VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PRINCIP HLASOVÉ KOMUNIKACE V IP SÍTÍCH A JEJÍ BEZPEČNOST

VOICE-OVER-IP PRINCIPLE AND SECURITY PROBLEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. PETR BOŘUTA
AUTHOR

VEDOUCÍ PRÁCE Ing. JOSEF VYORAL
SUPERVISOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor

Telekomunikační a informační technika

Student:	Bořuta Petr Bc.	***ID:***	89718
Ročník:	2	***Akademický rok:***	2007/2008

NÁZEV TÉMATU:

Princip hlasové komunikace v IP sítích a její bezpečnost

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte standardy a protokoly používané v technologii hlasové komunikace v IP sítích, především se soustřeďte na principy zabezpečení. Názorným způsobem prezentujte proces sestavení, udržování a ukončení relace VoIP pomocí SIP protokolu. Vytvořte pokusnou síť pro realizaci VoIP spojení v laboratoři FEKT. Na takto realizované spojení aplikujte vybrané útoky využívajících bezpečnostních nedostatků použitých protokolů, postupy těchto útoků popište v teoretické části práce. Závěrem zhodnoťte bezpečnostní rizika SIP relace.

DOPORUČENÁ LITERATURA:

[1] Novotný, V., Účastnická koncová zařízení. Vysoké učení technické v Brně, nakladatelství VUTIUM, 2002.

[2] Endler D., Collier M., Voice Over IP Security Secrets & Solutions. ISBN 9780072263640, McGraw, 2007.

Termín zadání:	11.2.2008	***Termín odevzdání:***	28.5.2008

Vedoucí práce: Ing. Josef Vyoral

prof. Ing. Kamil Vrba, CSc.

předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práve třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Petr Bořuta

Bytem: bří Jaroňků 5217, 76001, Zlín

Narozen/a (datum a místo): 10.6.1984, Zlín

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 60200 Brno 2

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   ☐ disertační práce

   ☑ diplomová práce

   ☐ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Princip hlasové komunikace v IP sítích a její bezpečnost

Vedoucí/školitel VŠKP: Ing. Josef Vyoral

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

☑ tištěné formě - počet exemplářů 1

☑ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Tato práce se zaměřuje na technologii VoIP a její zabezpečení. V první části je popsán obecný princip a struktura VoIP systémů a jednotlivé protokoly svázané s VoIP. V další části je rozebráno zajištění kvality služeb a objasněn proces SIP komunikace. V poslední části jsou uvedeny bezpečnostní rizika VoIP protokolů. V praktické části je popsáno vytvoření pokusné VoIP sítě, na kterou bylo praktikováno několik útoků a následně i metodika zabezpečení. V závěru jsou zhodnoceny bezpečnostní problémy při VoIP komunikaci.

Klíčová slova

VoIP, SIP, bezpečnost, útoky, Asterisk

Abstract

This master's thesis deals with security properties of protocols used for VoIP systems. In the first part, there is a description of most commonly used protocols and structure of VoIP systems. This part also discuss signaling and transport protocols. The second part of this paper describes techniques of ensuring quality of services. The next part presents SIP messages and communication. Last part of this paper overviews security risks of VoIP protocol. Practical part of this thesis describes creation of a testing VoIP network, on which several attacks has been made, fallowed by securing of mentioned VoIP network. Result of this thesis is evaluation of security risks connected to VoIP communication.

Keywords

VoIP, SIP, security, attack, Asterisk

Bibliografická citace mé práce:

BOŘUTA, P. *Princip hlasové komunikace v IP sítích a její bezpečnost.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 75s. Vedoucí diplomové práce Ing. Josef Vyoral.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Princip hlasové komunikace v IP sítích a její bezpečnost jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne . . . .

(podpis autora)

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Josefovi Vyoralovi za pomoc při zpracování diplomové práce. A dále slečně Bc. Zdence Ševčíkové za rady při finálním zpracování práce.

Obsah:

Úvod **9**
1 Technologie VoIP **10**
1.1 Protokoly 12
1.1.1 TCP 13
1.1.2 UDP 14
1.1.3 RTP 14
1.1.4 RTCP 15
1.2 Standardy VoIP 15
1.2.1 SIP 16
1.2.2 H.323 18
1.2.3 IAX 21
1.2.4 MGCP 22
1.2.5 MEGACO (H.248) 22
1.2.6 SCCP 23
1.3 Zvukové kodeky 23
2 Kvalita služeb **25**
3 Průběh SIP **28**
3.1 Žádosti 28
3.2 Odpovědi 28
3.3 Identifikace 29
3.4 Příklad spojení 30
3.4.1 Registrace 31
3.4.2 Navázání a ukončení spojení 32
4 Zabezpečení **35**
4.1 Typy útoků a zabezpečení 35
4.1.1 Zabezpečení signalizace SIP 36
4.1.2 Zabezpečí vlastního hovoru 39
4.2 Zapojení pokusné sítě 40
4.2.1 Instalace a nastavení 41
4.3 Útoky 45
4.3.1 Mapování sítě a zařízení 45
4.3.2 Zachycení RTP 53
4.3.3 Denial of service 54
4.3.4 Man in the middle 56
4.4 Ochrana 59
4.4.1 ZRTP 60
4.4.2 VLAN 62
4.4.3 IDS 65
4.4.4 TLS 67
4.4.5 VPN 68
5 Závěr **70**
Literatura **72**
Příloha A: Odpovědi SIP **74**

Seznam obrázků

Obrázek 1: Základní schéma VoIP 10
Obrázek 2: Použití protokolů 11
Obrázek 3: Srovnání telefonních sítí 12
Obrázek 4: Model OSI 13
Obrázek 5: Relace mezi servery 18
Obrázek 6: Struktura H.323 sítě 19
Obrázek 7: Jitter buffer 26
Obrázek 8: Registrace klienta 32
Obrázek 9: Ukázka SIP spojení 33
Obrázek 10: Použití redirect serveru 34
Obrázek 11: SIPS 38
Obrázek 12: IPsec 39
Obrázek 13: SRTP 40
Obrázek 14: Schéma zapojení 41
Obrázek 15: Program SIPSCAN 50
Obrázek 16: Útok na hesla přes Cain&Abel 52
Obrázek 17: Záznam RTP přes Cain&Abel 53
Obrázek 18: TCP Handshake 54
Obrázek 19: Man in the middle 57
Obrázek 20: ARP cache poisoning 58
Obrázek 21: Podvržená MAC adresa 58
Obrázek 22: MAC flooding 59
Obrázek 23: Zapojení SIP Serveru 61
Obrázek 24: Průběh stavů Zfone 62
Obrázek 25: Průběh ZRTP 62
Obrázek 26: VLAN 63
Obrázek 27: Rozdělení do VLAN 65
Obrázek 28: VPN 68

Seznam tabulek

Tabulka 1: Zvukové kodeky 23
Tabulka 2: Druhy zpoždění 26

Úvod

Už od počátku lidské společnosti je potřeba mezilidské komunikace jednou ze základních potřeb. Vývoj v oblasti komunikace se rozvíjí velikou rychlostí. Staré technologie jsou nahrazovány novými, které přináší kombinaci snazšího, srozumitelnějšího, rychlejšího a levnějšího sdělování. Vynález telegrafu a telefonu jsou základními kameny ve spojení na větší vzdálenosti než nám umožňuje lidský hlas.

Voice over Internet Protocol (VoIP) je technologie, která umožňuje telefonování prostřednictvím počítačové sítě. Přenáší digitalizovaný hlas prostřednictvím internetových protokolů. Umožňuje spojení s klasickou telefonní sítí. Jejímu širšímu rozšíření brání nedostatek kvalitního připojení k internetu v domácnostech, obava před velkou změnou při přechodu na novou technologii a nižší kvalitou IP telefonie [1].

První část práce je věnována technologii VoIP. Ve stručné formě jsem se snažil popsat jednotlivé protokoly, problémy i možnosti uplatnění VoIP systémů. Dále jsem se zaměřil na problematiku zajištění kvality služeb. Vzhledem k dominanci protokolu SIP jsem následně rozebral způsoby SIP komunikace včetně grafického znázornění.
V poslední části práce jsem popsal svůj návrh pokusné VoIP sítě. V začátcích VoIP byla bezpečnost neprávem opomíjena, ale v současné době zažívá rozmach. Názorně jsem popsal postupy jednotlivých útoků na VoIP a uvedl jednotlivá možná opatření zabraňující narušení bezpečnosti hlasové komunikace.

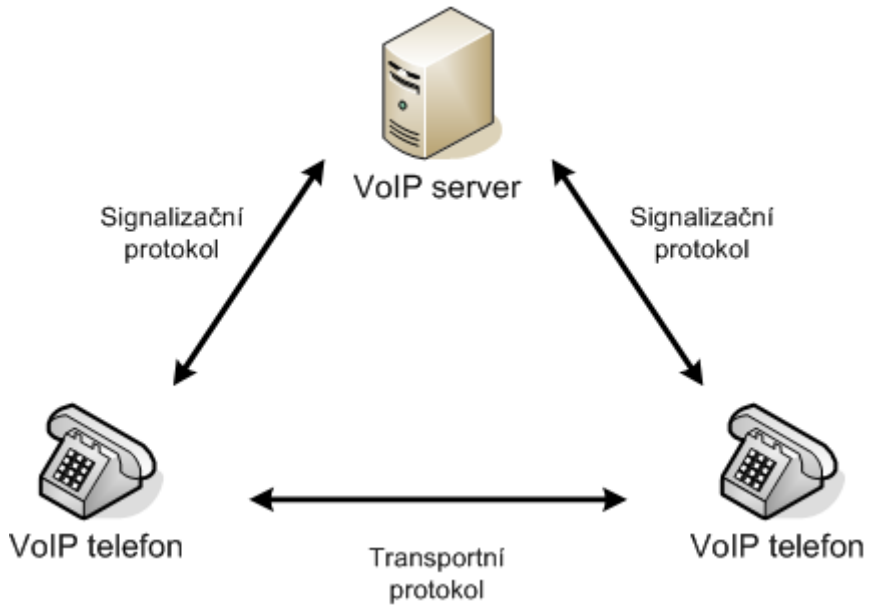
1 Technologie VoIP

VoIP je technologie, která nám umožňuje přenášet zvukovou informace mezi dvěma a více účastníky v reálném čase po datové síti založené na IP protokolu. Tato technologie je alternativou pro klasickou telefonní síť, označovanou Public Switched Telephone Network (PSTN).

VoIP se od PSTN liší v mnoha směrech. PSTN pro svůj provoz a přenos dat používá vlastní, uzavřenou síť, která byla navržena pouze pro potřeby telefonních služeb. VoIP nezavádí žádnou vlastni novou síť, ale používá již existující datovou síť. PSTN je obvodově přepínanou sítí, informace v lince je přenášena jako nepřetržitý digitální tok. Naproti tomu ve VoIP jsou všechny informace přenášeny v podobě paketu. Hlasová informace je nejprve digitalizována a upravena za pomoci zvukového kodeku, nejčastěji jsou to standardy rodiny G.7xx nebo některý z otevřených internetových formátů (viz kapitola 1.3). Takto digitalizovaná a upravená informace je rozdělena do paketu a odeslána do IP sítě. Na straně příjemce je pak hlas opět sestaven do původní podoby a interpretován (Obrázek 1).

Obrázek 1: Základní schéma VoIP

Ve VoIP se používají dvě skupiny protokolů (Obrázek 2). První skupina protokolů, signalizační, slouží pro navázaní spojení s volaným a dohodnutí se na pravidlech přenosu. Pro navázaní spojeni se používá proxy server, který má za úkol kontaktovat volaného a umožnit sestavení hovoru mezi účastníky. Mezi tyto protokoly patří například H.323 nebo SIP. Pro přenos zvukové informace se v IP sítích používá aplikační protokol RTP založený na transportním protokolu UDP.

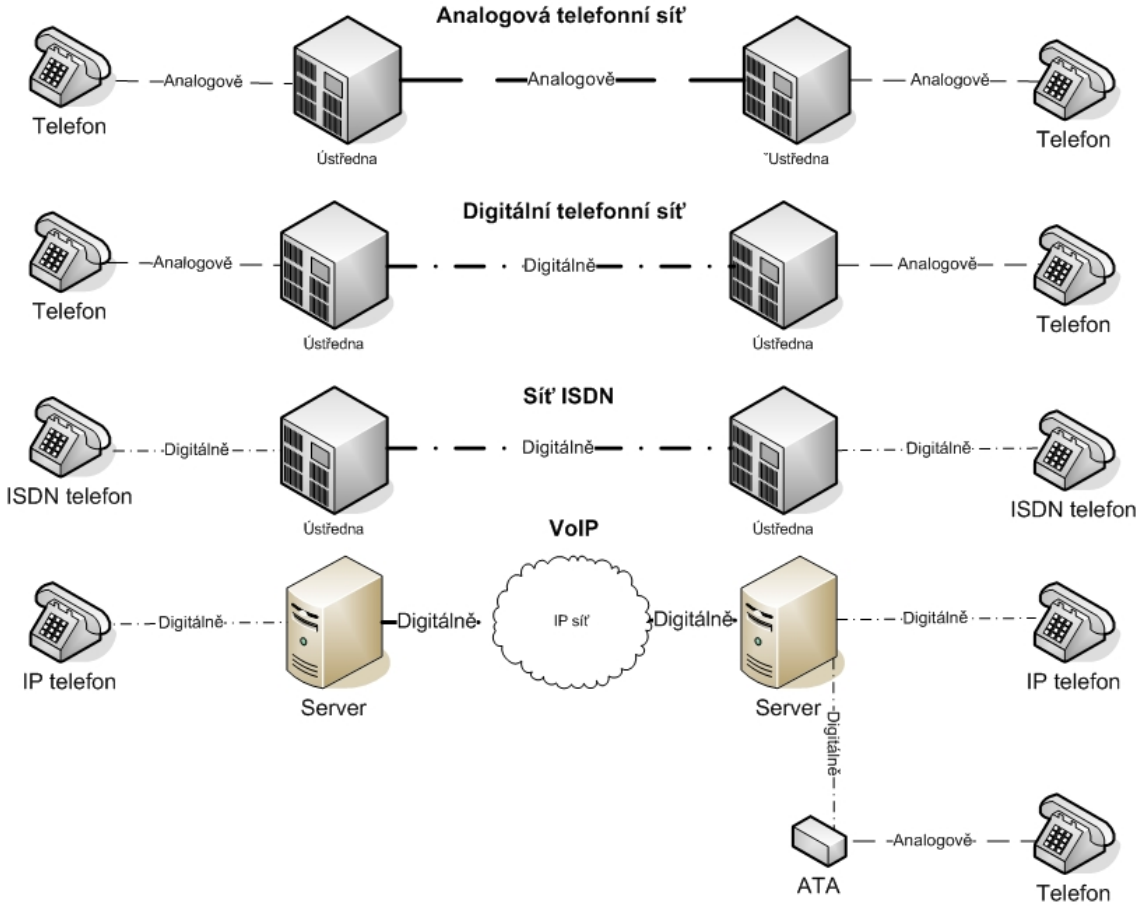


Obrázek 2: Použití protokolů

Vývoj telekomunikačních a počítačových sítí ukázal potřebu spojit tyto dvě sítě v jeden funkční celek (Obrázek 3). Tento proces, zvaný konvergence, označuje postupné přibližování navzájem odlišných komunikačních technologií. Datové sítě, které součastně podporují hlasové a obrazové služby, jsou označovány jako konvergentní sítě. Obecně se prosadily dvě metody konvergence.

První způsob představoval snahu telekomunikačních operátorů vytvořit sítě s integrovanými hlasovými a datovými službami, označované ISDN (Integrated Services Digital Network). ISDN vychází z telekomunikačního způsobu komunikace přes spojově orientované sítě. Zdlouhavý proces standardizace ISDN však zpomalil její nasazování a dnes již neodpovídá součastným komunikačním nárokům.

Další metoda konvergentní IP sítě představuje úplně odlišnou filozofii. Jejím základem je využití nespojově orientovaných paketových sítí založených na protokolu IP k implementaci telekomunikačních služeb. Toto řešení se ukázalo jako více perspektivní a vzhledem ke globálnímu rozšíření internetu je rozšířenější. IP sítě však musely být doplněny o pomocné technologie a protokoly umožňující přenos obrazu a zvuku v prostředí bez garance kvality služeb.



Obrázek 3: Srovnání telefonních sítí

1.1 Protokoly

Realizace datové komunikace mezi koncovými body sítě vyžaduje vykonání určitých procesů ze strany sítě. Jedná se například o navazování spojení, rozklad přenášených dat do bloků, přenos bloků, jejich zabezpečení proti chybám apod. V minulosti si jednotlivý výrobci vytvářeli svoje vlastní řešení pro vzájemnou komunikaci mezi prvky jednoho výrobce. Vznikaly tím tedy různé architektury, které mezi sebou neuměly komunikovat. Vznikala tedy snaha o standardizaci, která by umožnila bezbariérovou komunikaci. Postupným vývojem se ukázalo jako účelné rozčlenit komunikaci na jednotlivé vrstvy. Každá vrstva poskytuje služby vrstvám vyšším a zároveň se snaží zakrýt implementaci těchto služeb ve vrstvách nižších.

Na základě těchto poznatků byl navržen model OSI (Open System Interconnection Reference Model), který zachycuje celou problematiku komunikace. Stal se tak výchozím modelem komunikační architektury pro počítačově řízenou výměnu dat. Skládá se ze 7 vrstev (Obrázek 4).

Obrázek 4: Model OSI

Čtyři horní vrstvy se týkají zejména příslušné aplikace a může v nich docházet k zásahům do struktury i obsahu přenášených dat – konkrétní řešení je závislé na programovém řešení ve spojitosti s typem dané aplikace.

Spodní tři vrstvy se týkají především vlastního přenosu datových zpráv od jednoho účastníka k druhému, aniž by se zde docházelo ke změnám jejich obsahu i formy.

1.1.1 TCP

Protokol TCP (Transport Control Protocol) pracuje spolu protokolem UDP na transportní vrstvě. Protokol TCP je spojovanou službou, tj. službu která mezi dvěma aplikacemi naváže spojení – vytvoří na dobu spojení virtuální okruh. Tento okruh je plně duplexní (data se přenášejí současně oběma směry). Přenášené pakety jsou číslovány. Ztracená nebo poškozená data jsou znovu vyžádána. Integrita přenášených dat je zabezpečena kontrolním součtem. Konce spojení, odesílatel a adresát, jsou určeny tzv. číslem portu. Toto číslo je 16 bitové, takže může nabývat hodnot 0 až 65535. Protokol je popsán v RFC 962.

1.1.2 UDP

Protokol UDP (User Datagram Protocol) je určen pro aplikační protokoly a procesy, které požadují rychlý a nezabezpečený přenos paketů bez potvrzování příjmu a opakování přenosu. UDP je popsán v RFC 768.

V sadě protokolů Internetu poskytuje UDP velmi jednoduché rozhraní mezi síťovou vrstvou a aplikační vrstvou. UDP neposkytuje žádné záruky doručení a odesilatelova UDP vrstva si u jednou už odeslaných zpráv neudržuje žádný stav. UDP pouze přidává kontrolní součty a schopnost roztřiďovat UDP pakety mezi více aplikací běžících na stejném počítači (pomocí portů).

Zvláštností protokolu UDP je skutečnost, že adresátem UDP datagramu nemusí být pouze jednoznačná IP-adresa, tj. síťové rozhraní konkrétního počítače. Adresátem může být skupina stanic.Adresovat lze všeobecné oběžníky (broadcast), které jsou určeny pro všechny uzly v jedné síti. Podstatně zajímavějším případem je adresování adresných oběžníků (multicast), které jsou určeny jen vybraným uzlům a při kterých dochází k ohromné úspoře kapacity přenosových cest narozdíl spojení s jedním uzlem (unicast).

1.1.3 RTP

Protokol RTP podporuje přenos médií (audio a video) v reálném čase nad protokolem IP. Je využíván standardy SIP a H.323. Protokol zjistí příjemci každý ztracený paket. Hlavička RTP obsahuje informace, které pomáhají příjemci obnovit média (hlas). Protokol RTP doplňuje protokol UDP.

RTP zahrnuje:

- Pořadí

  Pořadové čísla v paketu RTP jsou používána pro detekci ztracených paketů

- Identifikace zatížení

  K nastavení nové šířky pásma potřebujeme měnit styl kódování medií. K uskutečnění této funkce je potřeba každého RTP paketu k určení charakteristiky kódovacího média.

- Rámec

  Hlas je posílán v logických jednotkách, které jsou nazávány rámce. K signalizaci začátku a konce rámce se používá ukazovatel rámce.

- Zdrojová identifikace

Identifikátor se vyžaduje k určení odesílatele rámce.

1.1.4 RTCP

RTCP je kontrolní protokol. V relaci RTP účastník pravidelně posílá RTCP pakety k získání informací o kvalitách služeb. Další služby, které RTCP provádí směrem k účastníkovi:

- Odezva QoS

RTCP se používá k podávání zpráv o kvalitách služeb. Informace obsahují počet ztracených paketů, kolísání a tyto informace jsou používány k nastavení rychlosti přenášených dat.

- Řízení relace

RTCP informuje účastníka o tom, že pakety opouští relaci.

- Identifikace

Informace typu emailové adresy, jména, telefonní čísla jsou zahrnuty v hlavičce paketu RTCP. Všichni uživatelé vědí identifikační údaje o ostatních zúčastněných v dané relaci.

- Synchronizace

Přestože obraz a zvuk jsou posílány přes různé toky, potřebujeme je synchronizovat dohromady u příjemce. Protokol RTCP obsahuje informace, které toky jsou požadovány pro synchronizaci.

1.2 Standardy VoIP

Jedna z nejdůležitějších věcí, která se musí řešit při nasazování IP telefonie, ať už z operátorského nebo uživatelského pohledu, je správný výběr signalizačního protokolu. Na tomto výběru pak závisí, s kým a za jakých podmínek se bude moci uživatel propojit. Na druhou stranu je třeba říci, že většina systémů VoIP v dnešní době podporuje více signalizačních protokolů (obvykle SIP a H.323).

1.2.1 SIP

Protokol SIP (Session Initiation Protocol) je textový signalizační protokol zajišťující budování, dohlížení a rušení telefonních hovorů na bázi VoIP. Tento protokol tedy zajišťuje pouze telefonní signalizaci a komunikaci mezi komponentami VoIP. Vlastní přenášení multimediálních dat zajišťují ostatní protokoly, například textový protokol SDP slouží pro popis médií a protokol RTP, který je užíván k přenosu multimédií v reálném čase. Díky své textové povaze (obdoba HTTP protokolu) je protokol SIP lehce pochopitelný a pružný.

Tento protokol vznikl na půdě organizace IETF (Internet Engineering Task Force), která má na starosti doporučení ohledně protokolů používaných v Internetu. Pracovala na něm skupina MMUSIC (Multiparty Multimedia Session Control), která v roce 1999 vytvořila RFC 2543. V témže roce se skupina přejmenovala na příznačný název SIP a vydala RFC 3261 [2].

SIP byl navržen v souladu s modelem Internetu. Jde tedy o end-to-end orientovaný signalizační protokol. Princip je založen na tom, že pokud je to možné, měly by být operace komunikačního protokolu definovány tak, aby byly prováděny buď v koncových bodech komunikačního systému, nebo co nejblíže zařízení, které je ovládáno. Veškerá logika je tedy uložená v koncových zařízeních (vyjma směrování SIP zpráv), koncové zařízení zná i jednotlivé stavy komunikace, tím je zvýšena odolnost komunikace proti chybám. Nevýhodou je pak ale vyšší režie v hlavičkách zpráv.

U klasické PSTN jsou všechny stavy a mechanizmy uloženy v síti a jednotlivé koncové body (telefony) jsou velmi jednoduché. Cílem SIPu je zajistit stejné funkce jakou mají klasické PSTN, ale end-to-end návrh umožní SIP sítím vyšší výkonnost a umožní implementaci nových služeb, které mohou být jen stěží nasazeny v klasických telefonních sítích [3].

Používá oddělený signalizační a datový tok – pro signalizaci používá TCP nebo UDP protokol (standardně port 5060) a pro RTP používá UDP protokol. Tato vlastnost má problémy s Network Address Translation (NAT). Pokud jsou totiž obě strany spojení za NAT, pak nemůže žádná z obou stran adresovat tu druhou. Tento problém řeší STUN server (Simple Traversal of UDP over NATs). STUN server je kontaktován klientem v případě, že se klient nachází v neveřejném adresním prostoru, a tedy komunikuje prostřednictvím překladu adres (NAT). STUN server vrátí klientovi IP adresu, kterou má použít při inicializaci volání.

SIP přesně nespecifikuje používané protokoly k přenosu dat. Proto může protokol RTP fungovat jak se spojově orientovaný přenos (TCP) či nespojově

orientovaný s UDP. Spojení s UDP poskytuje větší rychlost, díky nespojovanému a nepotvrzovanému přenosu dat, a snadnější implementace, jenž je způsobena jednodušší architektuře UDP. Stejně tak není SIP pevně svázán s žádnými konkrétními protokoly pro přenos multimediálních dat. Uvnitř zprávy protokolu SIP při navázání spojení je proto zapouzdřena zpráva jiného protokolu, který specifikuje použitá kódování pro multimediální data, jejich parametry a čísla portů, na kterých mají být data vysílána nebo přijímána. Obvykle se pro tento účel používá SDP (Session Description Protocol), který je rovněž textový. SDP je popsán v RFC 2327.

SIP je protokol typu klient-server. Klient navazuje spojení se serverem. Jedno zařízení může pracovat současně jako klient i jako server. Rozlišujeme 2 druhy SIP komponentů: uživatele (UA, User Agent) a servery. UA je chápán jako koncový terminál schopný zpracovávat multimediální data a SIP signalizaci. Server je pro multimediální data pouze prostředníkem přenosové cesty. Pomáhá vytvořit komunikační cestu mezi UA. Nad daným spojením pak může dohlížet (například pro tarifikaci, záznamy hovorů, apod.). UA se rozděluje na dvě části:

- UAC (UA Client) vysílá žádosti a přijímá odpovědi,
- UAS (UA Server) přijímá žádosti a odesílající odpovědi.

Servery jsou v SIP architektuře zařízení, jejichž úkolem je zprostředkovat kontakt mezi volajícími a volanými (tedy mezi UA), což ale nevylučuje přímý kontakt koncových zařízení bez účasti serverů.

- **Redirect server**

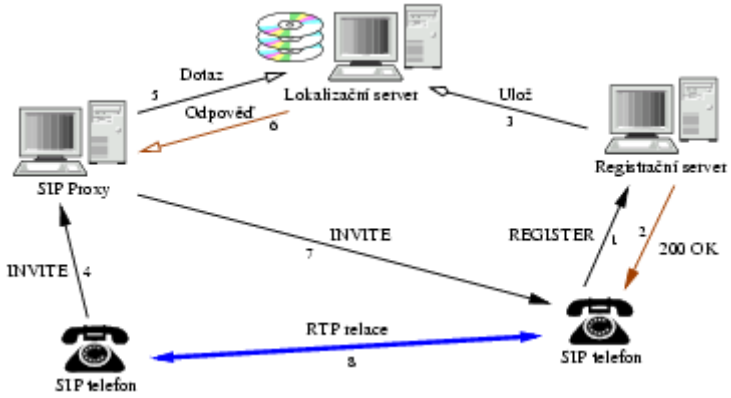
  Pokud klient nezná adresu serveru, na kterém se nachází volaný účastník, kontaktuje redirect server, který pomocí lokalizační sužby zjistí adresu volaného a předá ji zpět volajícímu klientovi. Negeneruje žádosti o spojení s volaným klientem, tato úloha je již na volajícím. Pokud je k dispozici více adres, může kontaktovat více serverů.

- **Proxy server**

  Chová se podobně jako redirect server, ale po získání adresy volaného klienta, sám naváže spojení a potvrdí spojení volajícímu klientovi. Obsahuje tedy funkce klienta a serveru, dokáže interpretovat a přepisovat hlavičky žádostí. Maskuje tedy pravého uživatele, což se dá využít k zabezpečení firemní sítě a podobně.

- **Registrar server**

  Registrar server přijímá od uživatelů požadavky na registraci, tím získává informaci o jejich aktuální poloze (IP adresa, port a uživatelské jméno) a ukládá informaci do lokalizační databáze (location server). Registrace se musí obnovovat. Obrázek 5 ukazuje příklad komunikace mezi servery.



Obrázek 5: Relace mezi servery [4]

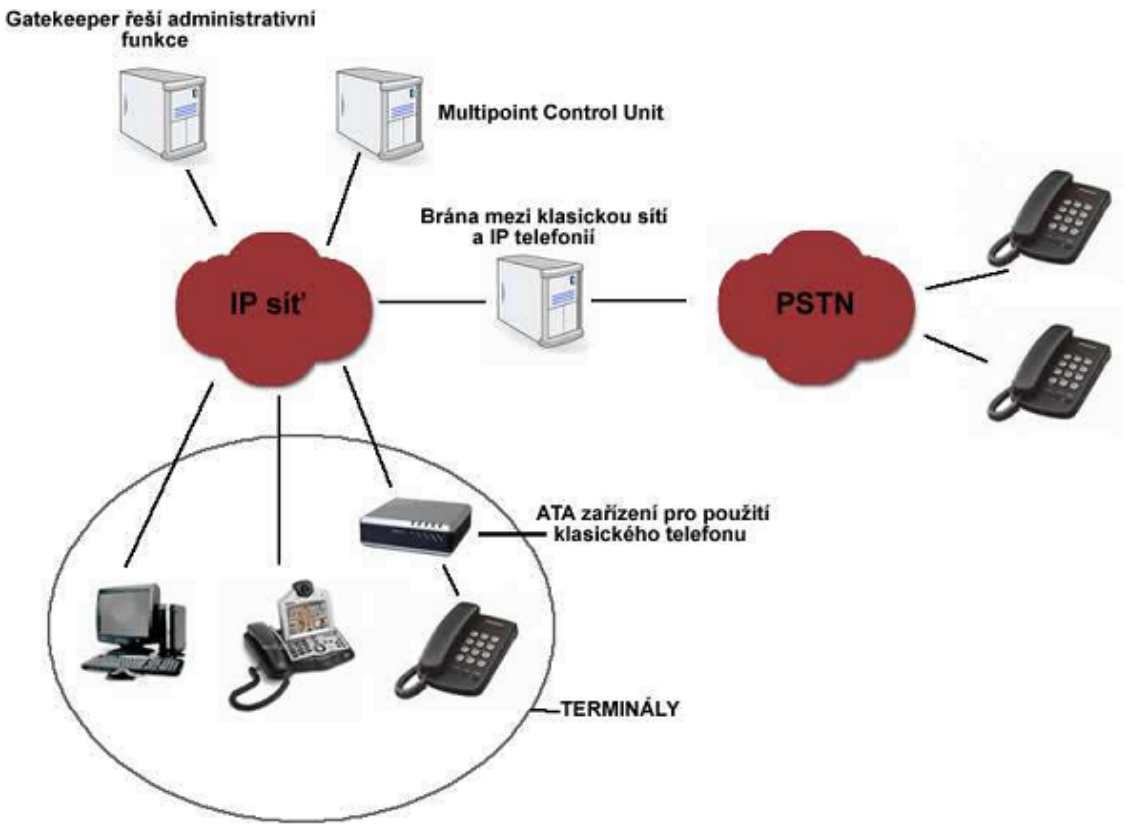
- **Location server**

  Jeho databáze obsahuje informace o umístění klientů a případně i dalších proxy serverů. Komunikace mezi lokalizačním serverem a ostatními servery, které využívají jeho databázi, probíhá skrze protokol LDAP (Lightweight Directory Access Protocol), který slouží pro přístup k adresářovým službám v komunikacích online.

Jakkoliv jsou proxy, redirect a registrar servery definovány odděleně, v praxi se často jedná o jednu aplikaci, která se pak nazývá jako SIP server .

1.2.2 H.323

H.323 je zastřešující standard definující multimediální komunikační infrastrukturu nad paketově orientovanými sítěmi. Jako takový se opírá o řadu dalších protokolů a standardů, které popisují vlastnosti povinných a volitelných součástí této infrastruktury a jejich vzájemnou interakci. Standard H.323 je schválený International Telecommunications Union – Telecommunication Standardization Sector (ITU-T). Obrázek 6 ukazuje strukturu H.323 sítě.



Obrázek 6: Struktura H.323 sítě

První generace H.323 byla spíše orientována na videokonferenci v počítačové síti. Druhá generace (roku 1998) už reflektovala požadavky na telefonii a zlepšila rychlost navazování spojení, podporu zabezpečení, kvalitu služeb a přesměrování hovorů. V další generaci (roku 1999) přibyly hlavně doplňkové služby, použití protokolu UDP místo TCP při navazování spojení v zatížených sítí apod. Generace čtvrtá (roku 2000) zajistila větší spolehlivost a snazší rozšiřitelnost. Další generace (pátá v roce 2003 a šestá v roce 2006) přinesly už jen drobné změny.

Komponenty v H.323 [5]:

Terminál H.323

Terminál je koncový bod v H.323, který umožňuje obousměrnou komunikaci v reálném čase. Komunikovat lze s jiným terminálem, bránou nebo Multipoint Control Unit (MCU). Terminály musí dle standardu povinně implementovat pouze přenos hlasu, volitelně mohou přenášet obraz i jiná uživatelská data. Terminál může být jak IP telefon tak i klasický telefon připojený přes bránu.

Brána (Gateway)

Brána je nepovinným elementem H.323 sítě, poskytující vhodný překlad mezi přenosovými formáty a též mezi komunikačními formáty. Dalšími funkcemi brány může být například sestavení a ukončení hovoru mezi dvěma terminály nebo překlad mezi různými audio, video a datovými formáty. Snaží se tedy o spojení mezi sítěmi postavenými na jiných protokolech.

Gatekeeper

Gatekeeper je v H.323 síti nepovinným prvkem, který ostatním koncovým zařízením, jako jsou terminál, brána nebo MCU, poskytuje službu řízení hovorů. Ačkoliv je gatekeeper logicky oddělen od koncových zařízení, muže být fyzickou součástí kteréhokoliv z nich. Seskupení H.323 koncových zařízení, jejichž vzájemnou komunikaci řídí gatekeeper, se nazývá zóna. V každé zóně se nachází právě jeden gatekeeper (je možné použít i záložní gatekeepery). Existence gatekeeperu je nepovinná. Terminály mohou komunikovat i přímo mezi sebou, pokud je ale gatekeeper přítomen, jsou povinné komunikovat přes něj.

Gatekeeper musí poskytovat tyto služby:

- Překlad adres (Address Translation)
  Gatekeeper překládá zástupné identifikátory koncových zařízení na jejich transportní adresy.

- Řízení přístupů (AdmissionsControl)
  Gatekeeper úvěruje oprávněnost přístupu do H.323 sítě použitím řídicích zpráv.

- Kontrola šířky pásma (Bandwidth control)
  Gatekeeper muže na základě zjištění nedostatečné šířky pásma odmítnout hovor.

- Správa zóny (Zone Management)
  Spravuje tedy svoji zónu tvořenou zaregistrovanými zařízeními a poskytuje jim dané služby.

Za účelem podpory vícebodových konferencí může gatekeeper sloužit jako směrovač H.245 řídících kanálů pro dva terminály. Ve chvíli, kdy se konference rozšíří o další terminály, může gatekeeper přenechat správu H.245 řídících kanálů některé jednotce Multipoint Controler (MC) v síti.

Multipoint Processor Unit (MCU)

MCU je v H.323 infrastruktuře koncovým bodem, který terminálům v H.323 síti poskytuje podporu pro vícebodové konference. MCU je složeno z žádného, jednoho nebo několika MP a z jednoho MC.

- Multipoint Controler (MC) zastupuje úlohu řídícího prvku v konferencích mezi třemi a více koncovými prvky. MC se stará o výměnu podporovaných vlastností, jako je například množina podporovaných komprimačních algoritmu, mezi všemi účastníky konference. Od každého účastníka získá množinu jím podporovaných vlastností. Nasbírané informace vyhodnotí a následně zašle všem klientům množinu vlastností, jimiž by se měli řídit při vysílání. V závislosti na připojování a odpojování klientů ke konferenci může MC poopravit množinu vlastností, kterou zasílá terminálům.

- Multipoint Procesor (MP) přijímá audio, video a nebo datové toky od všech účastníků konference. Přijatá data zpracuje a rozesílá zpět účastníkům. MP je volitelná komponenta MCU a může být umístěna v sítí i mimo MCU.

Základní protokoly zastřešené doporučením H.323:

H.225.0, RAS – hovorová signalizace
Q.931 – hovorová signalizace
H.245 – vyjednání parametrů audia/videa
H.235 – bezpečnostní a ověřovací mechanismy
RTP – Real Time Protocol
H.450.x – doplňkové služby

1.2.3 IAX

IAX (Inter-Asterisk eXchange) je komunikační binární protokol navržený pro komunikaci mezi open-source ústřednami Asterisk firmy Digium. Používá se pro komunikaci mezi servery ale i v komunikaci klient-server. Při vzniku byl kladen důraz na minimalizaci šířky pásma pro signalizaci a vlastní přenášení hlasu. Dokáže zprostředkovat přenos v jednom UDP (port 4569) kanálu jak signalizace, tak užitečných dat. Signalizace je přenášena dohromady s hlasem, což činí IAX přístupnějším pro firewally a mnohem lépe pracuje za NATem. Výhoda jediného kanálu pro signalizaci i data je i jednodušší detekce chyby. Spojení buď funguje, nebo nefunguje. U jiných protokolů se může stát, že přijde signalizace zvednutí sluchátka, ale nemusíme nic slyšet – problém při navázání RTP spojení. Podporuje multiplexování kanálů do

jednoho kanálu. Data více hovorů, resp.od více uživatelů jsou sloučena do jednotného proudu paketů. Jediný IP datagram může nést informace o více hovorech, což minimalizuje režii (velikost IP hlaviček) a zefektivňuje celkový provoz. Díky multiplexování bude vždy signalizace i hlasové pakety procházet přes VoIP operátora, což není zrovna žádoucí, volá-li se v rámci vlastní sítě operátora. Pro signalizaci zde slouží 32 bitů, což je oproti textové SIP signalizaci výhoda. Jednoúčelová zařízení používající tento protokol mohou být řádově jednodušší, ale textová SIP signalizace se administrátorovi zase jednodušeji sleduje.

IAX je extrémně pružný a může být využit se všemi typy multimediálního datového toku včetně videa, primárně je však určen pro přenos hlasu. Nedostatkem je malá podpora výrobců.

1.2.4 MGCP

Protokol MGCP (RFC 2705) znamená Media Gateway Control Protocol. Funkce tohoto protokolu spočívá v tom, že umožní řízení telefonních bran MG (Media Gateway) z externích objektů. Externím objektem může být MGC (Media Gateway Controller) nebo CA (Call Agent). MGC nebo CA umožňuje řízení a kontrolu signálů, posíláním/přijímáním příkazů do/z brány. Brána má na starosti konverzi mezi paketovou a přepínavou sítí a také informování CA o událostech na straně klienta.

Protokol MGCP umí spolupracovat s protokoly SIP a H.323 a je založený na centrálním řízení s větší inteligencí, které řídí menší zařízení s omezenou inteligencí. Toto je také jeden z hlavních rozdílů proti protokolům H.323 a SIP. Protokol se snaží oddělit část, která má na starosti sestavování a řízení hovorů od funkcí, které se zabývají spojením na nižší úrovni. V síti s protokoly H.323, SIP a MGCP potom spojení funguje tak, že protokol MGCP má na starosti řízení bran a protokoly SIP a H.323 řídí hovor.

1.2.5 MEGACO (H.248)

Další standardní protokol, je produktem společného úsilí ITU a IETF (Megaco název pro IETF a H.248 pro ITU-T). Muže být použit u bran (gateway) a komunikace server-server. Hardwarové ani softwarové telefony ho nepoužívají.Vychází z MGCP, na rozdíl od něj ale podporuje rozsáhlejší sítě.

1.2.6 SCCP

Protokol SCCP (Skinny Client Control Protocol), někdy též označován zkráceně jen Skinny, je proprietální standard firmy Cisco pro VoIP. Skinny byl původně vyvinut firmou Selsius Corporation. Už samotný název značí (Skinny = vyzáblý), že tento protokol je navrhnut pro co největší nekomplikovanost a pro jednoduché implementování do zařízení. Tento protokol slouží ke komunikaci mezi IP telefony firmy Cisco a Cisco Call Managerem (CCM) [2]. CCM je softwarová komponenta, která běží na serverech schválených firmou Cisto. Program CCM rozhoduje o předávání hovorů, řídí IP telefony a může podporovat konferenční hovory a další volitelné funkce. K samotnému přenosu hlasu používá protokol RTP.

1.3 Zvukové kodeky

Hlas je ve své přirozené podobě analogový, tedy představuje plynule se měnící vlnovou křivku. V sítích VoIP se hlas naopak přenáší v digitální podobě pomocí kódování. Kodek (COde, DECode) je matematický algoritmus, který zajišťuje kódování a dekódování zvukového signálu do digitální podoby. Je jich několik druhů (Tabulka 1) a jejich hlavními parametry jsou:

- Šířka pásma nutná pro přenos zvuku
- Kvalita přenášeného zvuku (MOS)
- Zátěž na procesor (převážně při kódování)
- Zda-li je kodek volně k použití nebo si musíme koupit licenci

Tabulka 1: Zvukové kodeky

Kodek	Algoritmus	Šířka pásma [kbit/s]	Relativní složitost	MOS
G.711	PCM	64	nízká	4,1
G.723.1	ACELP	5,3/6,3	střední	3,9/3,6
G.726	ADPCM	32	nízká	3,85
G.728	LD-CELP	16	nízká až střední	3,61
G.729	CS-ACELP	8	střední až vysoká	3,92

U kodeků, které používají patentované algoritmy, není zaručeno, že všechny komponenty sítě je právě kvůli licenci budou obsahovat. Některé kodeky se snaží data komprimovat, aby se snížila šířka pásma nutná pro přenos zvuku – zde je nevýhoda větší zátěže CPU počítače. Pokud brána spojení fungující na rozdílných kodecích, musí

dvakrát dekódovat a kódovat data, což při větším počtu hovorů přináší netriviální zátěž procesoru. Kodeky jako G.729 se například na zvuk snaží napasovat vzory, které odpovídají lidskému hlasu, některé další kodeky se snaží posílat diference mezi jednotlivými časovými úseky, což ale při přenosu přes nekvalitní médium může přinášet velice nekvalitní přenos hlasu. V tabulce je vidět, že šířka pásma u určitých kodeků je řádově jiná.

Pulsně-kódová modulace PCM neprovádí žádnou komprimaci, ale jen vzorkuje a kvantuje hlas. Je to základní modulace používaná v G.711.

Adaptivní diferenční pulsně-kódová modulace ADPCM používá rozdílový signál. Používají se jen rozdíly aktuálního vzorku oproti předchozímu vzorku. Příkladem ADPCM je G.726.

CS-ACELP (Conjugate Structure Algebric Code Excited Linear Predication) používá tzv. knihu kódů vytvořenou na základě vzorků řeči. Poté již jen zkoumá, zda-li se vzorek hlasu nenachází v knize kódů. Pokud odpovída, lze poté zaslat místo skutečného vzorku pouze umístěná v knize kódů. Příkladem CS-ACELP je G.729. Tato komprese je určena ke kódování řeči, proto jiné zdroje zvuku jako například hudba nebudou vykazovat stejnou kvalitu jako řeč.

Pro měření kvality hlasu (kodeku) se používá MOS (Mean opinion score). Jedná se o subjektivní charakteristiku. Hodnocení MOS je prováděno náhodně vybranou skupinou lidí, která známkuje kvalitu přeneseného hlasu v porovnání s klasickým analogovým spojením. Hodnoty mají rozsah od 1 (nedostatečná kvalita) až po 5 (neznatelné zhoršení kvality). Pro telefonní kvalitu hlasu je vhodná hodnota MOS okolo 4. Kodek G.711 má hodnotu MOS 4,1. Kodek G.729 má hodnotu 3,92, ale přináší podstatnou úsporu šířky pásma.

Další možnosti jsou PSQM (Perceptual Sérech Quality Measurement), u kterého se digitálně měří rozdíl mezi původním signálem a signálem po průchodu kodekem. A PESQ (Perceptual Evaluation of Speech Quality) měří kvalitu taktéž digitálně, ale snaží se srovnávat známější hodnoty MOS [2].

2 Kvalita služeb

Integrace hlasových a datových sítí do jediné sítě přináší řadu problémů. Dříve bývaly přenosy hlasu a dat fyzicky oddělené. Hlasové služby POTS (Plain Old Telephone Service) používaly veřejnou komutovanou telefonní síť PSTN. Při zavedení VoIP se využívají stávající datové sítě, čímž dochází k významným úsporám v nákladech na vybavení, údržbu a obsluhující personál. Jednotlivé aplikace ale zápasí o dostupnou šířku pásma. Hlasové aplikace jsou ale více náchylné na kvalitu přenosu než datové aplikace (http, FTP, …). Při neomezené šířce přenosového pásma by tyto problémy nevznikaly. Ale dostupná šířka pásma mezi koncovými body neodpovídá většinou po celé délce požadavkům na přenos hlasových aplikací. Proto je nutnost použít nástroje, které zajistí kvalitu služeb QoS (Quality of Service).

QoS je tedy možné chápat jako schopnost sítě poskytnout dané aplikaci bez omezení její funkce či výkonu dostačující kvalitu přenosu. Sada nástrojů QoS ovlivňuje tyto vlastnosti:

- Zpoždění (latency, delay) – Doba, která je potřebná k doručení paketu od zdroje k cíly.

- Časová nestabilita (jitter) – Doba doručení paketů je proměnná, tj. rozestupy mezi příchody paketů jsou nerovnoměrné.

- Zahazování paketů (packet loss) – Paket nedorazí k cíly z důvodu vzniku kritické situace.

Aplikace, které mají přísné nároky na QoS, se nazývají kritické. Příkladem takových aplikací je právě real-time přenos obrazu a zvuku (tedy VoIP). Podle společnosti Cisco nesmí pro VoIP překročit zpoždění v jednom směru 150ms, časová nestabilita 30ms a ztrátovost paketů nesmí být vyšší jak 1% [2].

Zpoždění

Latence představuje důležitou metriku, která má značný vliv na kvalitu VoIP. Latence v paketově orientovaných sítích vyjadřuje dobu, za kterou se vyslaný paket dostane k příjemci.VoIP komunikace musí být prováděna v reálném čase a velké zpožděni může vest k tomu, že řeč nebude rozpoznatelná.

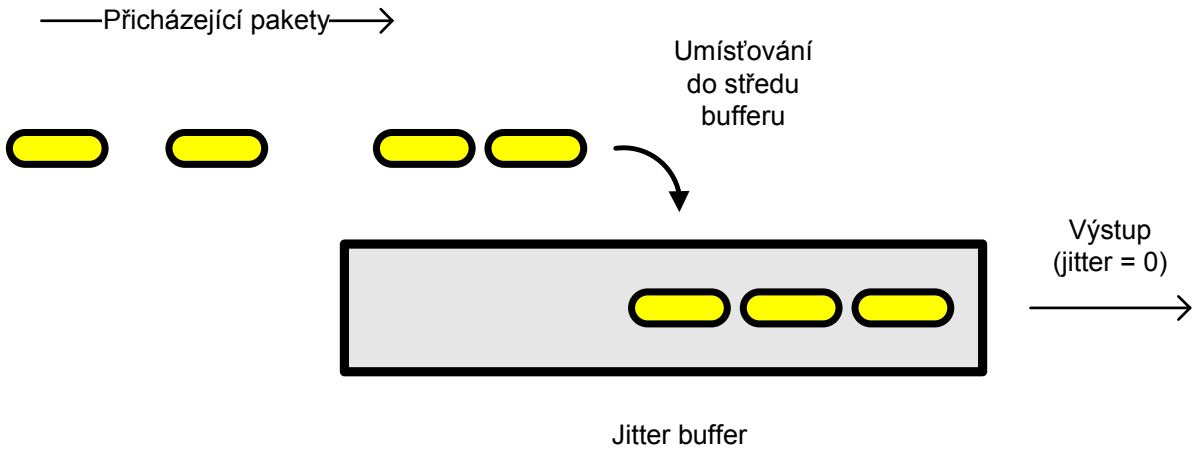
Z hlediska velikosti zpoždění jsou výhodné malé velikosti fragmentů. To ale způsobuje větší zatížení sítě, protože se zmenší rozdíl mezi velikostí hlavičky a daty, které paket obsahuje. Tabulka 2 zobrazuje seznam druhů zpoždění [2].

Tabulka 2: Druhy zpoždění

Druhy zpoždění	**Popis**
Jitter Buffer	Závisí na velikosti jitter bufferu, který zadržuje pakety, aby potlačil jitter.
Kodek	Podle typu použitého kodeku dochází při zpracování hlasu k jeho zpoždění.
Store and forward	V zařízeních, které přeposílávají pakety, musí dojít nejdříve k jejich celému uložení než dojde k odeslání.
Propagace	Čas potřebný k cestě dat z jednoho konce sítě na druhý. Je to způsobeno konečnou rychlostí šíření signálu po přenosovém médiu.
Serializace	Doba potřebná k vložení paketu na přenosové médium.
Paketování	Vzniká při umísťování části hlasu do paketu.
Přenos	Součet store and forward, serializačního a propagačního zpoždění.

Jitter

Nejdůležitější z QoS pro VoIP je jitter. Jitter je variace v čase příchodu paketu porovnávaném s očekávaným časem příchodu, jak ho změřil přijímač. Na eliminaci jitter se používá jitter buffer [2].



Obrázek 7: Jitter buffer

První paket je umístěn do středu bufferu. Jak jsou pakety přijímány, jsou umístěny za předcházející paket relativně k tomu, jak budou přehrávány. Velikost bufferu se měří časem, který potřebuje paket k tomu, aby se ze středu posunul až k

přehrání, a je obvykle nastavena v softwaru. Čím delší je buffer, tím více odchylek zvládne. Čím je buffer kratší, tím spíš se zaplní.

Z tohoto fungování vyplývají dvě důležité skutečnosti. Zaprvé: pokud se buffer přeplní předčasným příchodem paketů, budou další pakety zahozeny. Podobně, pokud se pakety zpozdí a buffer se vyprázdní, bude muset přijímač simulovat nějaký zvuk nebo počítat s úsekem ticha. Zadruhé: buffer přináší zpoždění odpovídající času, které potřebuje paket pro cestu z centra bufferu až k přehrání. Proto platí, čím delší buffer, tím delší zpoždění známé jako jitter buffer delay.

Ztrátovost paketů

Ztrátovost paketů značí, kolik procent paketů nedorazí od odesilatele k cíly. Ztráty mohou vzniknout průchodem sítí nebo při použití jiter bufferu. Ztráta na síti může být způsobena poškozením paketu, který, pokud to detekuje obsluhující zařízení, je zahozen. Při přesunu větších paketů je větší šance, že se jeho malá část poškodí a bude zahozena. VoIP pakety bývají malé v porovnání s datovými pakety (FTP, ..). Častější příčina zahození paketu je zahlcení sítě. Pokud je směrovač zahlcen velkým množstvím paketů, které není schopen zpracovat, podle různých mechanizmů dochází k jejich zahazováním. Protokol RTP označuje hlasové pakety pořadovým číslem, díky kterému cílové zařízení pozná nedoručení paketu. Tuto chybu se snaží napravit přehráním staršího paketu nebo vyplnění daného úseku tichem.

3 Průběh SIP

Zprávy protokolu SIP jsou dvojího druhu – Request a Response (žádosti a odpovědi). SIP využívá k signalizaci textových zpráv. V nichž jsou obsaženy žádosti, někdy se jim říká metody.

3.1 Žádosti

Mezi hlavní žádosti patří:

INVITE - žádost o navázání spojení nebo o změnu parametrů již existujícího spojení.
BYE - žádost o ukončení spojení
ACK - žádost, kterou klient potvrzuje, že obdržel odpověď na žádost INVITE
REGISTER - žádost o registraci klienta u registračního serveru
CANCEL - žádost o zrušení probíhající žádosti INVITE
OPTIONS - žádost o zaslání přehledu funkcí podporovaných serverem
INFO - znamená přenos informací během hovoru
UPDATE - dovoluje klientovi aktualizovat parametry spojení
PRACK - potvrzení prozatímní (1xx) odpovědi
SUBSCRIBE - přihlášení k odběru informací
NOTIFY - doručení informace
MESSAGE - zpráva
REFER - požadavek jiného UA k relaci (např. inicializace spojení přes web)
PUBLISH - aktualizace stavu informace na server

3.2 Odpovědi

Odpovědi SIP jsou rozděleny do šesti skupin. Obsahují návratový kód, který ve formě třímístného číslo označuje výsledek žádosti obdobně, jak je tomu například v protokolu HTTP. Podrobnější přehled viz Příloha A. Lze je rozdělit do šesti skupin podle první číslice:

1*xx* - žádost přijata, pokračuji ve zpracování žádosti,
2*xx* - znamená úspěšné provedení žádosti,
3*xx* - označuje přesměrování (odpověď od redirect serveru),
4*xx* - chyba způsobená klientem (chybný formát žádosti),
5*xx* - chyba způsobená serverem,
6*xx* - obecná chyba, žádost nemůže být akceptována ani jiným serverem.

3.3 Identifikace

Základní adresa uživatele má tvar e-mailové adresy, nazývá se URI (Uniform Resource Identifier). Existují dvě podoby, SIP URI a SIPS URI. Forma zápisu je stejná, ovšem druhá verze (uvozená "sips:") je určena pro zabezpečenou formu komunikace. Běžně používaný pojem URL (Uniform Resource Locator) je specifickou podtřídou URI.

Příkladem SIP URI je zápis ve tvaru:

sip:novak@vutbr.cz

Toto je adresa, kterou normálně použije volající, když chce navázat spojení s druhou stranou. Adresa však může obsahovat i další údaje popisující způsob kontaktu uživatele. Úplný tvar adresy je následující:

sip:[uživatel[:heslo]@]hostname[:port][;parametry][?hlavičky]

Hranaté závorky označují nepovinné části. Jediným povinným údajem je ve skutečnosti jméno počítače/telefonu, místo něhož může být uvedena přímo IP-adresa zařízení. Před jméno zařízení lze volitelně uvést jméno uživatele případně heslo na tomto zařízení. Heslo se ale přenáší ve formě obyčejného textu a tudíž představuje bezpečnostní riziko. Pracuje-li zařízení jako brána do klasické telefonní sítě, ve které se uživatelé označují telefonními čísly, uvádí se místo jména uživatele jeho telefonní číslo. Hostname označuje vzdálený server poskytující SIP služby. Položka „port“ specifikuje číslo portu, na který se má požadavek zaslat. Pokud není zadáno, implicitně se předpokládá port 5060 pro přenos skrze protokoly TCP a UDP, nebo port 5061 pro SIPS přenášený přes TCP (SIPS musí být přenášen spolehlivou službou) a TLS (Transport Layer Security). Za jménem zařízení může následovat seznam parametrů a hlaviček. Parametry určují, jaký protokol má být použit v transportní vrstvě (UDP nebo TCP), zda je uvedeno jméno nebo telefonní číslo uživatele a další údaje. Hlavičky se potom stávají přímo součástí signalizační zprávy.

3.4 Příklad spojení

Komunikace užívající SIP signalizaci je tvořena zprávami, které jsou obvykle přenášeny v samostatných UDP datagramech. Každá zpráva obsahuje hlavičku zprávy (header) a vlastní obsah (body). V prvním řádku zprávy je identifikován její typ.

Známe dva typy zpráv :

- žádost (metoda),
- odpověď.

Žádosti jsou obvykle užívány k inicializaci procedury (sestavení, ukončení spojení) nebo oznamují příjemci určitý požadavek. Odpovědi jsou užívány k potvrzení, že žádost byla přijata a zpracována a obsahuje stav zpracování. Typická SIP žádost může vypadat následovně [8]:

```
INVITE sip:7170@vutbr.cz SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5060
Max-Forwards: 10
;From: „novak“ <sip:novak@vutbr.cz>tag=76ff7a07-c091-4192-84a0-
d56e91fe104f
To: <sip:novak@bat.vutbr.cz>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username=„novak“, realm=„vutbr.cz“,
algorithm=„MD5“, uri=„sip:novak@bat.vutbr.cz“,
nonce=„3cef7539000000001771328f5ae1b8b7f0d742da1feb5753c“,
response=„53fe98db10e1074
b03b3e06438bda70f“
Content-Type: application/sdp
Content-Length: 451

v=0
o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DVI4/16000
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap: 3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

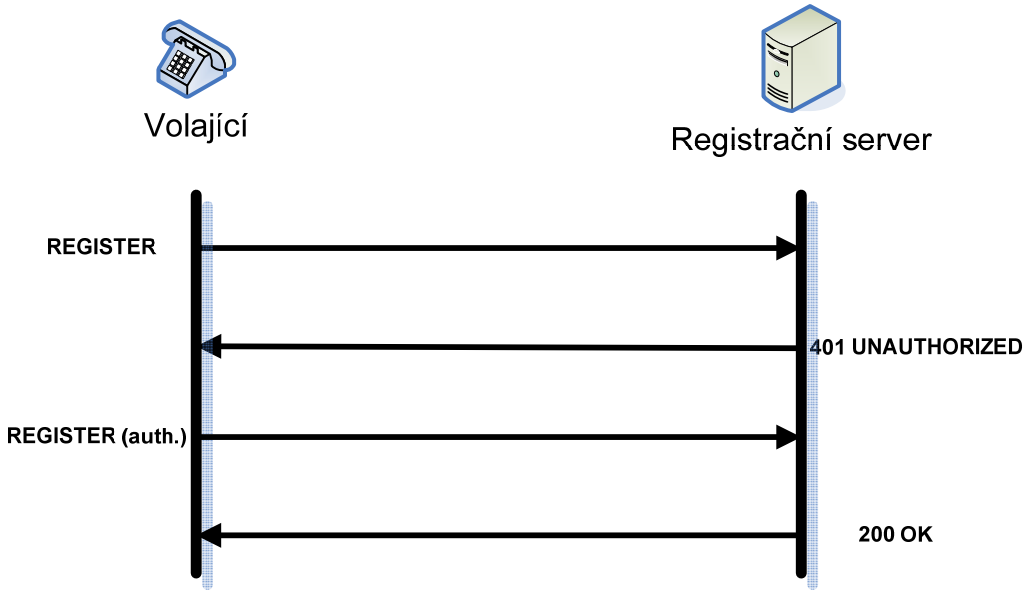
První řádek nám říká, že se jedná o zprávu INVITE, jež je užívána k sestavení spojení. URI na první řádce —sip:7170@vutbr.cz se nazývá Request URI a obsahuje URI dalšího skoku zprávy (next hop). V tomto případě bude hostitelem „vutbr.cz“. Request URI je tedy aktuální adresát požadavku.

SIP žádost obsahuje v hlavičce jedno nebo více polí Via, jež jsou užívány k záznamu cesty žádosti. Následně jsou užívány ke směrování SIP odpovědí přesně takovou cestou, jakou byly odeslány. Naše INVITE zpráva obsahuje jedno pole Via, které bylo vytvořeno UA, který odeslal žádost. Z pole Via můžeme říct, že UA používá IP adresu 195.37.77.100 a port 5060. Pole hlavičky From a To identifikuje iniciátora volání (volající) a příjemce (volaného). Pole From obsahuje parametr tag, který slouží jako identifikátor dialogu daného uživatele. Pole hlavičky Call-ID je identifikátor dialogu a jeho cílem je identifikovat zprávy náležející jednomu volání. Takovéto zprávy mají stejný identifikátor Call-ID. Ke správě pořadí požadavků je užíváno pole CSeq. Protože žádosti mohou být odeslány nespolehlivým přenosem, který může způsobit zpřeházení zpráv, pořadové číslo se musí ve zprávě nacházet, aby příjemce mohl rozpoznat opakování přenosu a selektovat žádosti. V hlavičce je pole Contact obsahující IP adresu a port, na kterém odesílatel očekává další žádosti odesílané volaným [8].

Hlavička zprávy je oddělena od těla zprávy prázdným řádkem. Tělo zprávy žádosti INVITE obsahuje protokol SDP a popisuje typ média vyhovující odesílateli.

3.4.1 Registrace

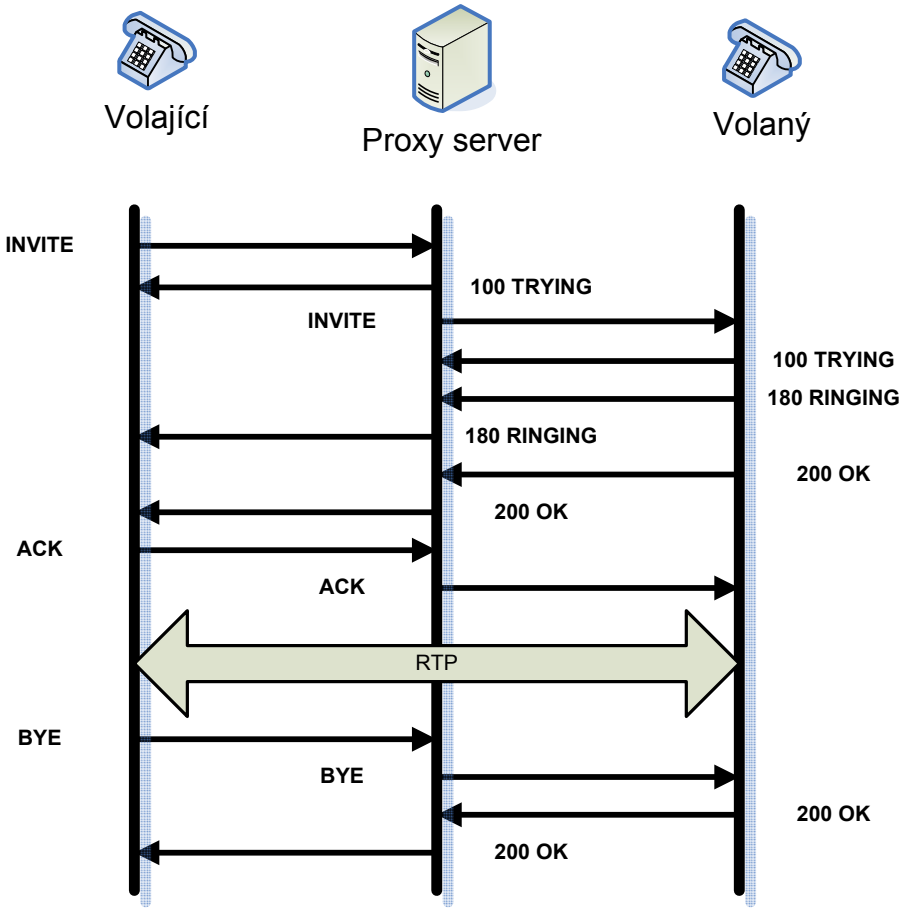
Uživatelé se musí sami registrovat na registrar serveru, aby byli dosažitelní. Registrace se skládá ze zprávy REGISTER (Obrázek 8) následovanou odpovědí 200 OK, kterou posílá registrar server v případě, že je registrace úspěšná. Pokud jsou registrace ověřovány, tak uživatel může dostat negativní odpověď 401 nebo 407, vysvětlující, že tato registrace není oprávněná. V případě, že proxy server potřebuje před zpracováním požadavku uživatele ověřit, žádá o to v odpovědi 407 Proxy Authentication Required a v hlavičce Proxy-Authenticate je obsažena výzva. Klient doplní do požadavku hlavičku Proxy-Authorization s patřičnými údaji. Celá procedura úspěšné registrace je zakončena konečnou odpovědí 200 OK [8].



Obrázek 8: Registrace klienta

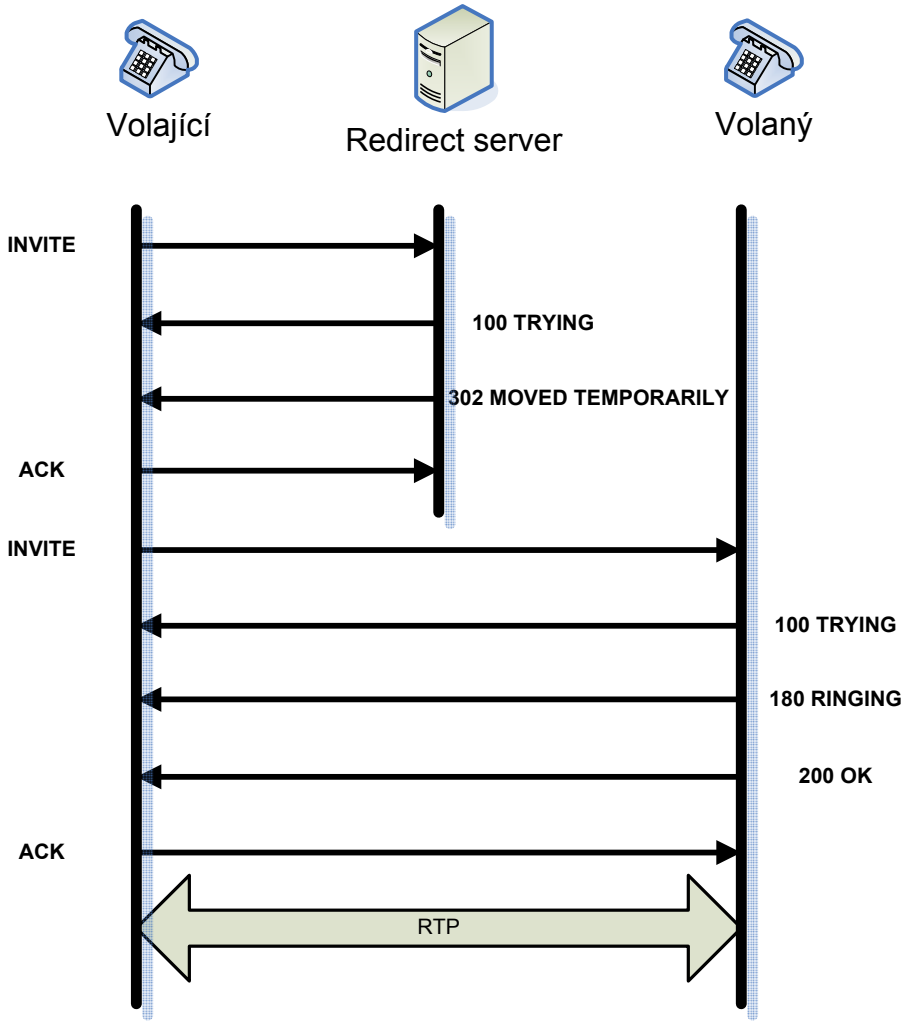
3.4.2 Navázání a ukončení spojení

Zahájení spojení obsahuje jednu žádost INVITE (Obrázek 9), obvykle odeslanou na proxy server, který odpovídá zprávou 100 Trying, což pro volajícího znamená, aby neopakoval žádost, protože je požadavek přeposíláván dál. Všechny informativní odpovědi generované volaným jsou posílané volajícímu, např. zpráva 180 Ringing. Zpráva 200 OK je generována ve chvíli, kdy volaný vyzvedne a je opakována, dokud odesílající UA neobdrží ACK od volajícího, v té chvíli je možné považovat spojení za sestavené. Spojení je ukončeno odesláním žádosti BYE v rámci dialogu zahájeného zprávou INVITE. Zprávy BYE jsou odesílány přímo z jednoho UA druhému UA, jestliže Proxy nepotřebuje zaznamenávat směrování. Účastník spojení, který zavěsí, odesílá zprávu BYE a z druhé strany, která se účastní spojení, přichází odpověď 200 OK, ta potvrzuje zprávu BYE a spojení je ukončeno [8].



Obrázek 9: Ukázka SIP spojení

Klienti většinou neznají IP adresu serveru, u kterého se nachází volaný uživatel. V tomto případě klient kontaktuje proxy nebo redirect server (Obrázek 10). Ten nejprve prostřednictvím lokalizační služby zjistí IP adresu serveru volaného uživatele. Redirect server pouze sdělí volajícímu klientu IP adresu serveru volaného uživatele a klient potom musí sám navázat spojení přímo se serverem volaného uživatele. Lokalizační služba může vrátit několik různých adres volaného uživatele. Klient pak může zkoušet kontaktovat jednotlivé adresy buď postupně nebo paralelně, dokud některý server neodpoví. Ostatním serverům se pošle žádost CANCEL na zrušení žádosti INVITE. Nakonec klient vždy potvrdí žádostí ACK, že obdržel konečnou odpověď na předchozí žádost INVITE [8].



Obrázek 10: Použití redirect serveru

4 Zabezpečení

Při návrhu protokolů k používání v počítačových sítích se ze začátku moc nezakládalo jejich zabezpečení a spíše se kladl důraz na standardizaci, funkčnost a spolehlivost. Vzhledem k zatím malému rozšíření a novosti VoIP protokolů oproti klasickým protokolům (http, ftp, atd.) jsou útoky na ně zatím relativně méně časté. Řadu dnes známých útoků na počítačové protokoly lze modifikovat tak, aby jimi bylo možné ohrožovat protokoly VoIP.

Pro zajištění bezpečnosti lze definovat tři základní vlastnosti:

Důvěrnost (utajení): zajišťuje, že zprávy mezi příjemcem a odesilatelem nemůže nikdo přečíst tak, aby z nich dostal přenášené informace.

Autentičnost: identita účastníků musí být ověřena tak, aby nemohlo dojít ke komunikaci se třetí osobou.

Integrita: posílané zprávy dorazí k příjemci bez modifikace. Tuto skutečnost si může příjemce ověřit.

4.1 Typy útoků a zabezpečení

Jednotlivé typy útoků jsou tyto:

MitM (Man in the Middle) – Tento útok představuje odposlouchávání a případné měnění dat mezi účastníky třetí osobou, která je označována jako „muž uprostřed“. Možná obrana – šifrování a testování integrity dat přenášených v signalizačním kanále [9].

Podvržení – Získání informací o spojení je spolu s odposlechem hovorů velmi atraktivní útok. V případě kompromitace SIP serveru může získat útočník plnou kontrolu nad správou příchozích i ochozích hovorů. Může hovory přesměrovávat, zaznamenávat či vkládat pakety s upraveným obsahem. Možná obrana – šifrování a integrita datového spojení, správná konfigurace proxy serverů [9].

Rušení spojení – Posíláním žádosti BYE k účastníkovy probíhajícího hovoru může dojít k ukončení spojení. Přijímání zprav z autorizovaných zdrojů je možné tento útok eliminovat. Jedná se o jeden z možných DoS útoků.

DoS útoky – Tento typ útoků (Denial of Service = odmítnutí služby) se zaměřuje na znepřístupnění služby. Útok je vyvolán velkým počtem požadavků směrovaných na konkrétní službu/server. Typickým cílem DoS útoku budou SIP proxy. Zahlcení serveru může vzniknout posláním velkého množství žádostí INVITE. Možná obrana – limity pro zpracovávání INVITE zpráv, autentizace uživatelů před posláním INVITE [9].

SPAM over VoIP– Rozesílání nevyžádaných zpráv reklamního charakteru. Někdy též nazýván Spam over Internet Telephony (SPIT). Obrana je velmi komplikovaná, nelze totiž dopředu určit, zda-li příchozí hovor je SPIT či jde u obchodníků o případného zákazníka.

Voice Phishing – Získávání osobních dat od důvěřivých lidí. Obrana závisí na opatrnosti uživatelů a jejich obeznámení s problematikou phishingu.

Přestože možnosti zabezpečení VoIP komunikace existují, v praxi se používají takřka výjimečně. Otázka zabezpečení není v současnosti zřejmě jak pro operátory, tak pro klienty natolik důležitá, aby se jí důsledně věnovali. S dalším rozšiřováním IP telefonie však důraz na zabezpečení bude jistě narůstat a snad se tedy již brzy dočkáme doby, kdy schopnost realizovat zabezpečený hovor bude stejně důležitá jako nízká cena a vysoká kvalita hovoru.

Problematiku zabezpečení lze rozdělit na dvě části: zabezpečení signalizačního kanálu a zabezpečení vlastního hovoru.

4.1.1 Zabezpečení signalizace SIP

Protože struktura zpráv protokolu SIP vychází z protokolu http je možné aplikovat bezpečnostní mechanismy dostupné pro HTTP i do prostředí SIP [9].

HTTP základní autentizace

Tato metoda zajišťuje pouze autentizaci uživatele na základě sdíleného hesla. Heslo se posílá v otevřeném tvaru, kdy je na něj použito kódování Base64, což ale neznamená, že je zašifrované. Je možné ho jednoduše odchytit a tím pádem je tato varianta autentizace nebezpečná a ve SIP verzi 2 je už zakázána. Není zajištěna integrita a utajení zpráv.

Rozšířená HTTP autentizace

Vychází ze základní autentizace, ale místo přenosu hesla v otevřené podobě se využívá hash funkce MD5, která se vytvoří z náhodně vygenerovaného řetězce, loginu a hesla. Tato varianta autentizace je bezpečnější než základní HTTP autentizace, protože zde nedochází k přenosu hesla v otevřeném tvaru. V případě volby slabého hesla je tato metoda náchylná na slovníkový útok. MD5 je ale již považován za nevyhovující hash. Není zajištěna integrita a utajení zpráv, jen autentizace.

Server zašle klientovi výzvu „401 Unauthorized“, ve které specifikuje hashovaní algoritmus (MD5), náhodnou hodnotu „nounce“ (690e7eaf) a doménu „realm“ (asterisk) [13]:

WWW-Authenticate:Digest algorithm=MD5, realm="asterisk", nonce="690e7eaf"

Klient s pomocí svého username (103), password (heslo), typu SIP žádosti (RESGISTER) a URI (sip:default) vypočítá hash, které odešle zpět serveru, který si díky tomu bude moci ověřit uživatele. Obecný výpočet:

A1 = HASH (username:realm:password)
A2 = HASH (metoda:URI)
Odpověď = HASH (A1:nounce:A2)

Příklad:

A1 = HASH (103:asterisk:heslo)
Výsledek: fa984df5e076f31657686835563bbfbc

A2 = HASH (REGISTER:sip:default)
Výsledek: fcc15d3f5434de3b1934b4975ec4b3e7

Výstup:
HASH (fa984df5e076f31657686835563bbfbc:690e7eaf:
fcc15d3f5434de3b1934b4975ec4b3e7)

Celkový výsledek:
7c951642879942e589ef58c9f7fcffe9

Tento výsledek pošle znovu ve zprávě REGISTER:

Authorization: Digest username="103", realm="asterisk", nonce="690e7eaf",
uri="sip:default", response="7c951642879942e589ef58c9f7fcffe9", algorithm=MD5

Pretty Good Privacy (PGP)

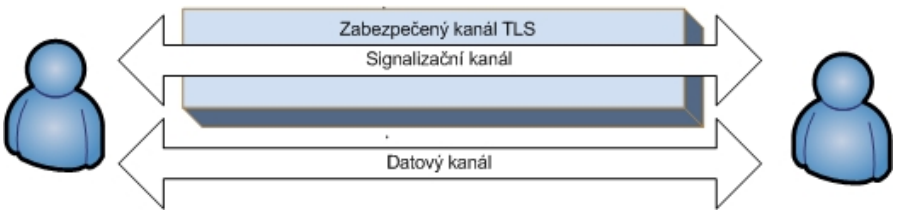
Alternativně lze použít k zabezpečení program Pretty Good Privacy. Jedná se o sadu šifrovacích programů pro bezpečnou e-mailovou komunikaci. Od verze 2 SIPu je tato volba vypuštěna na úkor standardu S/MIME. Pomocí certifikatů je zajištěna integrita a utajení zpráv.

S/MIME

MIME formát se používá u e-mailové komunikace k definici zpráv, aby mohly obsahovat nejenom text ale i obrázky, video a zvuk. S/MIME standard obsahuje metody pro zabezpečení MIME a zajistí integritu, důvěrnost a možnost ověření. Tělo MIME zprávy je symetrickým klíčem (DES, 3DES, AES) šifrováno. Zpráva se skládá z součastně ze zašifrovaných (elektrický podpis hlavičky SIP zprávy) i nezašifrovaných dat. K identifikaci koncového uživatele a jeho veřejného klíce jsou používány certifikáty. Je vyžadována spolupráce s infrastrukturou veřejných klíčů a kvůli správě bezpečnostních certifikátů také zvyšuje složitost řešení a nutná důvěryhodnost certifikační autority pro všechny strany komunikace. Podle literatury [11] je ale toto zabezpečení nepoužíváno a za 6 let ještě nebylo plně využito.

SIPS (TLS)

Použití jednotného identifikátoru zdroje URI (Uniform Ressource Identifier) ve tvaru sips:volany@nejakadomena.cz na místo sip:volany@nejakadomena.cz ve zprávě INVITE značí požadavek na zabezpečení celé cesty pomocí protokolu TLS. Protokol TLS vychází z protokolu SSL. Během přenosu SIP zpráv se jednotlivé údaje ve zprávě mění kvůli směrovacím informacím a proto je šifrované spojení ustanoveno mezi každým proxy serverem (Obrázek 11). Místo UDP zpráv se používá transportní protokol TCP a nutná je správa certifikátů resp. klíčů.



Obrázek 11: SIPS

IP Security (IPsec)

IPsec je zcela obecný nástroj k realizaci bezpečných šifrovaných spojení (Obrázek 12). Pracuje na síťové vrstvě a je tvořen protokoly – AH a ESP. Toto rozšíření je tak nezávislé na dalších (vyšších) protokolech TCP/UDP.

Bezpečnostní rozšíření vypadá následovně:

- Ověřování - při přijetí paketu může dojít k ověření zda vyslaný paket odpovídá odesilateli či zda vůbec existuje.
- Šifrování - obě strany se předem dohodnou na formě šifrování paketu. Poté dojde k zašifrování celého paketu kromě IP hlavičky, případně celého paketu a bude přidána nová IP hlavička.

Základní protokoly:

- Authentication Header (AH) - zajišťuje autentizace odesílatele a příjemce, integritu dat v hlavičce, ale vlastní data nejsou šifrována.
- Encapsulation payload security (ESP) - přidává šifrování paketů.

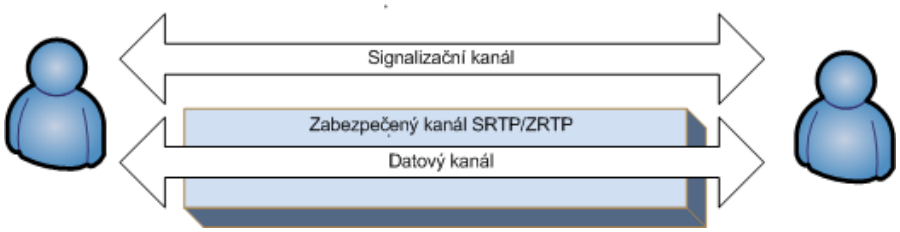
Obrázek 12: IPsec

4.1.2 Zabezpečí vlastního hovoru

Hlasové data se v IP sítích přenášejí pomocí protokolu RTP, který jako transportní protokol používá UDP Protože hlasové přenosy jsou velmi citlivé na zpoždění a jitter, nesmí žádné zabezpečení významně ovlivnit tyto parametry [9].

SRTP

Secure Real-time Transport Protocol (SRTP) představuje rozšíření RTP. Základním cílem je pro RTP a RTCP pakety zajistit utajení a autentizaci (Obrázek 13). O utajení se stará moderní algoritmus AES v režimu CTR a o integritu hashovaní algoritmus HMAC-SHA1. Data jsou pak zvětšena o autentizační hlavičku.



Obrázek 13: SRTP

IPsec

IPsec představuje druhou možnost jak zabezpečit datový proud na úrovni síťové vrstvy. Zde je možné využít stejných SA jaké se použily při zabezpečení signalizačního kanálu. Hlavní nevýhodou je velká režie IPsec (37 bytů na RTP paket v případě šifrování pomocí 3DES a 53 bytů na RTP paket v případě použití algoritmu AES-128) a špatná schopnost průchodu IPsec paketů skrze NAT. Překlad by měl být udělán před zapouzdřením paketu IP do paketu IPSec. Při zapouzdřování do paketu IPsec může výsledný paket přesáhnout maximální hodnotu MTU, což má následek ve fragmentování paketů a tím i zpomalení komunikace.

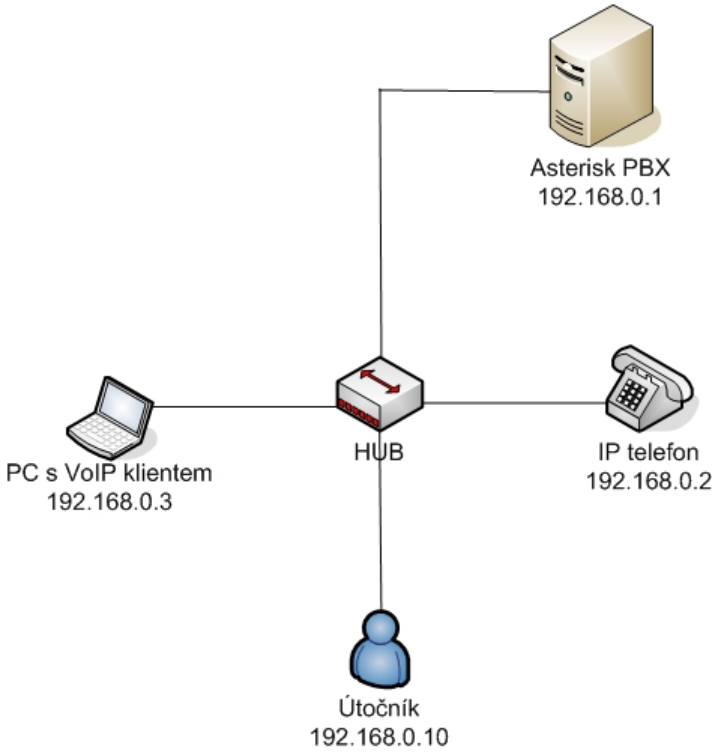
ZRTP

Protokol ZRTP rozšiřuje zabezpečený protokol SRTP tím, že k němu přidává mechanismy pro počáteční výměnu symetrických klíčů, a chrání i proti útokům typu "man in the middle". Autorem protokolu je Philipa R. Zimmermanna, který navrhl i Pretty Good Privacy.

4.2 Zapojení pokusné sítě

Pro realizaci VoIP spojení jsem vytvořil malou síť (Obrázek 14), ve které jsem realizoval jednotlivé útoky. Jako hlavní prvek VoIP sítě je ústředna, kterou jsem realizoval na počítači s operačním systémem GNU/Linux Fedora 8 a programem Asterisk 1.4. Asterisk je komplexní pobočková ústředna (PBX), která umožňuje propojit jednotlivé místní telefonní sítě pomocí internetu, funguje jako VoIP brána (MGCP, SIP, IAX, H.323), interaktivní hlasový průvodce, správa hlasových schránek a podobně. Jedná se o obecnou distribuci s podmínkami GNU (General Public Licence). Kromě výše zmiňovaného Linuxu, běží i pod jinými operačními systémy (Max OS X, OpenBSD, FreeBSD a Sun Soloris). Kromě klasického Asterisku lze pořídit i

AsterisNOW, který obsahuji i grafické rozhraní AsteriskGUI, speciální linuxovou distribuci a samotný program Asterisk. Jedná se o spojení, které zjednodušuj instalaci a i následnou správu celého systému. Výhodou Asterisku je široká internetová komunita zajišťující podporu, kterou ale neposkytuje výrobce firma Digium. Hardwarový VoIP telefon jsem použil Well 3130IF, který podporuje protokoly SIP a H.323 a může fungovat i jako routek, firewall a DHCP server. Jako další zařízení jsou použity klasické počítače s operačním systémem Microsoft Windows nebo Linux a příslušní softwarový VoIP klienti. Jako spojovací prvek jsem použil HUB, který přeposílává všechny pakety na všechny porty kromě zdrojového. Při použití SWITCHE, který posílá pakety na základě linkové adresy MAC, by se musel použít útok MAC flooding (viz kapitola 4.3.4).



Obrázek 14: Schéma zapojení

4.2.1 Instalace a nastavení

Instalace Asterisku lze provést pomocí balíčkových systémů nebo přímou kompilací. Při použití balíčkového systému stačí stáhnout z příslušného repozitáře bálíček obsahující samotnou aplikaci, informace o jejím umístění v systému, procedury které se mají provést při instalaci či odstranění aplikace a informace o tom, jaké další aplikace potřebuje program ke své činnosti (tzv. závislosti). Jedná se o jednoduchou a rychlou instalaci. Druhou možností je použít kompilaci zdrojového kódu, která přináší výhodu v nastavení aplikace přímo pro daný počítač a tedy i zrychlení, možnost výběru příslušné verze aplikace (u balíčků je většinou nutné počítat se zpožděním vydání

aktuální verze pro konkrétní linux). Nevýhodou je složitější instalace a řešení závislostí mezi aplikacemi.

Příklad kompilace (s uživatelskými právy roota, ve složce aplikace). Nejdříve se provede ověření kompilátoru a závislostí příkazem:

```
./configure
```

Samotnou kompilaci provede příkaz

```
make
```

Instalaci provedeme příkazem

```
make install
```

Samotné spuštění Asterisku v konzoli je následující:

```
asterisk -vvvc
```

V tomto konzolovém modu lze zadávat jednotlivé příkazy. Hlavni nastaveni se ale provádí v konfiguračních souborech. Pro opětovné připojení k asterisk konzoli se použije prikaz:

```
asterisk –r
```

Při změně konfigurace je nutné znovu načtení těchto dat pomocí příkazu v asterisk konzoli:

```
reload
```

Konfigurační soubory jsou tou hlavní esencí, která umožňuje dostat z Asterisku schopnosti, díky kterým je nejpoužívanějším open-source PBX řešením. Nacházejí se v adresáři /etc/asterisk/, uloženy v textových souborech, které umožňují jednoduchou administraci. Pro naše účely stačí nastavit vytáčecí plán (extensions.conf) a nastaveni SIP (sip.conf). Příklady konfiguračních souborů:

SIP.CONF

```
[general]          // Všeobecné nastavení
bindport=5060      // Port UDP, na kterém Asterisk naslouchá
context=default    // Standardní kontext, který určuje vytáčecí pravidla pro uživatele
                   v tomto kontexu.
```

disallow=all // Zakáže používání všech kodeků. Ostatní se musí povilit.
allow=ulaw
allow=alaw

[alice] // Definice jednotlivých uživatelů
type=friend // Definuje povolený způsob komunikace pro daného uživatele (user – příchozí hovory, peer - odchozí hovory, friend - obousměrná komunikace).
host= dynamic // Definuje IP adresu uživatele
username=alice // Uživatelské jméno pro registraci na asterisku.
secret=password // Heslo pro registraci
callerid=Alice <1111> // Obsahuje řetězec, který se zobrazí na telefonu volaného uživatele.

Moje nastavení:

```
[general]
port = 5060
context = default

[101]
type=friend
username=101
secret=aaaa
host=dynamic

[102]
type=friend
username=102
secret=1234
host=dynamic

[103]
type=friend
username=103
secret=heslo
host=dynamic

[104]
type=friend
username=104
host=dynamic

[105]
type=friend
username=105
md5secret=1169525b11f74902b11c14fd9232c1da
host=dynamic
```

EXTENSIONS.CONF

Musí se zde nastavit chování ústředny při vyřizování spojení. Ke každému číslu je nutné přiřadit akci, kterých může být i více. Příkazy jsou prováděny v pořadí podle prioritního čísla.

Obecný zápis syntaxe:

exten => číslo,priorita,příkaz(parametry)

Lze zde použít i zástupní znaky pro telefonní čísla:

_ – Uvozuje začátek použití částečně odpovídajícího čísla
X – Nahrazuje číslici od 0 do 9.
Z – Nahrazuje číslici od 1 do 9.
N – Nahrazuje číslici od 2 do 9.
[24-7] – Nahrazuje číslici z rozsahu 2, 4, 6, nebo 7.
. – Nahrazuje jeden nebo více znaků.

[default] // Kontext „default“ pro uživatele spadající do tohoto kontextu.
exten => 1234,1,Dial(SIP/Alice) // Pokud je voláno číslo 1234, přepojí se hovor uživateli Alice (definováno v sip.conf).

Moje nastavení:

```
[general]
static=yes
writeprotect=yes

[default]
exten => 101,1,Dial(SIP/101)
exten => 102,1,Dial(SIP/102)
exten => 103,1,Dial(SIP/103)
exten => 104,1,Dial(SIP/104)
exten => 105,1,Dial(SIP/105)
```

Nastavení jednotlivých klientů je následující:

Server (SIP server, proxy): IP adresa Asterisku,
User name (SIP number): Položka „username“ uvedená v nastavení v SIP.CONF,

Password: Položka „secret“ uvedená v nastavení v SIP.CONF,
SIP port: Port nastavený v SIP.CONF jako „bindport“.

Jako softwarové VoIP klienty jsem používal X-lite, Express Talk a SJphone ve Windows a v Linuxu Twinkle. Přehled VoIP klientů je v literatuře [13].

4.3 Útoky

Jednotlivé útoky jsem prováděl na pokusné SIP síti. Jejich realizace v reálném firemní síti by byla obtížnější vzhledem k předpokládaným bezpečnostním mechanizmům a složitější infrastruktuře.

4.3.1 Mapování sítě a zařízení

Většina použitých programů a postupů jsou napsány pod operačním systémem Linux, který v této kategorii suverénně vede a zároveň je vyzdvihována jeho větší bezpečnost oproti platformě Microsoft Windows. S těmito tvrzeními lze souhlasit, ale jistě má na to i vliv menší rozšíření Linuxu.

Inspekce sítě

Prvním krokem k jakýmkoliv nekalým úmyslům je získání pokud možno co největšího množství informací o budoucí oběti. O jeho struktuře, jednotlivých zařízení a podobně. Základní informace lze získat pomocí webových stránek, kde jednotlivé společnosti dobrovolně (např. kvůli styku s obchodními partnery) zobrazují telefonní čísla, použité technologie a další užitečné údaje, které můžou být zneužity. Lze též použít takzvaný „Google hacking“, kdy se využívá obrovská databáze Google, ve které se můžou nacházet i zajímavé informace o nastavení jednotlivých zařízení. I z nabídky zaměstnání lze vyvodit, o jaký druh technologie (v našem případě VOIP zařízení) má daná společnost zájem. Pokud shání experta na Cisco VoIP zařízení, je možné očekávat, že organizace používá zařízení od firmy Cisco a při útocích tyto informace zúročit. Další možností je využití WHOIS, což je služba, umožňující vyhledat informace o doménách a jejich vlastnících (datum registrace a vypršení domény), informace o IP adresách (v jakém bloku se adresa nachází a jaká organizace má tuto IP adresu přidělenu), informace o autonomních systémech (jaká organizace tento autonomní systém provozuje, jak je propojen s ostatními autonomními systémy apod.) a další. Služba je volně dostupná na stránkách registrátorů domén. Další možností je procházení

internetových diskuzí, kde uživatelé řeší svoje problémy s počítačovými sítěmi. Aby jim mohli pomoci „zkušenější“ uživatelé, musejí popsat přesně svůj problém. A právě při takovém popisu mohou zveřejnit i citlivé data o svém síťovém a bezpečnostním nastavení. Samozřejmě šance, že se někde nachází příspěvek k dopředu danému cíly, je skoro nulová, ale pokud by útočník hledal potencionální oběť, mohlo by mu tyto informace výrazně usnadnit útok.

Velká část útoků nepochází z vnější sítě (Internet), ale z vnitřní (LAN). Dokonce 65-80% pochází z vnitřní zóny, u které by se dalo očekávat, že bude bezpečná [14]. Někteří administrátoři se zaobírají natolik ochranou před vnějškem, že zapomínají na vnitřní stavbu sítě.

Skenování sítě

Po získání základních informací, jako je rozsah IP adres, přistoupíme k zjištění přítomnosti jednotlivých zařízení a jejich identifikaci. Typickým prvním krokem je zkoušení příkazu PING, který posílá ICMP zprávy (typ 8: ICMP ECHO REQUEST) a zjišťuje dostupnost (přijetím zprávy typu 0: IMP ECHO REPLY). Využívá se při diagnostice sítě a řešení problémů. ICMP zprávy ale můžou být na firewallu či směrovači zakázány, právě kvůli jejich zneužití.

Zadávání jednotlivých IP adres je ale časově náročné, a proto je lepší použít linuxový program FPING [15], který dokáže procházet zadaný rozsah sítě. Přepínač „-a“ zobrazí jen živé zařízení, s pomocí „-g“ se prochází zadaný rozsah sítě (v našem případě od 192.168.0.1 po 192.168.0.254) a „-s“ vypíše souhrnné informace.

```
fping -a -g 192.168.0.1 192.168.0.254
```

Výstup:

```
192.168.0.1 is alive
192.168.0.2 is alive
192.168.0.3 is alive
192.168.0.10 is alive
```

Velmi oblíbený linuxový nástroj NMAP (dostupný z [15]) dokáže mimo jiné zastat stejné funkce jako FPING a ještě funguje jako skener portů a identifikátor operačního systému. Na lokální síti zjistí i linkovou MAC adresu.

```
nmap -sP 192.168.0.1-254
```

Výstup:

```
Starting Nmap 4.52 ( http://insecure.org ) at 2008-04-20 17:17 CEST
Host 192.168.0.1 appears to be up.
```

```
MAC Address: 00:0A:E4:4A:41:09 (Wistron)
Host 192.168.0.2 appears to be up.
MAC Address: 00:09:45:60:88:B7 (Palmmicro Communications)
Host 192.168.0.3 appears to be up.
MAC Address: 00:0A:E4:EB:53:04 (Wistron)
Host 192.168.0.10 appears to be up.
Nmap done: 254 IP addresses (4 hosts up) scanned in 6.325 seconds
```

Pokud je ICMP zpráva typu 8 blokována, lze vyzkoušet ostatní druhy, které nemusejí být zablokovány. Podle RFC 1122 (Requirements for Internet Hosts - Communication Layers) by měla být funkce ECHO REQUEST/REPLY povolená.

Kromě ICMP zpráv lze využít i protokol ARP, který slouží v jednom segmentu sítě k získání linkové MAC adresy ze síťové IP adresy. Vysílá se ARP dotaz k získání určité IP adresy broadcastem, díky kterému odpoví všechny zařízení v lokální síti. Všechny stanice, které obdrží tento dotaz si zapíší MAC adresu a IP adresu do svojí ARP tabulky. Tato tabulka se musí pravidelně obnovovat. Příjemce, u kterého se bude IP adresa shodovat s IP adresou v dotazu, pošle ARP odpověď s vlastní IP a MAC adresou. Tyto dotazy nejsou zabezpečeny a lze díky nim podvrhnout svoji identitu (viz dále). ARP protokol je základním kamenem ethernetových sítí, proto jej nelze nijak vypnou či blokovat. Jeho funkčnost je ale jen na lokálních sítích oddělenými směrovači či VLAN (což je asi jediný způsob, jak omezit arping).

V Linuxu lze použít program ARPING (dostupný z [15]), který přeloží zadanou IP adresu na MAC adresu a zároveň zjistí dostupnost zařízení. Přepínačem „-I“ se vybere síťový adapter, přes který se budou posílat dotazy (obvykle eth0 či eth1), „-c“ určí počet opakování dotazu.

```
arping -I eth1 -c 3 192.168.0.1
```

Výstup:

```
ARPING 192.168.0.1 from 192.168.0.10 eth1
Unicast reply from 192.168.0.1 [00:0A:E4:4A:41:09]  6.815ms
Unicast reply from 192.168.0.1 [00:0A:E4:4A:41:09]  0.825ms
Unicast reply from 192.168.0.1 [00:0A:E4:4A:41:09]  0.847ms
Sent 3 probes (1 broadcast(s))
Received 3 response(s)
```

Díky skenování sítě zjistíme, na jaké zařízení se máme soustředit v dalších krocích útoků.

Skenování portů a služeb

Skenování portů slouží útočníkovi k zjištění, jaké služby na cílovém zařízení běží. Tím se dozví, na kterých portech cílové zařízení naslouchá. Většina služeb funguje na známých portech (například telnet na portu 23, SMTP na portu 25), proto lze z otevřených portů vyvodit spuštěné služby. Ke skenování lze použít různé programy, například NETCAT či STROBE, ale nejvšestrannější je linuxový program NMAP. Protokol SIP obvykle potřebuje otevřené porty TCP/UDP 5060, protokol SCCP 2000 až 2002. Pro ověření UDP je potřeba zadat přepínač „-sU“, pro zjištění jaká služba na daném portu pracuje „-sV“, při zadání „-PN“ program NMAP nezjišťuje pingem dostupnost daného zařízení (při zakázaném ICMP) a vynutí se přímo skenování , „-vv“ bude vypisován podrobný zpráva činnosti programu a „-p“ určí skenované porty.

```
nmap -PN -sU 192.168.0.1
```

Výstup:

```
Starting Nmap 4.52 ( http://insecure.org ) at 2008-04-20 17:28 CEST
Interesting ports on 192.168.0.1:
Not shown: 1484 closed ports
PORT      STATE        SERVICE
123/udp   open|filtered ntp
5060/udp  open|filtered sip
5353/udp  open|filtered zeroconf
32770/udp open|filtered sometimes-rpc4
MAC Address: 00:0A:E4:4A:41:09 (Wistron)
Nmap done: 1 IP address (1 host up) scanned in 1480.636 seconds
```

Jak je vidět z výstupu, na zařízení je otevřený port 5060, což je i správně detekováno jak port pro SIP komunikaci. Možnost použití jiného portu než je standardní 5060 pro SIP povede k horší detekci služeb na daném stroji, ale zvyšuje nekompatibilitu mezi nenastavenými zařízeními. Asterisk obvykle má otevřený port 5060 pro SIP signalizaci, port 2000/TCP pro Cisco protokol Skinny a 4569/TCP pro komunikaci protokolem IAX. Jednotlivé porty mohou být ve stavech open (zařízení naslouchá na tomto portu), closed (žádné aplikace, které by naslouchaly), filtered (existuje zde překážka (firewall, síťový filtr, …), která brání zjistit plně stav portu) a unfiltered (port je přístupný, ale nelze zjistit stav open nebo closed).

Jakmile už známe IP adresy stanic a jejich používané porty, lze identifikovat jaké operační sytému (firmware) na nich pracují. Bude nám znovu nápomocen NMAP a to s přepínačem „-O“ (případně „-A“). Přepínač „--fuzzy“ způsobí, že se vypíšou možne verze OS s procentním odhadem správnosti.

```
nmap -O -v -PN 192.168.0.1
```

Výstup:

```
Initiating OS detection (try #1) against 192.168.0.1
Host 192.168.0.1 appears to be up ... good.
Interesting ports on 192.168.0.1:
Not shown: 1709 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
2000/tcp open  callbook
MAC Address: 00:0A:E4:4A:41:09 (Wistron)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.22 - 2.6.23
Uptime: 0.341 days
Network Distance: 1 hop
```

Z výstupu je patrno, že počítač je identifikován jako Linux s jádrem 2.6, což odpovídá nainstalované Fedoře 8. Další podobné programy jsou XPROBE2 nebo p0f. Nejlepší obranou proti skenování je vypnutí nepoužívaných služeb (WWW, telnet a podobně).

Obdobně u VoIP telefonu WELL (192.168.0.2):

```
Not shown: 1708 closed ports
PORT     STATE    SERVICE
23/tcp   open     telnet
80/tcp   filtered http
1025/tcp open     NFS-or-IIS
MAC Address: 00:09:45:60:88:B7 (Palmmicro Communications)
Device type: VoIP gateway
Running: ShoreTel embedded
OS details: ShoreTel ShoreGear-T1 VoIP switch
```

Detekce není dokonalá, telefon je rozpoznán jako VoIP switch. Tento telefon obsahuje i switch a router, díky kterému lze za telefon zapojit další zařízení, a proto jej pravděpodobně NMAP detekoval právě jako switch.

Nyní již víme, co za zařízení jsou v síti a můžeme se zaměřit na hlavní server – Asterisk.

Zjištění nastavení SIP

Nyní už víme, na jakých zařízeních jsou aktivní SIP služby, a proto se zaměříme na zjištění jaké SIP účty jsou používány. Využívají se k tomu SIP zprávy INVITE (žádost o spojení), REGISTER (žádost o registraci) a OPTIONS (žádost o podporované funkce.)

Zpráva poslaná Asterisku (192.168.0.1) od uživatele (192.168.0.3) při použití špatného přihlašovacího údaje (500):

```
REGISTER sip:192.168.0.1 SIP/2.0
*
From: 500 <sip:500@192.168.0.3>;tag=120975822
To: 500 <sip:500@192.168.0.3>
Contact: "500" <sip:506@192.168.0.3:5060>
*
```

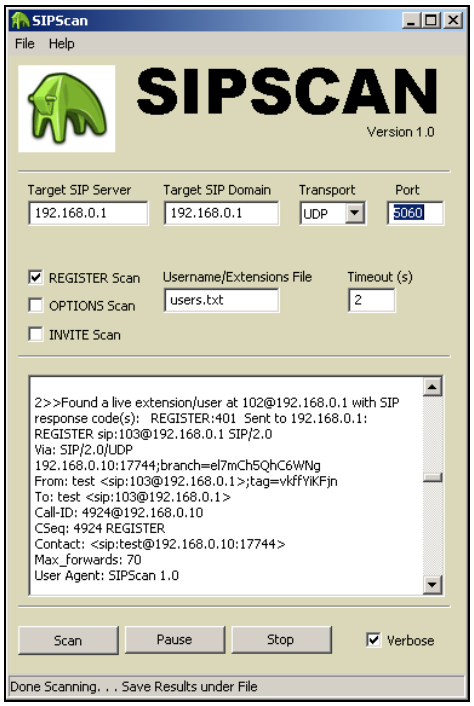
Odpověd:

```
SIP/2.0 404 Not found
```

Při použití korektního údaje (username 103 místo 500) je odpověď:

```
SIP/2.0 200 OK
```

Díky těmto odpovědím, lze zjistit, kteří uživatelé mají přistup k Asterisku. Pro posílání REGISTER zpráv použijeme program SIPSCAN, který celou činnost zautomatizuje. V textovém souboru „users.txt“ jsou definovaní uživatelé, kteří se budou pokoušet přihlásit k ústředně. Samozřejmě, pokud bychom chtěli vyzkoušet všechny možné kombinace, byl by seznam obrovský. Předpokládáme ale, že v ústředně jsou použity běžné čísla, tudíž obvykle 100, 101, …., 1000, …., 9999.



Obrázek 15: Program SIPSCAN

Nastavení:

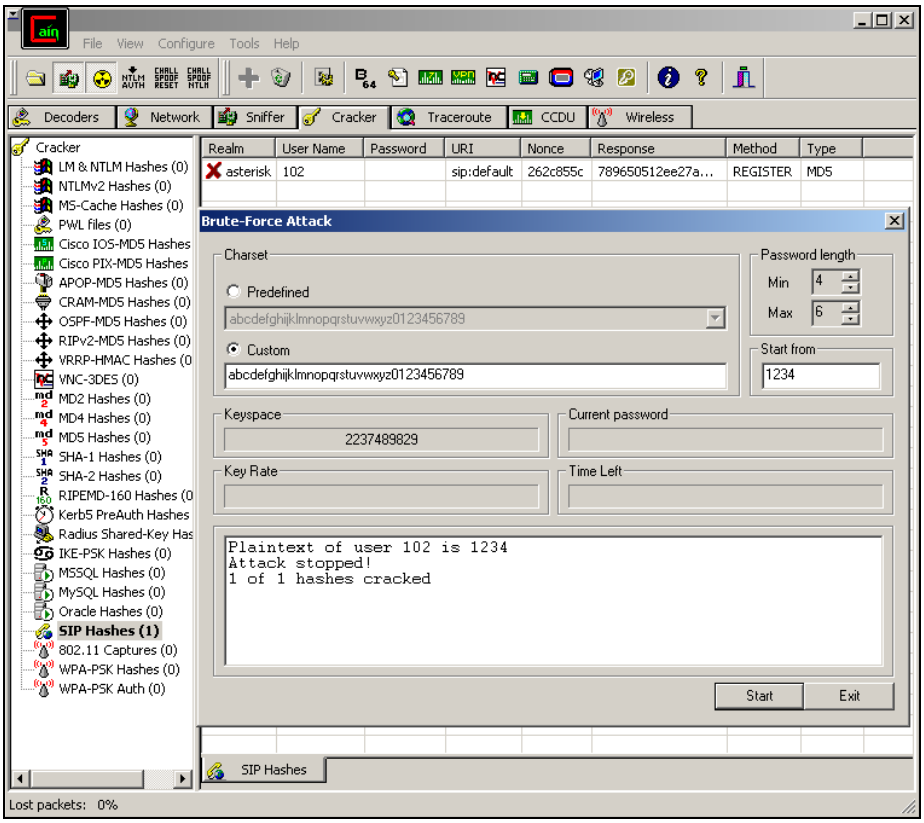
- Target SIP server a Target SIP Domain = IP adresa Astrisku
- Transport = protokol UDP
- Port = obvyklý port 5060
- Scan = zvoleno Register
- Username/Extensions File = textový soubor s uvedenými názvy uživatelů
- Timeout = časoví interval mezi dotazy

Program SIPSCAN nám vypíše, jaké údaje měli správnou odezvu na ústředně:

Found a live extension/user at 100@192.168.0.1 with SIP response code(s): REGISTER:200

Nyní již známe strukturu sítě, jednotlivé služby a uživatelské účty, které jsou aktivní na Asterisku.

K zjištění hesla, které slouží k přihlášení k Asterisku, využijeme program Cain&Abel (Obrázek 16). Jedná se o skvělý sniffer (zachytává a analyzuje pakety) a prolamovač hesel (dostupný z [15]). V záložkách Sniffer-Passwords-SIP zobrazuje zachycenou SIP komunikaci, kterou dokáže poslat do záložky Cracker (Send to Cracker). Zde se může použít slovníkový útok či útok brutální silou. U slovníkového útoku se musí zadat slovník slov, které se budou zkoušet pro výpočet hashe, aby souhlasil s odchyceným. Při brutálním útoku se vyberou zkoušené znaky (písmena, číslice, speciální znaky) a rozsah velikosti hesla. Jednoduchá hesla typu „1234“ jsou zjištěna okamžitě. Pokud se použijí hesla, které jsou delší, kombinují více typů znaků a netvoří známé slova, čas k rozluštění neúměrně stoupá.



Obrázek 16: Útok na hesla přes Cain&Abel

Soubor programů SIPVicious slouží k auditu VoIP systémů postavených na protokolu SIP (dostupné na http://code.google.com/p/sipvicious/). Jedná se o skripty napsané v Pythonu, tudíž je lze spouštět na různých operačních systémech. Součástí je Svmap, který slouží jako skener SIP zařízení v daném rozsahu IP adres.

Příklad použití Svmap:

```
./svmap.py -v -p 5060-5070 192.168.0.1/24

| SIP Device        | User Agent                                |
-------------------------------------------------------------
| 192.168.0.10:5063 | Twinkle/1.2                               |
| 192.168.0.3:5060  | SJphone/1.65.377a (SJ Labs)               |
| 192.168.0.10:5060 | OpenSER (1.3.1-tls (i386/linux))          |
| 192.168.0.1:5061  | Twinkle/1.0.1                             |
| 192.168.0.1:5060  | Asterisk PBX                              |
```

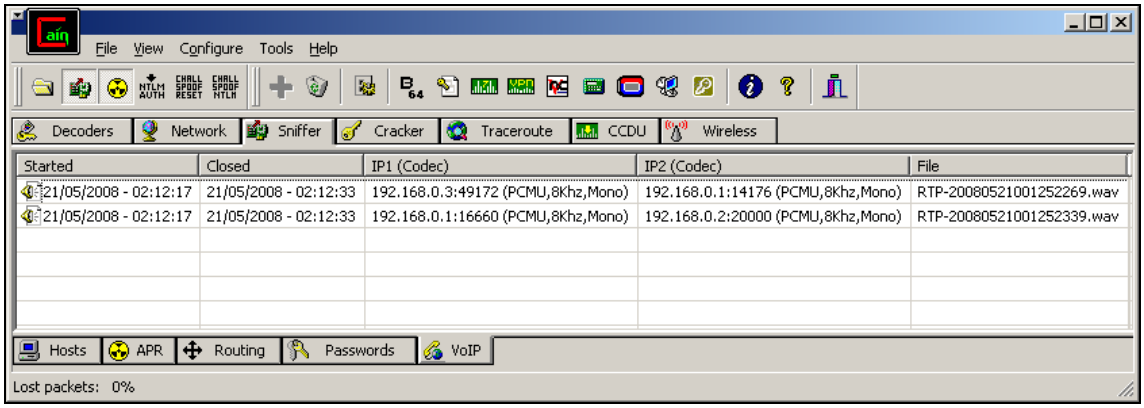
Jednotlivé postupy na skenování sítě a služeb je vhodné kombinovat, protože ne vždy každý postup správně funguje. Zjištění struktury sítě je důležitý prvek v provádění jakýkoliv dalších útoků.

Obrana proti zjišťování nastavení je obtížné, protože ústředna musí zpracovávat SIP dotazy a odpovídat na ně. Oddělení provozu VoIP sítě od datové sítě pomocí VLAN je doporučená metoda ochrany [16].

4.3.2 Zachycení RTP

Pro SIP i H.323 se hlas přenáší v RTP/UDP paketech. Protokol RTP nešifruje přenášený hlas a pokud jdou data odchytávat, například s pomocí programu WIRESHARK (dostupný z [15]), lze je analyzovat a přehrát přenášený hlas. Wireshark je protokolový analyzátor a paketový sniffer. Odchytává síťovou komunikaci a pomáhá při problémech v síti. Pokud je síťové zařízení nastaveno v promiskuitním módu, přijímá všechny pakety, které procházejí sítí a ne pouze ty, jenž jsou směrovány přímo jemu. Vedlejší funkci programu Wireshark je i analyzování VoIP komunikace. Podporuje SIP, H.323 a RTP. Pokud je použit zvukový kodek G.711, dokáže zaznamenat a přehrát zvuk. Kodeky, které jsou zatíženy licencí (jako například G.729), nepodporuje.

Lepším řešením je použití programu Cain&Abel (dostupný z [15]), který je sice označován jako dešifrátor hesel, ale dokáže i sledovat síťový provoz. Obrázek 17 znázorňuje zachycený RTP přenos mezi 2 stanicemi (každý směr je zvlášť). Po zvolení „Play“ je záznam jednoduše přehrán.



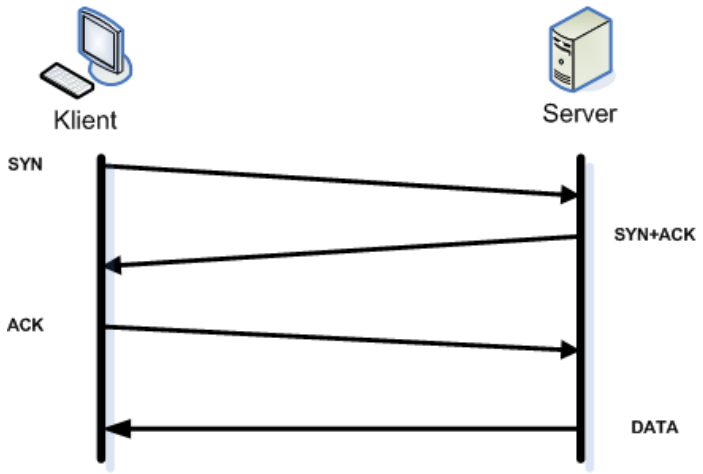
Obrázek 17: Záznam RTP přes Cain&Abel

4.3.3 Denial of service

DoS alias Denial of Service značí odmítnutí služby. Jedná se tedy o útoky, které zahlcující požadavky danou službu/zařízení a způsobí tím pád, nefunkčnost a nedostupnost pro ostatní uživatele.

Jedním z typů DoS je i flood (záplavový) útok. Spočívá v posílání mnoha zpráv, které zahltí oběť. Existuje jich více typů lišící se podle použitého protokolu. Například u protokolu ICMP se posílají zprávy typu ECHO, což odpovídá použití příkazu ping. Přijímací strana odpovídá ICMP ECHO REPLY a přitom zachovává velikost paketu. Tudíž linka mezi útočníkem a obětí bude dvojnásobně zahlcena pakety. Útočník ale potřebuje rychlejší linku než má oběť.

Další typ útoků je SYN FLOOD. Při navazování komunikace protokolem TCP se nejdříve pošle paket s příznakem SYN. Následuje odpověď serveru SYN+ACK (Obrázek 18). Pokud server nechce přijmout spojení, tak neodpovídá nebo pošle příznak RST.



Obrázek 18: TCP Handshake

Útok spočívá v posílání více paketů s příznakem SYN, díky kterým si musí server pro takovéto spojení alokuje systémové prostředky. Při hromadném posílání SYN paketů dojde k znepřístupnění dané služby. Pro realizaci lze použít linuxový program HPING2 (dostupný z [15]). Dokáže posílat několik takových paketů a nastavit náhodnou zdrojovou adresu. Přepínač „-S“ značí příznak SYN, „-p“ je cílový port, dále je nastavena náhodná zdrojová adresa, rychlost posílání paketů a cílová IP adresa:

```
hping -S --rand-source --faster -p 5060 --udp 192.168.0.2
```

Obdobně lze posílat pakety na UDP/5060 (SIP):

```
hping --rand-source --faster -p 5060 -udp 192.168.0.2
```

Cílový telefon Well nebyl schopen přijmout hovor nebo bylo spojení velmi nekvalitní. Útok je velice jednoduchý, mnoho počítačů proti němu není ochráněno dodnes. Výhodou je, že není potřeba velkého datového toku (mohou se posílat i malé pakety) a lze falšovat IP adresu odesilatele. Obrana spočívá v nastavení firewallu a použití SYN cookies. Problém telefonu Well spočívá ve stejné reakci útoku s libovolným portem. Vnitřní systém je málo výkonný a vždy dochází k odepření služby.

K posíláni UDP paketů lze použit i malý program UDPFLOOD (dostupný na [16]). Syntaxe je následující: IP adresa zdroje a cíle, port zdroje a cíle, počet odeslaných paketů.

```
./udpflood 192.168.0.10 192.168.0.1 9 5060 100000
```

Při útoku na server Asterisk (192.168.0.1) nedošlo k přerušení probíhajících volání, bylo patrné nepatrné zvýšení zátěže procesoru. Telefon Well se choval obdobně ale s výpadky hlasu. Při několikanásobném provedení útoku nebyl telefon schopen žádné komunikace. U softwarového SIP klienta SJphone (192.168.0.3) došlo jen ke zhoršení kvality hovoru.

Mezi DoS útoky lze ve VoIP zařadit i posílání SIP zpráv, které znemožňují nebo ukončují hovor. Jednou z možností je použití INVITE zpráv, které slouží k navázání spojení. Použití programu INVITEFLOOD (dostupný z [16]) je obdobné jako u UDPFLOOD, přibylo jen uživatelské jméno cíle (102).

```
./inviteflood eth0 102 192.168.0.2 192.168.0.2 10000
```

Telefon Well reagoval na této útok zvoněním, následné přijmutí hovoru (nedochází k přenosu RTP, tudíž na lince není nic slyšet) a položení má za následek další zvonění. Telefon je pro ostatní účastníky nedostupný a SIP odpověď na volání je 486 Busy Here (Zaneprázdněný)

Při pokusu na server Asterisk vyvolá ústředna varování „Too many open files“ (příliš mnoho otevřených souborů). Při pokusu o volání je SIP odpověď 500 Server Internal Error (Vnitřní chyba serveru).

K ukončení probíhajících hovorů slouží SIP žádost BYE (Obrázek 9). K poslaní této žádosti postačí program TEARDOWN [16] a pro zachytávání přenosu WIRESHARK. Jednotlivé SIP spojení jsou označeny neměnnými položkami Call-ID, To Tag a From Tag v celém průběhu spojení. Pomocí WIRESHAR tyto údaje odchytíme v paketu SIP:

```
ACK sip:103@192.168.0.3 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bK581fb1c8;rport
From: "102" <sip:102@192.168.0.1>;tag=as0d32d2ac
To: <sip:103@192.168.0.3>;tag=49bf569325
Contact: <sip:102@192.168.0.1>
Call-ID: 788d29257c3e65b252de85e422813fb2@192.168.0.1
CSeq: 102 ACK
User-Agent: Asterisk PBX
Max-Forwards: 70
Content-Length: 0
```

Tyto údaje se použijí k ukončení spojení pomocí TEARDOWN (syntaxe: síťové rozhraní, SIP uživatelské jméno, IP adresa serveru, IP adresa telefonu, Call-ID, To Tag a From Tag):

```
./teardown eth1 102 192.168.0.1 192.168.0.2
788d29257c3e65b252de85e422813fb2@192.168.0.1 as0d32d2ac 49bf569325
```

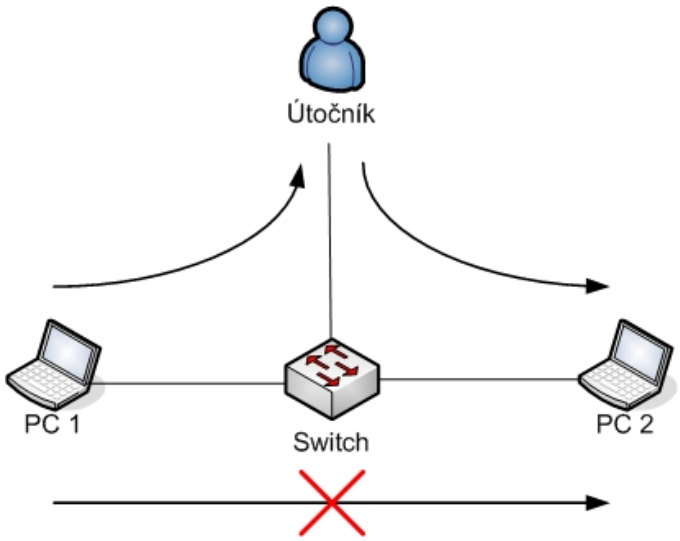
Výstup:

```
teardown - Version 1.0
        Feb. 17, 2006

source IPv4 addr:port   = 192.168.0.10:9
dest   IPv4 addr:port   = 192.168.0.2:5060
targeted UA             = 102@192.168.0.1
From Tag                = as0d32d2ac
To Tag                  = 49bf569325
Call ID                 = 788d29257c3e65b252de85e422813fb2@192.168.0.1
```

Spojení je ukončeno vytvořeným BYE paketem, který může být poslán buď serveru nebo samotnému klientovi. Žádost BYE není chráněna rozšířenou http autentizací a proto je při splnění podmínek identifikace hovoru (call-id a tagy) kladně vyřízena.

4.3.4 Man in the middle

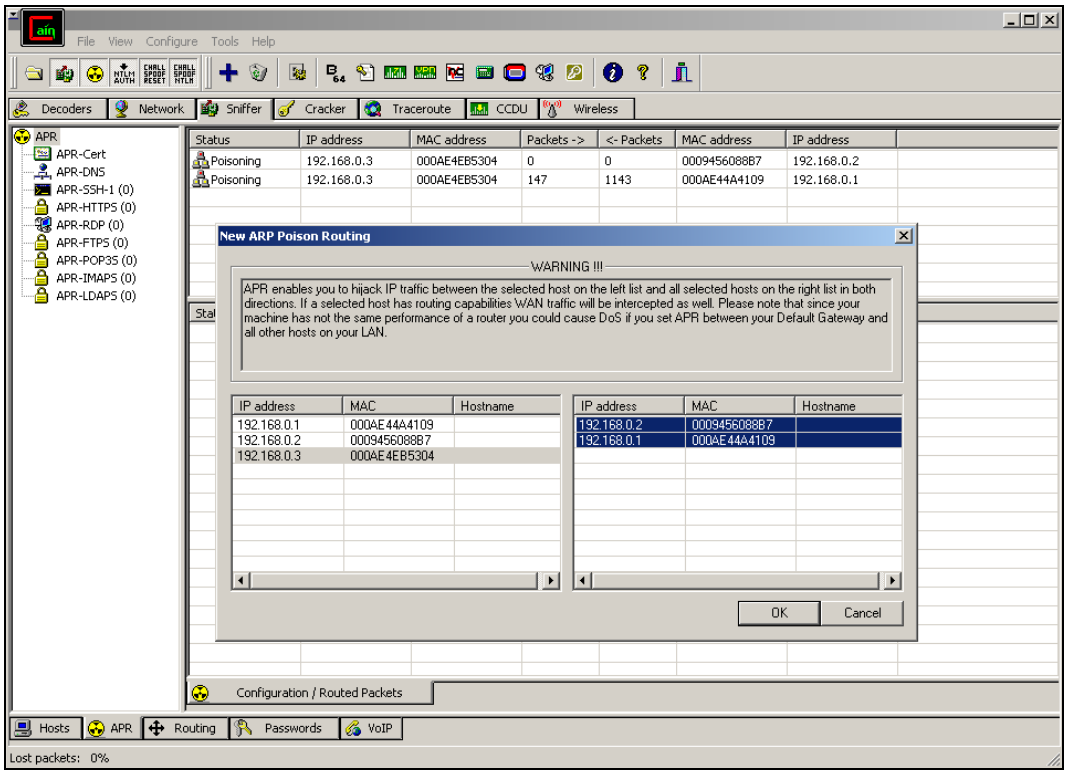
„Muž uprostřed“ alias „Man in the middle” (MiM) je útok, při kterém je datový tok přesměrován přes útočníka, který může odposlouchávat, měnit či jinak manipulovat s procházejícími daty (Obrázek 19). Existují více druhů MiM útoků (ARP Cache poisoning, DHCP Spoofing, ICMP Redirecting, Port stealing, DNS Spoofing a další).



Obrázek 19: Man in the middle

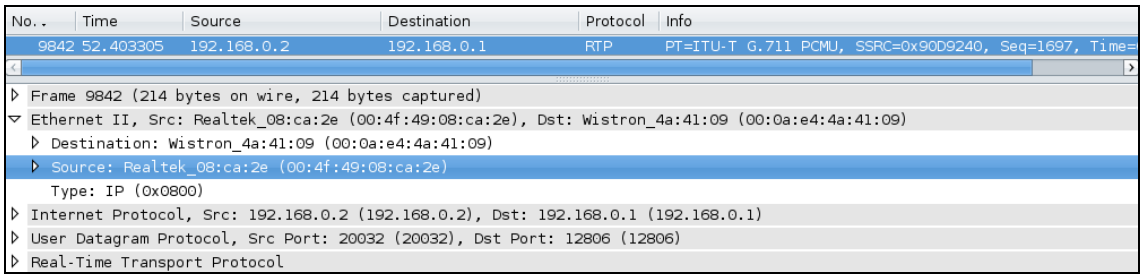
ARP cache poisoning využívá vlastnost sítě, kdy každé koncové zařízení si uchovává tabulku (ARP cache), ve které jsou spárovány IP adresy a MAC adresy linkové vrstvy OSI modelu. Tato tabulka se musí pravidelně obnovovat. Útočník tedy pravidelně posílá stanicím PC1 a PC2 podvržené ARP zprávy, že protilehlá stanice PC má MAC adresu útočníka (Obrázek 19). Při posílání neexistujících MAC adres dojde k přerušení komunikace a jedná se tedy o DoS útok.

Jednoduchá realizace tohoto útoku je s pomocí programů Dsniff v Linuxu a Cain&Abel ve Windows. V Cain&Abel stačí vybrat Sniffer -> ARP -> New ARP Poison Routing, kdy v levé části dialogového okna vybereme stanici, kterou chceme odposlouchávat a v pravé části zvolíme stanice, pro které přiřadíme naši (útočníkovu) MAC adresu (Obrázek 20).



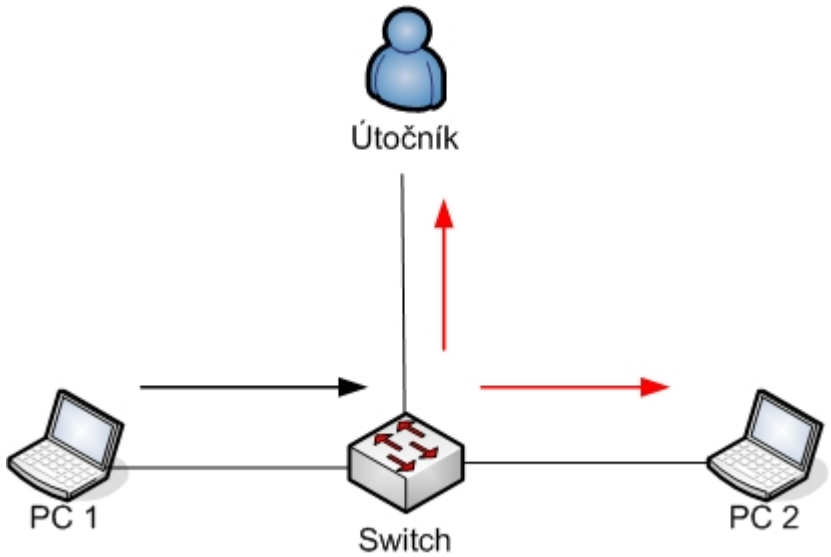
Obrázek 20: ARP cache poisoning

Následně lze s využitím znalostí z kapitoly „4.3.2 Zachycení RTP“ jednoduše daný hovor odposlouchávat. Tento útok lze i jednoduše zapínat a vypínat, hovořící strany nic nepoznají, ale zachycena je pak jen část hovoru během aktivního ARP cache poisoningu. Obrázek 21 ukazuje zachycený RTP paket s podvrženou MAC adresou (Source) při přenosu mezi telefonem (192.168.0.1) a ústřednou (192.168.2). MAC adresa 00-4F-49-08-CA-2E patří útočníkovi (192.168.0.10).



Obrázek 21: Podvržená MAC adresa

Specifický je MAC flooding, při kterém se docílí, že switch přeposílá všechny pakety a jakýkoliv počítač připojený na switch vidí veškerý provoz. Nezabrání se ale doručení původních dat (Obrázek 22).



Obrázek 22: MAC flooding

Jeho realizace spočívá v naplnění CAM tabulky (Content Addressable Memory table) na switchi, ve které se ukládají MAC adresy k příslušným portům, aby byl switch schopen přepínat datový tok jen k příslušnému cíly (portu). Tato tabulka je omezená a po naplnění by se měl switch chovat jako obyčejný HUB.

K realizaci jsem použil program MACOF ze souboru programů DSNIFF (dostupný na [15]). Generuje přibližně 1000 paketů za sekundu.

```
macof –i eth1
```

Další možný způsob je s použitím programu ARPTOXIN (dostupný na [15]). Generuje přibližně 2000 paketů za sekundu při zadání rozmezí mezi vysíláním na 0 ms („-p“). Přepínač „-d“ určuje síťové rozhraní.

```
Arptoxin –d 2 –p 0
```

Oba programy fungovali jen na switchi bez uvedeného názvu výrobce (noname). Kdy switch začal přeposílávat všechny dotazy na ostatní porty. U výrobku firmy WLINX SW-005CM-X se ale tato situace neopakovala a ostatní připojená zařízení neviděla kompletní provoz v dané síti.

4.4 Ochrana

Kromě klasických náležitostí, které jsou nutné v obyčejné datové sítí splnit pro zachování funkčnost a bezpečnosti, je nutné ve VoIP sítích i tyto následující drobné zabezpečení zajistit.

- Uložení hesel u Asterisku

Bez použití hesla (položka „secret“ v „sip.conf“) se lze připojit k ústředně, aniž bychom museli ověřovat uživatele. Stačila by tedy jen znalost jeho uživatelského jména („username“). Heslo je ale uloženo jen jako obyčejný text, a proto je lepší použití hash. Místo položky „secret“ se nastaví „md5secret“. Výpočet hashe pomocí MD5 lze například v Linuxu realizovat takto (zadávané hodnoty jsou obdobné jako v kapitole 4.1.1, Rozšířená HTTP autentizace):

```
echo –n "username:realm:secret" | md5sum
```

Pro uživatele „105“, realm „asterisk“ a heslo „tajne“ je výsledný hash:

```
echo –n "105:asterisk:tajne" | md5sum
```

Výstup:
1169525b11f74902b11c14fd9232c1da

- Přístup na webové rozhraní

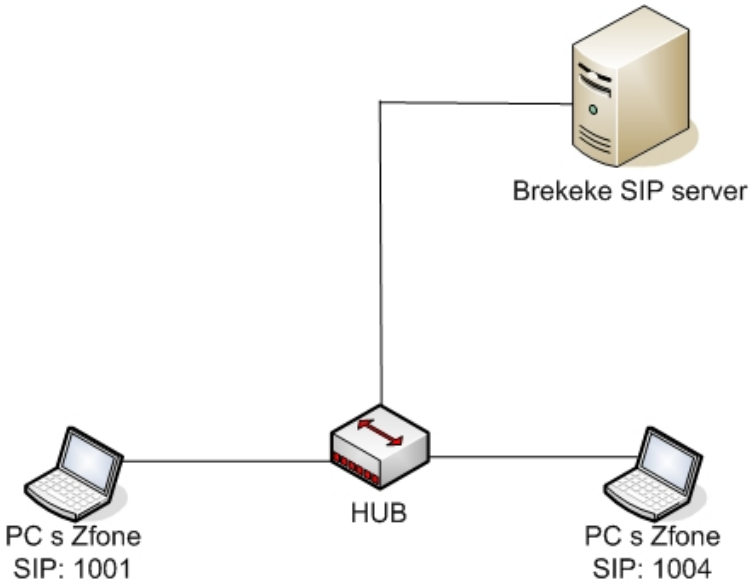
Pro ovladání a nastavování hardwarových VoIP telefonů se nejčastěji používá webové rozhraní. Do prohlížeče stačí zadat IP adresa telefonu a objeví se přihlašovací stránka. Pokud zůstane login a heslo nezměněno, lze obvykle z manuálu dostupného na internetu získat přednastavené údaje, které umožní vstup do nastavení. Pro telefon Well je kombinace například admin/admin. Je bezpodmínečně nutné změnit výchozí nastavení.

4.4.1 ZRTP

Pro zabezpečení hlasu je třeba použít jeden ze způsobů uvedených v kapitole 4.1.2. Vybral jsem variantu použití protokolu ZRTP místo RTP. Autorem je Phil Zimmermann, který je znám jako tvůrce PGP používaného k šifrované mailové komunikaci. ZRTP je rozšíření k SRTP, kdy navíc ještě pomocí Diffie-Hellmanova (DH) algoritmu vyměňují klíče, které jsou následně použity k přenosu tajných dat. Nevyužívají se certifikační autority, PKI (infrastruktura veřejných klíčů) či jiné „cizí“ prostředky k sestavení zabezpečených spojení, mimo výměnu klíčů mezi uživateli. Samotný DH algoritmus je náchylný na útok „man in the middle“ a proto se navíc ještě používá krátký autorizační text SAS. Tento SAS, což je hash z obou tajných hodnot DH algoritmu, si účastnící hovoru mezi sebou řeknou a pokud se shoduje, je velká pravděpodobnost ochrany před MiM útokem [11]. Protokol ZRTP používá program

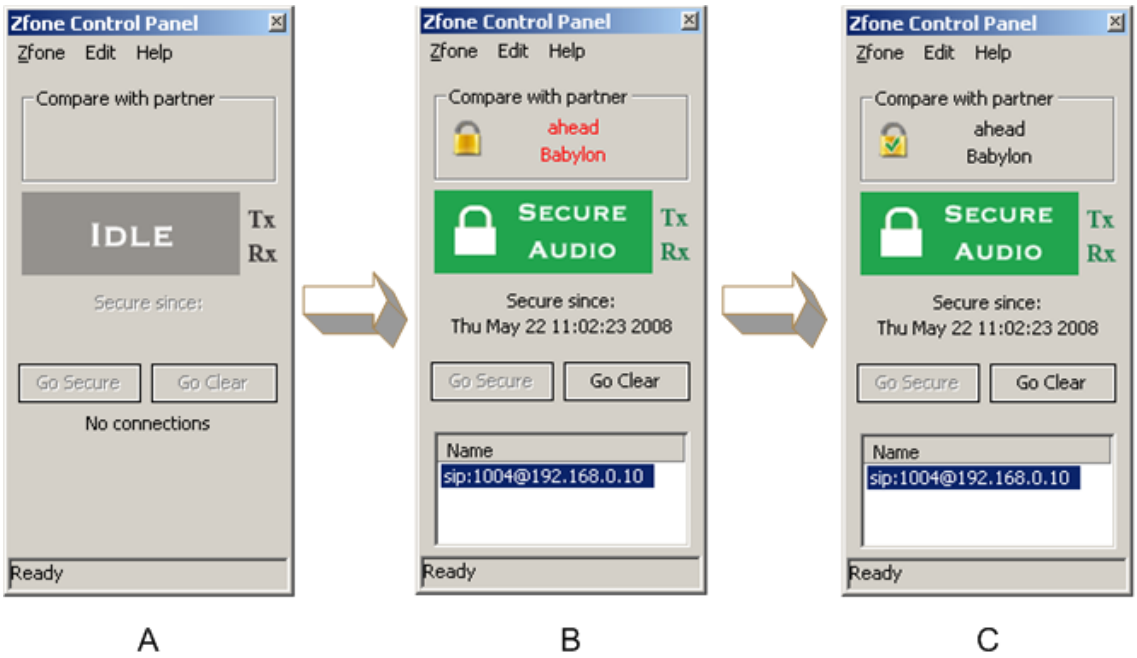
Zfone, který neslouží přímo k volání, ale je vrstvou pod softwarovým VoIP telefonem, sloužící k šifrování hovoru.

Pro pokusné použití Zfone jsem nemohl použít stávající ústřednu Asterisk, která není kompatibilní se ZRTP a potřebuje plug-in od výrobce Zfone. Tento plug-in ale není veřejně přístupný, a proto jsem použil Brekeke SIP Server (dostupný z http://www.brekeke.com), který funguje jen jako registrační a proxy SIP server. Použil jsem Windows verzi spravovanou přes webové rozhraní. Samotný Zfone není implementovaný v dostupných hardwarových VoIP telefonech, proto je třeba použít podporované softwarové klienty (například SJphone). V zapojení nebyl tedy použit VoIP telefon Well (Obrázek 23).



Obrázek 23: Zapojení SIP Serveru

Na počítačích byl spuštěn SJphone a následně Zfone. Pokud není spuštěn na obou stranách Zfone, komunikace sice není zabezpečena, ale je umožněno normální spojení. Obrázek 24 zobrazuje stavy Zfone při komunikaci, kdy stav „A“ je před uskutečněním hovoru, „B“ je zabezpečený hovor a stav „C“ je po výměně autorizačního textu (ahead Babylon) a odsouhlasením jeho správnosti v nastavení Zfone. Díky tomuto potvrzení jsou i následné hovory zabezpečeny použitím hashe z prvního ověřeného spojení.



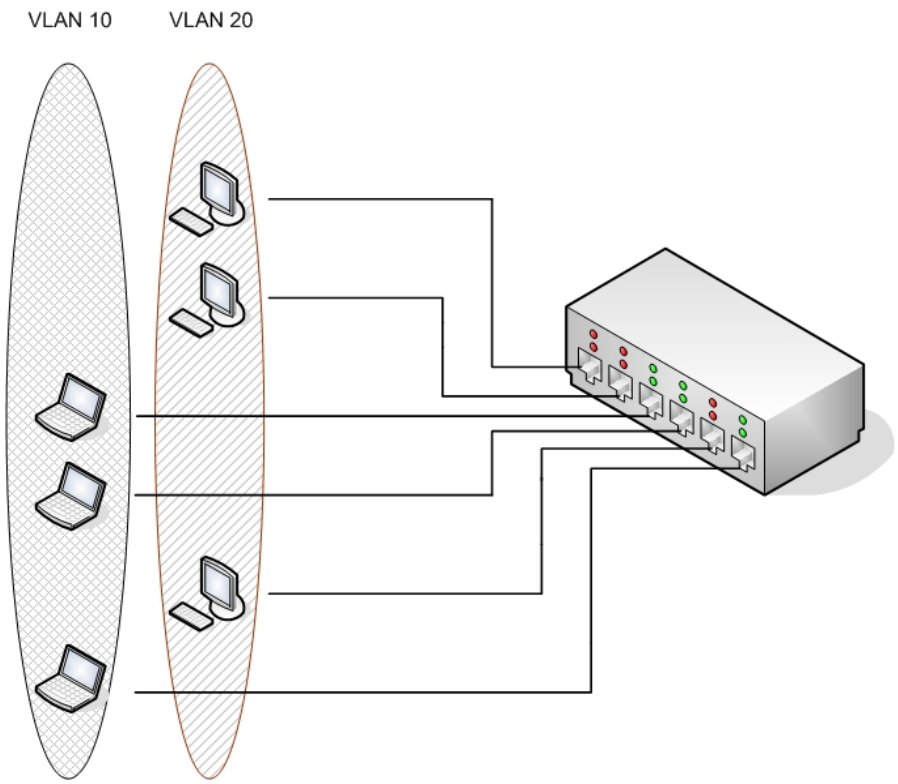
Obrázek 24: Průběh stavů Zfone

Samotný program Zfone je ještě ve stavu vývoje (beta verze), proto ještě není vše hotové. Například v logovacím souboru a v registrech operačního systému jsou zaznamenány tajné klíče [17]. Spojení občas nefunguje a je nutné restartovat aplikaci. Při sestavování spojení je patrná menší časová prodleva, která ale vzhledem k bezpečné komunikaci nepřesahuje únosnou mez. Obrázek 25 znázorňuje průběh ZRTP při krátkodobém vypnutí šifrování a následném zapnutí. Je zde vidět, že odchytnutím hovoru během zapnutého šifrování vzniká šum.

Obrázek 25: Průběh ZRTP v programu Wireshark

4.4.2 VLAN

Virtuální sítě LAN (VLAN) rozdělují síť do logických segmentů, ve kterých mohou stanice komunikovat, jako by byly na společné LAN. Dojde tedy k logickému rozdělení sítě bez ohledu na fyzické uspořádání. Jednotlivé VLANy bez použití směrování mezi VLAN nedokážou mezi sebou komunikovat a jsou bezpečně odděleny (Obrázek 26). Virtuální sítě umožňují možnost seskupování uživatelů v sítí podle skupin místo fyzického umístění, snížení zátěže broadcast zpráv (vysílány jsou jen v dané VLAN) a zvýšení bezpečnosti přenosu citlivých dat oddělením od normálního provozu.



Obrázek 26: VLAN

Nejčastější rozdělení se nachází na switchi, který pokud podporuje virtuální sítě, existuje již výchozí VLAN 1 obsahující všechny porty. Pro oddělení jednotlivých VLAN se používají tyto způsoby:

- Podle portu

  Jednotlivé porty jsou napevno připevněny k VLAN a všechny komunikace skrze tento port probíhá výhradně přes tuto VLAN. Nejčastější způsob.

- Podle MAC adresy

  Každá MAC adresa zařízení je přiřazena k jednotlivé VLAN. Při přepojení k jinému portu je přiřazena správná VLAN. MAC adresa ale není neměnná.

- Podle použitého síťového protokolu

  Zde se odděluje například provoz přes IP protokol a IPX protokol, kdy každý protokol má svoji VLAN.

- Podle autentizace

  Zařízení se musí autentizovat k serveru, který ověřuje jejich identitu, a následně je jim přiřazena povolená VLAN.

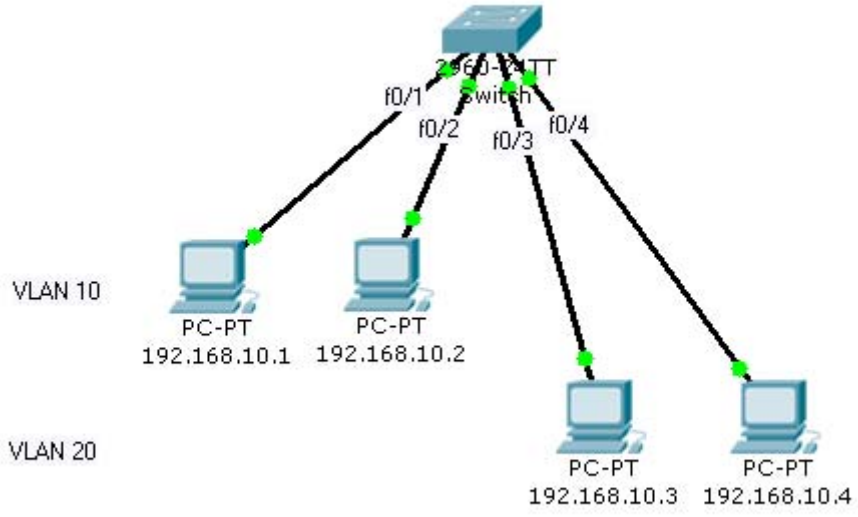
Pro použití je potřeba speciální switch, který podporuje VLAN. Využil jsem školní CISCO laboratoř, kde se nacházejí switche Cisco Catalyst 2960-24TT. K připojení ke konzoli switche jsem použil program Hyperterminal (9600 b/s, 8 datový bitů, parita žádná, 1 stop-bit a žádné řízení toku). Po klasickém úvodním nastavení (heslo, název switche a podobně) se musí nastavit v privilegovaném EXEC módu toto:

```
#configure terminal
//přepnutí do globálního konfiguračního módu
(config)#vlan 10
// vytvoření VLAN číslo 10
(config-vlan)#name data
//VLAN 10 je pojmenována jak „data“
(config-vlan)#end
//návrat do privilegovaného módu a uložení konfigurace do vlan.dat

#configure terminal
(config)#vlan 20
(config-vlan)#name voice
(config-vlan)#end

(config)#interface interface f0/1
//nastavení rozhraní fast ethernetu 0/1
(config-if)#switchport mode access
//nastavení rozhraní do přístupového módu
(config-if)#switchport access vlan 10
//zařazení rozhraní f0/1 do VLAN 10

(config-if)#interface interface f0/2
(config-if)#switchport mode access
(config-if)#switchport access vlan 10

(config-if)#interface interface f0/3
(config-if)#switchport mode access
(config-if)#switchport access vlan 20

(config-if)#interface interface f0/3
(config-if)#switchport mode access
(config-if)#switchport access vlan 20
```

Nyní jsou rozhraní f0/1 a f0/2 zařazeny do VLAN 10 a f0/3 s f0/4 jsou ve VLAN 20 (Obrázek 27). Komunikace je umožněná jen v rámci VLAN. Při pokus o ping ze stanice 192.168.10.1 (VLAN 10) na 192.168.10.3 (VLAN 20) je cíl nedostupný:

```
ping 192.168.10.3
Pinging 192.168.10.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
```

```
Request timed out.
Ping statistics for 192.168.10.3:Packets: Sent=4,Received=0,Lost=4 (100% loss)
```



Obrázek 27: Zapojení PC do VLAN

Tato vlastnost je využívána k oddělení datového provozu (například VoIP od obyčejných dat), kdy stanice připojené do jiného VLAN než jsou VoIP telefony, nemohou komunikovat a tudíž se ani případně pokoušet o útok.

4.4.3 IDS

Intrusion Detection System (IDS) je obranný systém, který detekuje podezřelé aktivity v počítačové síti. V češtině se označuji jako systémy pro detekci průniků. Tyto detektory monitorují činnost, analyzují ji a pokud danou akci vyhodnotí podle určitých pravidel jako útok, vyhlásí poplach. Základní rozdělení je na hostitelské, které jsou nasazovány přímo na serveru a monitorují činnost operačního systému, a síťové, které jsou umístěny na segmentu sítě a monitorují její chod. Síťové IDS sledují sekvenci bytů v jednotlivých paketech nebo údaje v hlavičkách paketů a porovnávají to s databází známých vzorků chování (rules) a podle nastavení provádějí další činnost (například logování).

Jedním z mnoha softwarových IDS je program SNORT, který běží jak pod Linuxem tak i pod Windows a je poskytován zdarma (dostupný na www.snort.org). Pod Windows je potřeba ještě nainstalovat WinPcap, který umožňuje zachytávání paketů. Následně se musí do složky „rules“ v adresáři „Snort“ nahrát příslušné pravidla, které budou následně aplikovány na provoz v síti.

Úpravy v snort.conf (v případě instalace do C:\Snort):

```
var RULE_PATH c:\Snort\rules
var PREPROC_RULE_PATH c:\Snort\rules
dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
include c:\snort\etc\classification.config
include c:\snort\etc\reference.config
include c:\Snort\rules\local.rules
include c:\Snort\rules\bleeding-voip.rules
```

Cesty k jednotlivým souborů, se musí upravit podle zvoleného instalačního adresář a pokud je použita Windows verze, nesmí se zapomenout na rozdíly v lomítkách („\“ a „/“). Pokud nějaké pravidlo nechceme použít (například kvůli nesmyslné chybové hlášce při spouštění), lze jej jako ostatní příkazy zakomentovat pomoci znaku „#“.

Samotné spuštění probíhá přes příkazovou řádku se syntaxí „-c“ pro umístění konfiguračního souboru, „-l“ pro nastavení losovacího adresáře, -„iX“ pro výběr X-tého síťového adaptéru (číslo zjistíme zadáním příkazu „snort -W“) a „-A console“ pro výpis upozornění do konzole.

```
snort -c c:\Snort\etc\snort.conf -l c:\snort\log -i2 -A console
```

Pro spuštění Snortu jako služba ve Windows je následující příkaz:

```
snort /SERVICE /INSTALL -c c:\Snort\etc\snort.conf -l c:\snort\log -i2 -E
```

Jednotlivé pravidla jsou uložena k adresáři „rules“ v textové podobě. Jsou rozdělena do dvou částí: hlavička a tělo.

Hlavička obsahuje informace o vykonané akci při detekci narušení (alert, log, pass, activate, dynamic, drop, reject, sdrop), použitý protokol, zdrojové a cílové IP adresy, rozsah portů a směr toku informací (například „->“ značí směr od zdroje k cíly, „<>“ oba směry).

Tělo obsahuje jednotlivé volby pravidel v kontextu „volba:hodnota“:

„content“ značí hledaný řetězec,
„msg“ zprávu uloženou do logu,
„gid“ označuje část Snortu, která generuje poplach (etc/generators),
„sid“ unikátní identifikátor konkrétního pravidlo,
„rev“ verzi pravidla,
„classtype“ je hodnota z kategorie vážnosti útoku,
„depth“ v kolika bytech na začátku paketu má Snort hledat položku “content“ a
„offset“ nastavuje začátek hledaní v bytech.

Příklad pravidla:

```
alert ip any 5060 -> any 5060 (msg:"Mnoho neutorizovanych pripojeni"; content:"SIP/2.0
401 Unauthorized"; depth:24; threshold: type both, track by_src, count 5, seconds 360;
classtype:attempted-dos; sid:2003194; rev:2;)
```

Při použití programu SIPSCAN (kapitola 4.3.1 Zjištění nastavení SIP) pro zjištění uživatelských účtů na serveru, vypíše Snort následující upozornění:

```
[**] [1:2003194:2] Mnoho neutorizovanych pripojeni [**] [Classification: Attempted
Denial of Service] [Priority: 2] {UDP} 192.168.0.1:5060 -> 192.168.0.10:4827
```

S těmito logy lze následně pracovat a díky tomu pak upravovat pravidla firewallu pro bezpečnější chod sítě.

4.4.4 TLS

K zajištění bezpečnosti signalizace SIP je možno použít stejného postupu jako u http protokolu. Všechny SIP zřízení musí podporovat jak UDP tak TCP protokol (RFC 3561, část 18. Transport). Podle tohoto doporučení je nutné posílat přes TCP větší zprávy a tato možnost musí být implementována, protože i kdyby přišla jen jediná zpráva, která by splňovala tyto podmínky, musí být poslána přes TCP protokol. Hraniční hodnota je stanovena na 1300 bytů. Za předpokladu se, že MTU (maximální přenosová jednotka) je neznáme nebo je použit Ethernet (MTU 1500 bytů). Rezerva 200 bytů je pro možný zápis cesty do hlavičky SIP zprávy. Doporučení (jak už z názvu vyplývá) ale není závazné, a proto mnoho softwarových implementací protokolu SIP nepodporují spojení přes TCP.

Bohužel ústředna Asterisk 1.4, kterou jsem zvolil jako hlavní prvek mojí sítě, nepodporuje TCP spojení pro SIP a tudíž ani TLS. Existuje sice beta verze 1.6, která už umožňuje spojení přes TCP a TLS, avšak je ještě nestabilní. I přesto jsem tuto beta verzi vyzkoušel. Spojení přes TCP fungovalo při použití softwarových SIP klientů eyeBeam, což jsou placené verze programu X-lite. Nastavení pro TCP spojení u Asterisku 1.6 v „sip.conf“:

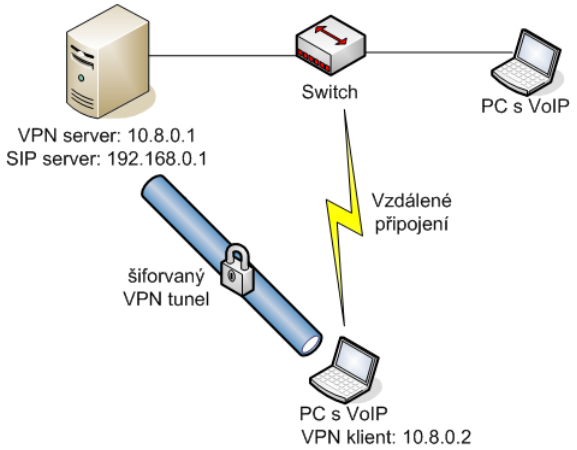
```
tcpenable=yes
tcpbindaddr=192.168.0.1:5061

[101]
transport=tcp
```

Bylo nutné nastavit jiný port (5061), než je použit pro UDP spojení (5060). Dále stačilo u nastavení klientů v „sip.conf“ přidat zvolený TCP protokol. Klienti se nastaví na odpovídající hodnoty (eyeBeam: Domain: 192.168.0.1:5061; Signalling Transport = TCP). Telefon Well nepodporuje SIP přes TCP a proto zůstal nastaven na transportní protokol UDP. Komunikace mezi všemi klienty probíhala bez problémů.

Problém nastal, pokud se nastavil TLS. Spojení funguje, jen když jsou vytvořeny certifikáty, díky kterým se může v komunikaci ověřit druhá strana. Asterisk nebyl schopen přijmout jakýkoliv certifikát a hlásil chybu. Nenalezl jsem odpověď na tento problém ani nikde na internetu, kde vzhledem k ještě probíhajícímu vývoji Asterisku 1.6, nebyly v diskusních fórech žádné odpovědi na dotazy k tomuto problému. Věřím, že až bude Asterisk ve verzi 1.6 odladěnější (první beta verze vyšla na začátku roku 2008) a s větší podporou mezi uživateli, bude komunikace přes TLS vhodná, i když většina hardwarových zařízení toto spojení ještě nepodporuje.

4.4.5 VPN

VPN alias Virtual Privat Network (virtuální privátní síť) slouží k vytvoření soukromé sítě skrze veřejnou síť, obvykle internet. Jedná se o spojení bod-bod, tedy mezi dvěmi zařízeními. K propojení je potřeba VPN server, ke kterému se připojují VPN klienti. Využívá se to nejčastěji k připojení vzdáleného pracoviště k firemní síti. Spojení funguje prostřednictvím tunelování, kdy původní data jsou zabalena do nových datagramů a ty jsou poslány prostřednictvím virtuálního spojení. Možností realizace VPN je na spojové (ATM), síťové (IPsec) a transportní (OpenVPN) vrstvě OSI modelu. Vyzkoušel jsem program OpenVPN [18], který pracuje na jediném UDP portu (1194). Díky tomu je snazší průchod přes firewall, kde stačí povolit jen jeden port a komunikace funguje. Během instalace je vytvořeno virtuální síťové rozhraní, přes které prochází komunikace ve VPN. Data jsou přenášena tunelem bez šifrování, s pomocí certifikátů nebo statickým klíčem (Obrázek 28).



Obrázek 28: VPN

Zprovoznění spočívalo v instalaci OpenVPN, kdy se ve Windows vytvořilo virtuální TAP rozhraní. Následně se musel vytvořit klíč pro šifrovanou komunikaci. Použil jsem statický klíč, který se vytvoří příkazem:

```
openvpn --genkey --secret secret.key
```

Tento klíč „secret.key“ se musí bezpečně dostat do druhého počítače, kde bude použit k šifrování. Zvolil jsem fyzické přenesení klíče. K nastavení VPN spojení se musí vytvořit 2 konfigurační soubory pro server (Linux OpenVPN: server.conf) a klienta (Windows OpenVPN: client.ovpn). Pro základní spojení nám stačí tyto hodnoty:

Server:

```
dev tun                      // typ virtuálního rozhraní
ifconfig 10.8.0.1 10.8.0.2   // IP adresa serveru a klienta ve VPN
secret static.key            // umístění sdíleného klíče
```

Klient:

```
remote 192.168.0.1           // IP adresa serveru pro vzdálené připojení
dev tun
ifconfig 10.8.0.2 10.8.0.1   // IP adresa klienta a serveru ve VPN
secret static.key
```

Vytvoření spojeni spočívá u klienta ve Windows ve zvolení nabídky „Start OpenVPN on this config file“ u klientského konfiguračního souboru a u serveru v zadání příkazu:

```
openvpn --config server.conf
```

Spojení je navázáno přes UDP pakety na portu 1194. Následně je nutné upravit konfigurační soubory u serveru Asterisk:

[sip.conf]

bindadr=0.0.0.0 // Asterisk bude naslouchat na všech rozhraních (případně nastavit přímo adresu VPN 10.8.0.1)

Následně i u VPN klienta je nutné nastavit softwarového SIP telefon ExpressTalk, aby naslouchal na rozhraní VPN (IP address: 10.8.0.2) a spojoval se na server přes VPN (SIP proxy: 10.8.0.1).

5 Závěr

V práci jsem uvedl základní mechanizmy používané ve Voice over Internet Protocol (VoIP, přenos hlasu přes IP protokol) a jejich návaznost na klasickou telefonní síť. Dále jsem popsal signalizační a transportní protokoly, které se využívají ve VoIP při přenosu multimediálních dat. Rozvedl jsem jednotlivé hlavní standardy ve VoIP. Zaměřil jsem se na signalizační protokol SIP, který je nyní nejrozšířenější a v České republice dominuje pro spojení zákazníků k VoIP poskytovateli. K převodu hlasu na digitální data se věnuje část zaměřená na zvukové kodeky. Zajištění kvality služeb je jedním z nejdůležitějších hledisek, které musí být splněno, aby se VoIP mohl zcela nahradit klasickou telefonní síť. Přesto, zajištění kvality služeb stoupá, zůstává nedůvěra k VoIP relativně veliká. Složitost zapojení a konfigurace, oproti PSTN síti značná novost, problémy s funkčností při výpadku elektrické sítě a nedostatek širokopásmového připojení k internetu jsou hlavní důvody v malém rozšíření VoIP v domácnostech.

Dále jsem u nejrozšířenějšího signalizačního protokolu SIP graficky znázornil proces sestavování a ukončení relace VoIP. Uvedl jsme jednotlivé typy zpráv, které signalizační protokol SIP používá, a popsal strukturu zpráv.

V poslední části se zmiňuji o nejpodceňovanější vlastnosti VoIP - bezpečnosti. Při vývoji protokolů se ve většině případů ze počátku nepočítalo se zabezpečením, a proto se musela ochrana implementovat později. Vzhledem k dnešní dominanci SIP protokolu nad ostatními jsem se zaměřil na útoky a možnou ochranu právě u tohoto protokolu. Principielně se základní útoky neliší od útoků na jiné síťové protokoly. SIP protokol je díky své textové orientaci velmi čitelný a lehce upravitelný, pokud se nepoužijí zabezpečovací mechanizmy. Samotný přenos hlasu je proveden přes protokol RTP, který je sice nezabezpečen, ale existují varianty (například ZRTP), které znemožňují jeho dekódování do srozumitelného zvuku.

Vytvořil jsem jednoduchou VoIP síť, ve které jsem realizoval hovory mezi účastníky. Použil jsem k tomu softwarovou ústřednu Asterisk a hardwarové a softwarové SIP telefony. Vzhledem k oddělenému přenosu hlasu a signalizace se zabezpečení zaměřuje na každou část zvlášť. Signalizace SIP je obdobou HTTP protokolu, a proto se nejčastěji k jeho zabezpečení používají techniky ze světa hypertextových dokumentu HTML. Kromě základní WWW autentizace, která už nesplňuje požadavky na autenticitu zpráv, je to hlavně rozšířená Digest autentizace, která využívá hashovacích funkcí k přenosu hesla. Tato metoda je nejrozšířenější u VoIP poskytovatelů. Metoda Secure MIME vychází z nezabezpečeného standardu MIME, který původně vznikl pro umožnění posílání souborů v mailové komunikaci.

Využívá asymetrické i symetrické kryptografie k zajištění autenticity, integrity a šifrování zprávy. I když je tento standard již delší domu na světě, jeho implementace do VoIP zařízení je slabá. Stejně jako existuje zabezpečený HTTPS k obyčejnému HTTP, vyskytuje se i obdobný protokol Secure SIP, který již nevyužívá transportní protokol UDP, ale protokol TCP, který již zajišťuje správné doručení. Použita je šifrovací vrstva TLS, která s pomocí certifikátů šifruje přenos zpráv. Nezajišťuje avšak bezpečnou komunikaci po celé trase, ale jen mezi dvěmi koncovými body. Tudíž i když se budeme snažit použít toto opatření, není zajištěno, že bude šifrován až do konce přenosu. Jeho rozšíření je větší jak u S/MIME, ale stále není zajištěna plná podpora u všech rozšířených zařízení. Využití virtuální privátní sítě VPN je obecně nejlepší možné řešení k zajištění zvláště vzdálené bezpečné komunikace, a to jak VoIP, tak i datové části komunikace. K vytvoření spojení VPN jsem využil program OpenVPN a následně simuloval vzdálené bezpečné připojení k VoIP ústředně.

Rozdělení fyzické sítě do virtuálních sítí VLAN je ve větších sítích běžná záležitost. Zvyšuje sice nároky na správu a směrování, ale možnost oddělit běžnou komunikaci například od VoIP sítě tuto nevýhodu vyvažuje. Kromě firewallu je vhodné využít i systém detekce průniků IDS, který napomáhá rozlišit podle předem daných pravidel potencionálně nebezpečné útoky a upozornit na ně, či případně i aktivně zasáhnout. Problémem jsou plané poplachy, které jsou u tohoto sytému časté. K realizaci tohoto sytému jsem využil volně dostupný program Snort. K zajištění bezpečnosti hlasových dat v paketech RTP slouží kryptografické protokoly, které vylepšují protokol RTP, a zajišťují tak utajení před útočníkem. Jejich rozšíření není veliké a například implementace protokolu ZRTP od vynálezce PGP Phila Zimmermann se stále nachází ve stavu vývoje.

Základním předpokladem k provedení cíleného útoku je zjištění parametrů a struktury vybrané oběti. Prakticky jsem vyzkoušel, jak lze zmapovat počítačovou síť a zjistit komponenty VoIP, na které se lze následně zaměřit pro získání informací o telefonním provozu. Kromě základních počítačových útoků Man in the middle a Denial of service jsem se zaměřil na odposlech VoIP hovoru.

Obecně se dá tedy říci, že úspěšně zaútočit na VoIP hovor by doopravdy dokázal i člověk s malými zkušenostmi ze světa zabezpečení počítačových sítí. Musel by však mít přistup přímo k datům, které proudí mezí účastníky hovoru. Proto je nejdůležitější se snažit zabezpečit vlastní síť a připojení k VoIP poskytovateli. Nemělo by se zapomínat na vlastní uživatele ve vnitřní síti, kteří i když mají již nějaké oprávnění, se snaží domoci větších pravomocí, a tím může dojít i k větším škodám než při útoku zvenčí. Obava o zabezpečení v síti poskytovatele internetu a jeho napojení na páteří linku je už spíše paranoidní.

Literatura

[1] Najděte klíč k VoIP zákazníkům. *Connect!*. 1.1.2007, č. květen, s. 10.

[2] WALLACE, Kevin. *VoIP bez předchozích znalostí*. 1. vyd. Brno : Computer Press, a.s., 2007. 231 s. ISBN 978-80-251-1458-2.

[3] Vozňák, Miroslav. QoS v sítích s technologií VoIP. *VoIP* [online]. 2004 [cit. 2007-12-20]. Dostupný z WWW: <http://homel.vsb.cz/~voz29/voip/Voz-TR01-2004.pdf >.

[4] SOUMAR, Michal. Signalizační protokol pro přenos hlasu přes datové sítě - SIP. *Elektrorevue* [online]. 2003 [cit. 2007-12-20]. Dostupný z WWW: <http://www.elektrorevue.cz/clanky/03003/index.html>.

[5] NOVOTNÝ, Vít. *Účastnická a koncová zařízení*. [s.l.] : [s.n.], 2002. 122 s.

[6] MOLNÁR, Karol. *Moderní síťové technologie* [online]. [2007] [cit. 2007-12-20]. Dostupný z WWW: <http://www.utko.feec.vutbr.cz/~molnar/index.php?stranka=mmos>.

[7] GRYGÁREK, Petr. Podpora multimediálních aplikací v Internetu. *Směrované a přepínané sítě* [online]. 2006 [cit. 2007-12-20]. Dostupný z WWW: <http://www.cs.vsb.cz/grygarek/SPS/lect/multimedia-ucitele.pdf>.

[8] Vozňák, Miroslav. *Signalizace SIP*. [s.l.] : [s.n.], 2006. 42 s.

[9] VANĚK, Tomáš. Bezpečnost provozu VoIP. *Teorie a praxe IP telefonie* [online]. 2006 [cit. 2007-12-20]. Dostupný z WWW: <http://www.nextsoft.cz/~malina/cs/articles/voip/clanek_Bezpecnost.pdf >.

[10] Vozňák, Miroslav. *Voice over IP* [online]. [2007-] [cit. 2007-12-20]. Dostupný z WWW: <http://homel.vsb.cz/~voz29/voip.html>.

[11] *The Zfone Project : FAQ* [online]. c2006 [cit. 2008-05-27]. Dostupný z WWW: <http://zfoneproject.com/faq.html>.

[12] *Comparison of VoIP software* [online]. 2006 , 23.5.2008 [cit. 2008-05-27]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Comparison_of_VoIP_software>.

[13] Informatik of the Universitat Bremen. *IP Telephony Cookbook : Authentication* [online]. [2006] [cit. 2008-05-27]. Dostupný z WWW: <http://www.informatik.uni-bremen.de/~prelle/terena/cookbook/main/ch04s03.html>.

[14] MIROSLAV, Ludvík. *Nové trendy v zajišťování informační bezpečnosti* [online]. 2004 [cit. 2008-05-27]. Dostupný z WWW: <http://www.systemonline.cz/clanky/nove-trendy-v-zajistovani-informacni-bezpecnosti.htm>.

[15] LYON, Gordon. *Top 100 Network Security Tools* [online]. 2006 [cit. 2008-05-27]. Dostupný z WWW: <http://sectools.org/index.html>.

[16] ENDLER, David, COLLIER, Mark. *Hacking Exposed VoIP* [online]. 2006 , 2008 [cit. 2008-05-27]. Dostupný z WWW: <http://www.hackingvoip.com/>.

[17] LIVINGSTONE, I. H. G. Forensic Analysis Zfone. *Detica* [online]. 2008 [cit. 2008-05-27]. Dostupný z WWW: <http://zfoneproject.com/docs/DFA028.17.04.D004-1.00.pdf>.

[18] *OpenVPN* [online]. c2008 [cit. 2008-05-27]. Dostupný z WWW: <http://openvpn.net/ >.

[19] Umíte vyjmenovat všechny známé odpovědi SIP?. *3CX* [online]. 2007 [cit. 2007-12-20]. Dostupný z WWW: <http://www.3cx.cz/voip-sip/sip-responses.php>.

Příloha A: Odpovědi SIP

Podrobnější výpis SIP odpovědí [19].

1xx = informační odpovědi

- 100 Trying (Pokus)
- 180 Ringing (Vyzvánění)
- 181 Call Is Being Forwarded (Hovor je přesměrován)
- 182 Queued (Fronta)
- 183 Session Progress (Postup relace)

2xx = odpovědi o dosažení úspěchu

- 200 OK
- 202 Accepted: Used for referrals (přijat: Použití pro doporučení)

3xx = odpovědi o přesměrování

- 300 Multiple Choices (Více voleb)
- 301 Moved Permanently (Trvale přemístěn)
- 302 Moved Temporarily (Dočasně přemístěn)
- 305 Use Proxy (Použití proxy)
- 380 Alternative Service (Alternativní služba)

4xx = selhání příkazu

- 400 Bad Request (Nesprávný příkaz)
- 401 Unauthorized: Used only by registrars. Proxys should use proxy authorization 407 (Neautorizovaný: Používají pouze registrátoři. Při využití proxy by se měla použít autorizace proxy 407)
- 402 Payment Required (Reserved for future use) (Požadována platba (rezervováno pro budoucí použití))
- 403 Forbidden (Zakázáno)
- 404 Not Found: User not found (Není nalezen: Uživatel nebyl nalezen)
- 405 Method Not Allowed (Metoda není povolena)
- 406 Not Acceptable (Není přijatelné)
- 407 Proxy Authentication Required (Požadovaná autentizace proxy)
- 408 Request Timeout: Couldn't find the user in time (Časový limit příkazu: Uživatele se nepodařilo najít včas)

- 410 Gone: The user existed once, but is not available here any more. (Je pryč: Uživatel kdysi existoval, ale již není k dispozici.)
- 413 Request Entity Too Large (Entita příkazu je příliš velká)
- 414 Request-URI Too Long (URI příkazu je příliš dlouhý)
- 415 Unsupported Media Type (Typ média není podporován)
- 416 Unsupported URI Scheme (Plán URI není podporován)
- 420 Bad Extension: Bad SIP Protocol Extension used, not understood by the server (Nesprávná přípona: Je použita nesprávná přípona protokolu SIP, pro server není srozumitelná)
- 421 Extension Required (Je požadována přípona)
- 423 Interval Too Brief (Interval je příliš krátký)
- 480 Temporarily Unavailable (Dočasně není k dispozici)
- 481 Call/Transaction Does Not Exist (Hovor/transakce neexistuje)
- 482 Loop Detected (Detekována smyčka)
- 483 Too Many Hops (Příliš mnoho skoků)
- 484 Address Incomplete (Adresa není úplná)
- 485 Ambiguous (Nejasný)
- 486 Busy Here (Zaneprázdněný)
- 487 Request Terminated (Příkaz ukončen)
- 488 Not Acceptable Here (Nepřijatelný)
- 491 Request Pending (Příkaz čeká)
- 493 Undecipherable: Could not decrypt S/MIME body part (Nerozluštitelný: Nepodařilo se rozluštit část těla S/MIME)

5xx = chyby serveru

- 500 Server Internal Error (Vnitřní chyba serveru)
- 501 Not Implemented: The SIP request method is not implemented here (Neimplementováno: Příkaz/metoda SIP zde není implementována)
- 502 Bad Gateway (Nesprávná brána)
- 503 Service Unavailable (Služba není k dispozici)
- 504 Server Time-out (Časová limit serveru)
- 505 Version Not Supported: The server does not support this version of the SIP protocol (Verze není podporována: Server nepodporuje tuto verzi protokolu SIP)
- 513 Message Too Large (Zpráva je příliš velká)

6xx = globální selhání

- 600 Busy Everywhere (Vše je zaneprázdněno)
- 603 Decline (Odmítnutí)
- 604 Does Not Exist Anywhere (Nikde neexistuje)
- 606 Not Acceptable (Nepřijatelný)