algorithm description, description of input parameters, the algorithm itself (presented also in an Excel sheet) and the key reference. Several algorithms have also been transformed into web-based calculators available to registered users of the project [13, 14].

In a co-operation between the University of Texas, Houston, with CESNET, a European mirror of MEDAL is being set up, also as a base for further work in representation of medical algorithms.

# 9 GRID-based execution of medical algorithms – project MediGRID

With the aim to design, develop and test a GRID-based environment for processing of health-related data, project MediGRID was launched in co-operation of two Czech hospitals and CESNET, the Czech national research network operator. Using an approach based on ontologies to describe medical data, their meanings, relations and possible use in medical algorithms, the project creates an environment where data and algorithms (more generally: knowledge) are treated as resources, giving users and user applications tools to navigate in this environment and to use the resources it offers. At the end of its first year, the project is testing its ontology-based approach on several algorithms selected from the application area of paediatrics. The algorithms are being implemented as grid resources (web services) and put together into applications that perform useful and non-trivial clinical calculations [15].

# 10 Conclusion

Use of algorithms in medicine is gaining acceptance as a clear step towards standardization and improvement of quality of care and towards containment of costs by avoiding unnecessary procedures and investigations. While execution of algorithms can be relatively straightforward, two crucial questions are left to doctors' decision before an algorithm is used:

- is there an algorithm suitable for this specific situation?
- is the available data valid for this specific algorithm?

Neither of these questions is trivial. To make these decisions easier is one of the current challenges to medical informatics: to find

- suitable formats of algorithm representation
- appropriate ways to bind them with underlying knowledge
- ways to bind them with unambiguous descriptions of data they require.

# References

[1] *Fact Sheet MEDLINE.* First published Nov. 17, 2004, updated Jun 8, 2005. http://www.nlm.nih.gov/pubs/factsheets/medline.html.

[2] Akobeng, A.K.: *Principles of evidence based medicine.* Arch Dis Child. 2005 Aug; 90 (8): 837–40.

[3] Hripcsak, G.: *Arden Syntax for Medical Logic Modules.* MD Comput. 1991 Mar–Apr; 8 (2): 76, 78.

[4] Patel, V.L., Allen, V.G., Arocha, J.F., Shortliffe, E.H.: *Representing clinical guidelines in GLIF: individual and collaborative expertise.* J Am Med Inform Assoc. 1998 Sep–Oct; 5 (5): 467–83.

[5] Boxwala, A.A., Greenes, R.A., Deibel, S.R.: *Free Full Text Architecture for a multipurpose guideline execution engine.* Proc AMIA Symp. 1999: 701–5.

[6] Georg, G., Colombet, I., Jaulent, M.C.: *Structuring Clinical Guidelines through the Recognition of Deontic Operators.* Stud Health Technol Inform. 2005; 116: 151–6.

[7] Hoffer, E., Akria, L., Tabak, A., Scherb, I., Rowe, J.M., Krivoy, N.: *A simple approximation for busulfan dose adjustment in adult patients undergoing bone marrow transplantation.* Ther Drug Monit. 2004 Jun; 26 (3): 331–5.

[8] Maitre, P.O., Shafer, S.L.: *A simple pocket calculator approach to predict anesthetic drug concentrations from pharmacokinetic data.* Anesthesiology, 1990 Aug; 73 (2): 332.

[9] Apgar, V.: *A proposal for a new method of evaluation of the newborn infant.* Anesth Analg. 1953; 32: 260.

[10] Knaus, W. A., Zimmerman, J.E. et al.: *APACHE – acute physiology and chronic health evaluation: a physiologically based classification system.* Crit Care Med. 1981; 9: 591–597.

[11] Knaus, W.A., Draper, E.A. et al.: *APACHE II: A severity of disease classification system.* Critical Care Medicine. 1985; 13: 818–829.

[12] Knaus, W.A., Wagner, D.P. et al.: *The APACHE III prognostic system: Risk prediction of hospital mortality for critically ill hospitalized adults.* Chest. 1991; 100: 1 619–1 636.

[13] Kantor, G., Svirbely, J.R., Johnson, K., Sriram, M.G., Rodrigueze, J.R., Smith, J.: *MEDAL: The Medical Algorithm Project.* MEDINFO 2001, London.

[14] *MEDAL, The Medical Algorithms Project.* http://www.medal.org. Release 15.1, June 2005.

[15] Vejvalka, J., Lesný, P., Holeček, T., Bouzková, H.: *Klasifikační systémy, ontologie a GRID.* MEDSOFT 2005, Beroun.

Supported by MSM6383917201 (Optická síť národního výzkumu a její nové aplikace) and by 1ET202090537 (MediGRID).