VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

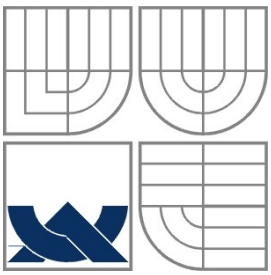
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJ PRO TESTOVÁNÍ BEZPEČNOSTI VOIP

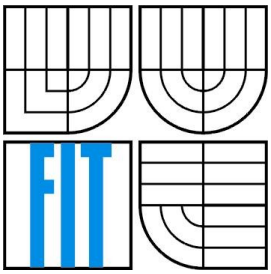
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE Pavel Mazálek
AUTHOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJ PRO TESTOVÁNÍ BEZPEČNOSTI VOIP

A TOOL FOR TESTING VOIP SECURITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Pavel Mazálek
AUTHOR

VEDOUCÍ PRÁCE — Ing. Petr Matoušek, Ph.D.
SUPERVISOR

BRNO 2012

Abstrakt

Tato práce je zaměřena na problematiku zabezpečení protokolu SIP. Zabývá se popisem základních principů signalizačního protokolu a nástinem protokolů potřebných pro přenos hlasových dat. Popisuje některé základní útoky na zařízení pracující s protokolem SIP a uvádí i příklady použití. Poslední kapitola je věnována popisu mnou vytvořeného programu, který má sloužit pro penetrační testování zařízení.

Abstract

This work is focused on the security of SIP. It describes basic principles of signaling protocol and it outlines the necessary protocols for voice data transfer. It describes some basic attacks on device working with SIP and provides examples and applications. The last chapter is devoted to descrie the program can be used for penetration testing of devices.

Klíčová slova

VoIP, bezpečnost, SIP, útoky, přenos hlasu, telefonie

Keywords

VoIP, security, SIP, attacks, transmission of voice, telephony

Citace

Pavel Mazálek: Nástroj pro testování bezpečnosti VoIP, bakalářská práce, Brno, FIT VUT v Brně, 2012

Nástroj pro testování bezpečnosti VoIP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Mazálek
4.5.2012

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Petru Matouškovi, Ph.D. za věnovaný čas a odbornou pomoc při řešení práce. Dále bych rád poděkoval kpt. Ing. Antonínu Mazálkovi, Ph.D. za pomoc při vymýšlení tématu práce a konzultace v průběhu její tvorby.

Obsah

1 Úvod

Hlasové služby jsou pro dnešní společnost věcí zcela samozřejmou a nepostradatelnou. Představují zdroj komunikace, zábavy, informací, ale samotné koncové terminály – stále častěji mobilní - je možné využít i k dalším aplikacím, jako je dálkové ovládání zařízení pouhým prozvoněním nebo zasláním zprávy či ovládání bankovního účtu. Tyto možnosti jsou dány technologickou vyspělostí dnešních systémů. Od doby Bellova vynálezu telefonu (1876) prošly hlasové služby řadou změn, z nichž změny v posledních desetiletích jsou nejzásadnější, umožňují mobilitu uživatelů a využívání datových sítí pro přenos hlasu, ačkoliv ještě před několika desítkami let tomu bylo zcela naopak, data byla přenášena přes sítě telefonní.

Významným mezníkem byl prudký rozvoj internetu. Dnes je dostupný téměř všude s poměrně nízkými náklady. Uživatel často platí pouze paušální poplatky za připojení, aniž by byl významně omezen přenosem dat. Přímo se tak nabízí myšlenka využití datové sítě Internet pro přenos hlasu. Technologie, která dokáže hlas pomoci IP protokolu přenášet, se nazývá VoIP (Voice over Internet Protocol) [1]. Protože původně IP protokol nebyl navržen pro přenos hlasu, ale na co nejefektivnější přenos dat (princip best effort), přinesly počátky nasazování VoIP značné problémy. Mezi nejpodstatnější patřilo vyřešit otázku zabezpečení co nejmenšího zpoždění hlasových paketů v datové síti, rozptyl zpoždění (jitter) a problematiku napájení koncových zařízení. Přes veškerá technická opatření je stále poměrně obtížné zabezpečit stejnou spolehlivost hlasových služeb pomoci VoIP, jakou dokáží zabezpečit klasické telefonní ústředny (spolehlivost 99,999%).

Ačkoliv VoIP služby svoji kvalitou nepředčí telefonní operátory, jejich popularita a počet uživatelů stále narůstá. Hlavním důvodem jsou výrazné finanční úspory. Volání v rámci operátora (případně pomoci IP adres) může představovat neplacenou službu a to bez ohledu na lokalizaci uživatelů. VoIP je možné využít stejným způsobem (bezplatně) také pro videotelefonii a telefonní konference. Kromě toho VoIP přináší nové služby. Umožňuje například současné vyzvánění telefonů (pracovní a soukromý), nebo využití doplňkových služeb, jako je přenos XML souborů (centrálně spravovaný adresář kontaktů, informace o počasí, firemní informace,..). V mnoha případech předběhly možnosti VoIP legislativní rámce. Například není zcela jednoznačně dořešena otázka mobility VoIP čísel (zda je možné nomadická čísla používat v zahraničí).

Dnešní koncová zařízení pro VoIP telefonii nabízejí uživateli stále více možností. Tím jsou také ale složitější a oproti klasickým analogovým telefonům zranitelnější. Obsahují vlastní procesor, firmvare, komunikují řadou síťových protokolů. Je tedy na místě ptát se, jakým způsobem jsou zabezpečeny proti hrozbám a útokům. Existují rizika při využívání této technologie? Že se nejedná o možné budoucí problémy, ale o aktuální problematiku, dokládají příklady z médií. Dobře známé jsou například případy zneužití VoIP ústředen[1] pro vedení cizího provozu na účet majitele ústředny, což má za následek nemalé finanční ztráty. Jedná se však pouze o jeden z nejznámějších z celé řády útoků a podvodů, které lze na VoIP ústředny (koncové uživatele) uskutečnit.

Jak je z výše uvedeného patrné, problematika bezpečností VoIP technologií je aktuální a dynamickou problematikou kterou, se chci zabývat ve své bakalářské práci. Vzhledem ke značnému rozsahu problematiky se zaměřím zejména na možnosti softwarového testování bezpečnosti VoIP.

1 http://www.lam.cz/index.php/testy-zarizeni/77

2 Protokol SIP

Protokol SIP (Session Initiation Protocol) [2] je signalizační protokol určený především pro hlasovou komunikaci přes internet. Byl vytvořen jako alternativa k protokolu H.323[2]. Použití protokolu SIP není omezeno pouze na internetovou telefonii, jeho využití je mnohem širší. Své uplatnění postupně našel také při videokonferenčních úlohách nebo v aplikacích instant messagingu. Postupem času sílila jeho obliba a v dnešní době je pravděpodobně nejpoužívanějším protokolem nejen pro vnitřní komunikaci v malých i velkých firmách ale i jako levnější řešení pevné telefonní linky. S rozšiřováním nasazení protokolu roste i riziko napadení a zneužití, proto je potřeba věnovat se bezpečnosti a snažit se předcházet zneužití.

Hlavní výhodou protokolu SIP je jeho jednoduchost. Oproti protokolu H.323, který je bitově orientovaný, představuje SIP textově orientovaný protokol umožňující snadnou analýzu. Při jeho návrhu se vycházelo z principů protokolu HTTP , který byl odborné veřejnosti dobře znám. V kombinaci s textovou orientací vznikl jednoduchý, principiálně známý a snadno implementovatelný protokol. Další jeho výhodou byla otevřenost standardu. Díky těmto přednostem brzy dosáhl značné popularity, stal se dominantním signalizačním protokolem v oblasti VoIP aplikací a stále více je využíván i v úlohách instant messagingu. Jako příklad uvádím jeho využití v mobilních sítích UMTS.

Mezi základní úlohy protokolu SIP patří navázání, modifikace a ukončení spojení s jedním nebo s více účastníky. Z pohledu modelu ISO-OSI představuje aplikační protokol. Otevřenost standardu umožňuje vytváření nových, sofistikovanějších, uživatelsky zaměřených aplikací.

Avšak i možnosti protokolu SIP jsou omezené. Například není schopen provádět managemant aktivních relací po jejich navázání. Dále není navržen pro správu zabezpečení kvality spojení, neumožňuje koncovým účastníkům volbu kodeků. Tyto úkoly řeší ve spolupráci s protokolem SDP.

2 http://www.itu.int/rec/T-REC-H.323/

2.1 Architektura SIP

První verze protokolu SIP byla standardizována v roce 1999 IETF a popsána v normě RFC 2543[3]. V roce 2002 byla vydána norma RFC 3261 [2], která opravuje nalezené chyby a snaží se některé části lépe popsat. Nová norma nebyla vydána jako dodatek, ale jako samostatná norma a je takřka dvakrát delší než norma původní. V současné době se používá SIP verze 2. Dále se budu zabývat pouze touto verzí protokolu.

Architektura SIP definuje dva základní druhy zařízení (obdoba HTTP spojení server-client):

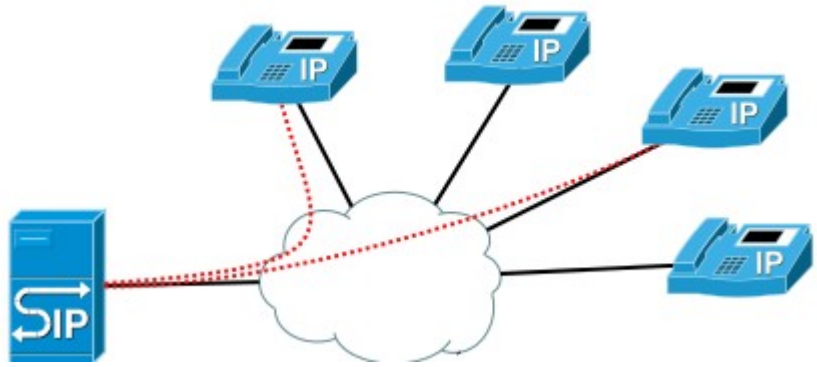
- UAC (User Agent Client) - představuje koncové zařízení, které používá klient. Může jít o počítačovou aplikaci nebo fyzický telefon položený doma na poličce. Velkou nevýhodou oproti běžné pevné telefonní lince je nefunkčnost v případě výpadku elektrického proudu. IP telefon je plně závislý na připojení k internetu a vlastním napájení.
- UAS (User Agent Server) - UAS představuje obecně server obsluhující provoz SIP sítě. Jedná se především o servery, ke kterým se přihlašují klientská zařízení, aby získala potřebné informace pro přihlášení, informace o umístění dalších klientů potřebné pro vytvoření hovoru, informace o dalších serverech atd. Přes tyto servery prochází signalizační pakety SIP, nikoliv samotný hlasový provoz. Ten probíhá vlastní cestou přímo mezi klientskými zařízeními.

Pomoci dvou základních prvků (UAC, UAS) je možné vytvořit libovolně velkou VoIP síť. V praxi zpravidla považujeme komunikaci s dalšími sítěmi, k čemuž slouží speciální prvky tzv. brány (gateway). Brány volíme podle typu sítě, se kterou chceme být propojeni. Typickým příkladem jsou brány pro připojení do klasických telefonních sítí PSTN (Public Switch Telephone Networks), GSM sítě, UMTS sítě, ISDN sítě, ale také do sítě se signalizačním protokolem H.323 či SCCP[4] atd. Hlavním úkolem brány je zajistit konverzi hlasového toku do příslušného formátu, přičemž v řadě případů je potřeba konvertovat také signalizační zprávy. Speciálním případem může být pouhá konverze hlasového toku za účelem snížení nároků na přenosové pásmo v síti WAN, nebo naopak pouhá konverze signalizačního protokolu při zachování datového toku při přechodu ze SIP sítě do sítě H.323.

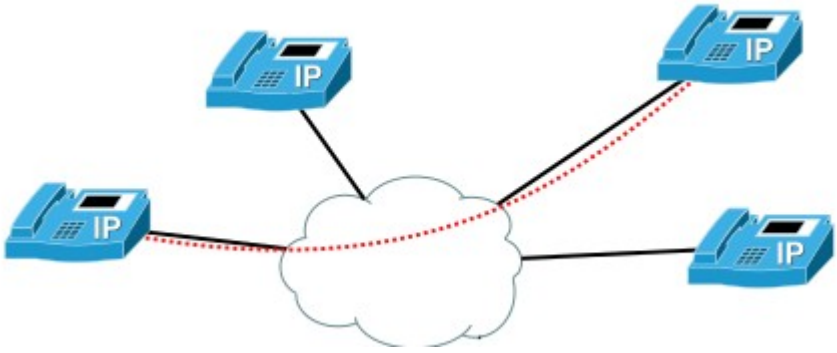
3 http://www.ietf.org/rfc/rfc2543.txt

4 Skinny Call Control Protocol - protokol pro VoIP vytvořený firmou Cisco

V reálné síti jsou zpravidla používána další standardní síťová zařízení (servery, routery) pro zabezpečení služeb typu DHCP, DNS atd. Architektura SIP nám umožňuje vytvářet sítě s decentralizovaným řízením, častějším případem však bývá síť s centralizovaným řízením. Koncový SIPový terminál totiž může vystupovat jak v roli UAC, tak v roli UAS. To nám umožňuje sestavit přímé spojení mezi koncovými terminály (případ decentralizovaného řízení). Sítě s centralizovaným řízením obsahují centrální prvek – server (servery), který se stará o vyhledání lokace koncového uživatele ve své databázi a bez jehož přítomnosti není možné spojení navázat. Tuto variantu používají poskytovatelé VoIP a firmy. Řada koncových zařízení podporuje obě metody - být zaregistrován u poskytovatele VoIP a zároveň umožňovat navázání spojení pomoci IP adresy. Na obrázku 1 je zobrazena architektura sítě využívající signalizační protokol SIP s centralizovanou topologií, na obrázku 2 s decentralizovanou topologií.



Obr. 1 - Síť s centralizovaným řízením



Obr. 2 - Síť s decentralizovaným řízením

Protokol SIP verze 2 rozeznává celkem čtyři druhy serverů [2]:

- **Proxy server** – tento server zastává dvě role - roli serveru, a za účelem vytváření požadavků roli klienta. Hlavní úlohou proxy serveru je směrovat a přeposílat zprávy SIP na další servery, které jsou blíže k cílovému zařízení. Proxy servery jsou velmi vhodná zařízení pro řízení bezpečnostní politiky. Například zde mohou být nastavena pravidla na filtrování hovorů z konkrétních čísel. Proxy server zprávy interpretuje, a je-li nezbytné, přepisuje její určité části ještě předtím, než zprávu předá dále.
- **Redirect server** – je to uživatelský agent, který na obdržené požadavky odpovídá zprávou typu 3xx. Spolu s touto odpovědí zasílá i nová (alternativní) URI, kde lze účastníka nalézt.
- **Location server** – funkcí tohoto serveru, kterou většinou obstarává proxy server nebo redirection server, je získat informaci, kde je možné hledaného účastníka kontaktovat. Tento server obsahuje seznam žádné, jedné nebo více adres účastníka. Vytváření a mazání těchto adres na serveru se děje zasíláním zpráv s požadavkem REQUEST.
- **Registrar server** – přijímá požadavky REQUEST a informace obsažené v požadavku předá location serveru.

V praxi je zpravidla jedno zařízení schopno zabezpečit všechny funkce.

2.2 Struktura zpráv SIP

SIP protokol pracuje na základě výměny textových zpráv. Tyto zprávy mají určitou strukturu. Zpráva protokolu SIP může obsahovat pouze textové znaky povolené normou UTF-8 (RFC 2279[5]) , nemůže tedy obsahovat znaky národní. Zprávy SIP přenášejí buď žádosti klienta nebo odpovědi serveru. Všechny zprávy, ať žádosti nebo odpovědi, mají povinně úvodní řádku, jedno nebo více polí hlavičky a volný řádek označující konec hlavičky. Volitelnou částí je tělo zprávy. Úvodní řádek, každý řádek hlavičky i řádek prázdný, musí být ukončen znakem CRLF (carriage-return line-feed sequence). Zpráva musí

5 http://www.ietf.org/rfc/rfc2279.txt

obsahovat prázdný řádek, i když neobsahuje tělo zprávy. Svoji strukturou je zpráva SIP obdoba zprávy HTTP protokolu.

Protokol SIP nedefinuje strukturu a obsah volitelné části zprávy (těla zprávy). Nejčastěji je tato část využívána pro přenos podporovaných kodeků při sestavování spojení. Tyto informace však vyžadují přesnou strukturu k čemuž je využíván protokol SDP. Zpráva SIP tak ve svém těle přenáší jiný protokol. Strukturu zpráv SIP objasňuje obrázek 3.

start-line (Request-Line / Status-Line)	`INVITE sip:bob@biloxi.com SIP/2.0`
message-header	`Via: SIP/2.0/UDP pc33.atlanta.com`
	`To: Bob <bob@biloxi.com>`
	`From: Alice <alice@atlanta.com>`
	`Call-ID: a84b4c76e66710`
	`CSeq: 314159 INVITE`
	`Max-Forwards: 70`
	`Date: Thu, 21 Feb 2002 13:02:03 GMT`
	`Contact: <sip:alice@pc33.atlanta.com>`
	`Content-Type: application/sdp`
	`Content-Length: 147`
	CRLF
[message-body]	`v=0`
	`o=UserA 2890844526 2890844526 IN IP4 here.com`
	`s=Session SDP`
	`c=IN IP4 pc33.atlanta.com`
	`t=0 0`
	`m=audio 49172 RTP/AVP 0`
	`a=rtpmap:0 PCMU/8000`

Obr. 3 – Struktura zprávy SIP obsahující ve svém těle protokol SDP [1]

Každý řádek musí být zakončen znakem CRLF, v textovém editoru je tento znak znázorněn pouze pro zdůraznění prázdného řádku.

- **Start-line** – známé spíše pod názvem URI (Uniform Resource Identifier). Toto URI je ve zprávách SIP definováno složením několika položek, konkretně sip:user:password@host:port;uri-parameters?headers. Sip říká, že se jedná o SIP URI. Následuje uživatelské jméno odesílatele, volitelně jeho přihlašovací jméno. Samotná norma však velice důrazně nedoporučuje zadávat heslo na tomto místě, neboť je velice snadno zjistitelné pouhým odposlechnutím síťového provozu. Za znakem @ je

uvedena adresa ústředny (tato adresa může být uvedena jako doménová adresa a nebo číselná hodnota adresy IPv4 nebo IPv6) a za dvojtečkou číslo portu (standardně 5060 pro TCP, UDP a SCTP), na který se požadavek zasílá. Parametry URI nastavují vlastnosti pro transportní vrstvu. Můžeme zde definovat TTL (Time To Live). Zadávají se ve tvaru parametr=hodnota a oddělují se středníkem.

Hlavička zprávy SIP obsahuje dva typy polí, povinná a volitelná. Povinná pole předávají základní informace nutné pro pro zajištění správného chodu sítě. Jedná se o pole To, From, Cseq, Call_ID- Max-Forwards a Via. Naproti tomu máme volitelné řády, které předávají dodatečné informace. Mezi nimi nalezneme například datum, kdy byla zpráva vytvořena, název zařízení atd.

- **Via** – pokud UAC vytváří požadavek, musí být toto pole umístěno v hlavičce zprávy. Indikuje transportní užití a identifikuje, kam má být požadavek odeslán. Tedy uvádí verzi protokolu SIP, transportní protokol (TCP,UDP SCTP), doménové jméno nebo číselnou adresu Ipv4 nebo Ipv6, cílový port (pro SIP standardně 5060) a nakonec branch. Branch začíná vždy „magickou“ konstantou "z9hG4bK“ a dále pokračuje náhodnými znaky. Tento branch musí být jedinečný napříč prostorem a časem pro všechny požadavky posílané UA (User Agentem).

- **To** - tento řádek obsahuje jméno uživatele, který zprávu odeslal, ve tvaru SIP URI (možné zadat i tel URL, viz RFC 2806[6]), které je uzavřeno v hranatých závorkách. Před tímto URI může být zadáno zobrazované jméno (např. na display zařízení).
  Příkladem může být To: Carol <sip:carol@chicago.com>.

- **From** - řádek, který dovoluje (nepovinně) identifikaci logickým jménem. Toto logické jméno se zobrazí příjemci. Logickým jménem se rozumí identifikační jméno volajícího, ne IP adresa volajícího. Může sloužit například k filtraci nebo přesměrování hovorů od různých volajících. Příkladem může být: From: "Bob" <sips:bob@biloxi.com>

6 http://www.ietf.org/rfc/rfc2806.txt

- **Call-ID** – tento údaj je složen z dlouhého řetězce náhodně generovaných znaků (na konec se někdy přidává jméno uživatele, IP adresa), který musí být unikátní a slouží jako jednoznačná identifikace dialogu se serverem.

- **Cseq** – v požadavcích obsahuje jedno decimální číslo a název požadavku. Číslo slouží pro určení pořadí zaslaných zpráv v rámci dialogu.

- **Max-Forward** – určuje maximální počet serverů, přes které může zpráva projít.

- **Contact** – toto pole má stejnou strukturu jako To a From, tedy zobrazované jméno a poté SIP URI. Pokud server obdrží registrační zprávu, vezme URI z tohoto pole a uloží si jej jako kontaktní adresu.

- **Content-Length –** obsahuje číslo udávající velikost těla zprávy. Číslo musí být rovno a nebo větší nule. Pokud zpráva neobsahuje tělo, je nastaveno na nulu.

- **Authorization** – toto pole předává serveru autorizační údaje klienta v takové podobě, aby se zabránilo odposlechu hesla uživatele. Následuje příklad autorizačního pole.

  ```
  Authorization: Digest username="Alice",
  realm="atlanta.com",nonce="84a4cc6f3082121f32b42a2187831a9e",
  response="7587245234b3434cc3412213e5f113a5432"
  ```

  Klient dostane od serveru zprávu, která obsahuje položky realm (jméno serveru) a nonce (náhodně vygenerované číslo serverem). Klient po přijetí těchto údajů vytvoří položku response. Ta v sobě určitým mechanizmem spojí uživatelské jméno, heslo, položku realm a nonce a zašifruje algoritmem MD5. Server provede stejnou operaci a pokud se response v přijaté zprávě shoduje s response, které si vypočítal server, autorizuje uživatele.

2.3 Komunikace metodou žádosti a odpovědi

Jak již bylo řečeno dříve, protokol SIP komunikuje na principu klient-server. Jedna strana vysílá žádosti, druhá strana ji zasílá odpovědi. V RFC 3261 [1] je definováno celkem šest základních typů žádostí, které jsou nazývány metodami. Typ použité metody je

specifikován v úvodním řádku zprávy SIP. Těchto šest základních metod je možné rozdělit podle účelu do čtyř skupin a to metody pro registraci (REGISTER), navázání spojení (INVITE, ACK, CANCEL), pro ukončení spojení (BYE) a zjišťování rozšířených možností (OPTIONS). Jednotlivé metody dále stručně popíši a na jednoduchém příkladu ukáži možnost využití.

- **REGISTER** – je metoda zajišťující přihlášení uživatele na SIP server. Pokud server nepožaduje autorizaci, tak po přijetí této metody vezme na vědomí, že je uživatel připraven přijímat hovory od ostatních účastníků. V druhém případě, kdy server požaduje autorizaci, vždy odpoví chybovým stavovým kódem 407 a zašle potřebné údaje pro přihlášení. Klient poté pošle další metodu REGISTER s autorizačním polem na server. Tento krok je nutné v určitých intervalech opakovat. Tento interval není specifikován žádnou normou a je na uživatelích/administrátorech, jak dlouhý jej zvolí. Na straně serveru se odpočítává čas, a pokud nedojde do uplynutí intervalu nový požadavek REGISTER, uživatel je automaticky od serveru odhlášen. Metoda REGISTER neslouží pouze k přihlašování, ale i k odhlašování od serveru a to tak, že pole Content-Length má hodnotu nula. Tím se odhlásí od serveru uživatel, který zprávu zaslal. Pokud máme zaregistrováno více zařízení a chceme je všechna zrušit, vložíme do pole Contact znak „*“.

- **INVITE** – pomocí této zprávy navazujeme spojení. Po jejím obdržení zašle druhá strana v případě úspěšného navázání odpověď 200 (OK) a telefon začne vyzvánět. Při každém navazování spojení se v těle zprávy zasílá protokol SDP, díky kterému se zařízení dohodnou například na použitém hlasovém kodeku.

- **ACK** – se posílá jako potvrzení žádosti. Zpráva je totožná s přijatou, pouze název metody je přejmenován na ACK. To proto, aby druhá strana bezpečně poznala, na kterou žádost klient odpovídá.

- **BYE** – ukončovací zpráva spojení při hovoru, například při položení sluchátka.

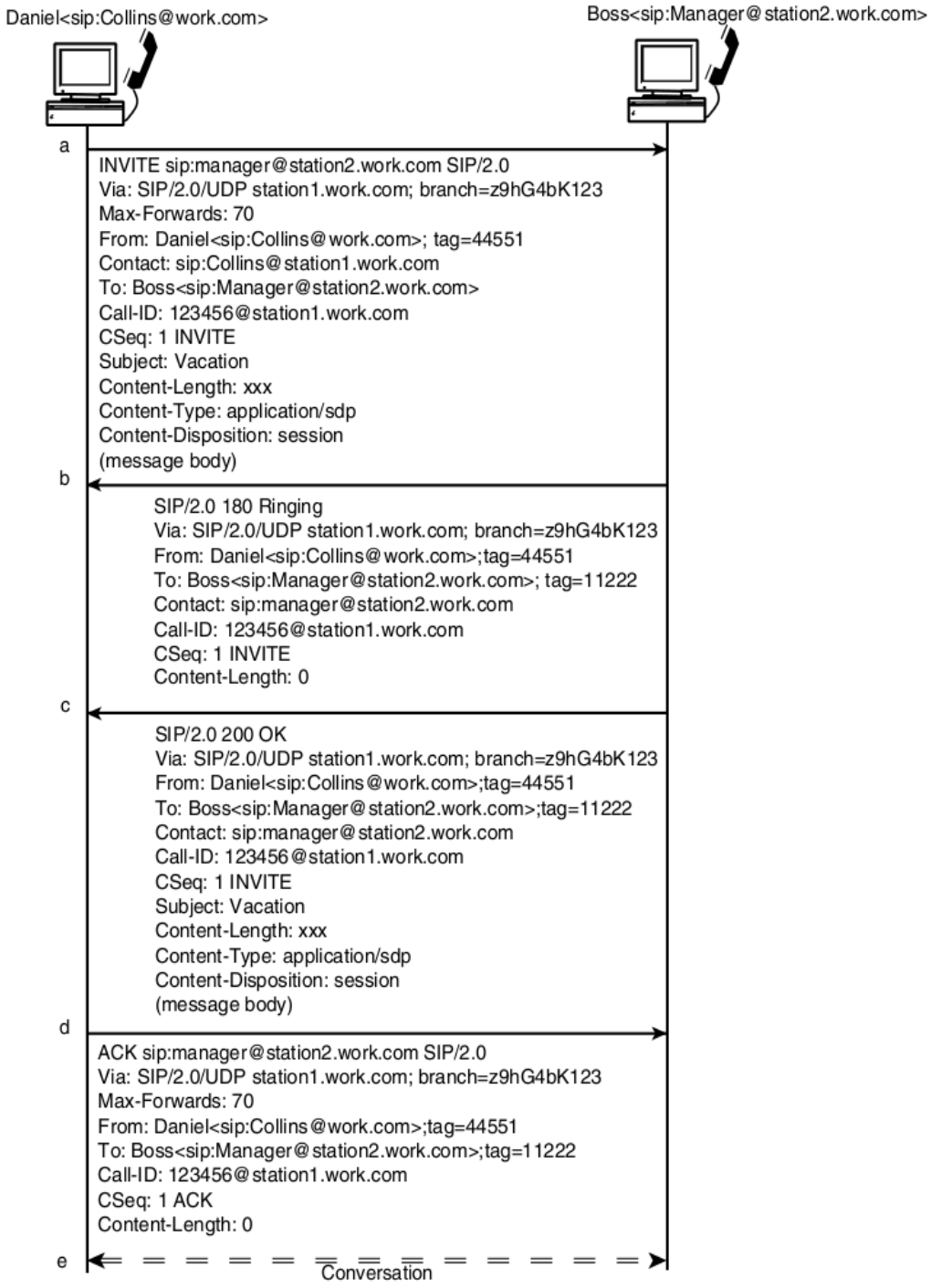
- **CANCEL** – slouží ke zrušení prováděné akce ještě před jejím dokončením (například zrušení INVITE, ještě než začne telefon zvonit).

- **OPTIONS** – dotaz na možnosti serveru

Příjemce SIPové zprávy po obdržení žádosti odesílá odpověď. Stejně jako počet metod žádostí je omezen i počet možných odpovědí a to na šest tříd. Třída odpovědi je opět uvedena v úvodním řádku SIPové zprávy a to formou třímístného číselného stavového kódu. Tento kód nemusí být pro člověka okamžitě srozumitelný, proto je uveden i krátký textový popis stavového kódu (viz následující příklady). Z číselného kódu je nejvýznamnější první číslice, která může nabývat číselné hodnoty 1- 6 a určuje třídu odpovědi. Následující dvě číslice již dále nedělí třídu do podskupin. V následujícím textu bude vysvětlen význam stavových kódů uvedených tříd. Dále budou následovat příklady.

1xx – informace byla přijata ke zpracování, ale výsledek bude znám později (ověření uživatele)
2xx – odpověď typu OK, vyjadřuje, že vše proběhlo v pořádku (uživatel registrován)
3xx – značí, že zaslaná zpráva byla přesměrována na novou adresu uživatele.
4xx - chyba klienta – špatný požadavek (špatná syntaxe zprávy, heslo, uživatelské jméno,...)
5xx – chyba na serveru, syntax zprávy správná
6xx – dotaz nemůže být proveden na žádném serveru

Jak probíhá zasílání žádostí a odpovědí je možné vidět na obrázku 4, který názorně ukazuje, jak probíhá inicializace hovoru nejen diagramem, ale i zprávami v textové podobě.



Obr. 4 – Zasílání žádostí a odpovědí [2]

3 Přenos hlasu a obrazu ve VoIP

V předchozí kapitole byl popsán signalizační protokol SIP, jehož úkolem je navázání, řízení a ukončení komunikace. Pro přenos hlasových a obrazových dat, ve spolupráci se signalizačním protokolem SIP, je používána dvojice protokolů přímo určených k přenosu těchto dat v reálném čase. Zajišťují také řízení datového toku mezi komunikujícími zařízeními. Komunikace probíhá přímo mezi koncovými zařízeními. Tyto protokoly jsou využívány ve více aplikacích, které potřebují přenášet malé množství dat v reálném čase.

3.1 Protokoly RTP a RTCP

Protokol RTP (Real-time Transport Protocol) [3] zajišťuje doručení dat v reálném čase. Používá se pro přenos zvuku a obrazu pro jednosměrnou, obousměrnou i skupinovou komunikaci. Protože při přenosu audiovizuálních dat je hlavním požadavkem doručit data v reálném čase, jako transportní mechanizmus se využívá protokol UDP. Dá se říci, že RTP je nadstavbou UDP tak, aby vyhovoval doručování v reálném čase. Kromě vlastních dat (tzv. užitečného zatížení) protokol RTP definuje a přenáší pořadová čísla paketů (slouží pro identifikaci ztráty nebo duplicity paketů), typ obsahu (určuje použitý zvukový nebo obrazový kodek, který se může během trvání komunikace měnit), indikaci začátku a konce rámce, identifikaci zdroje a údaje o synchronizaci. Synchronizace je důležitá pro sestavení plynulého toku na druhé straně. Přenáší informace například o zpoždění přenosu dat nebo jeho kolísání. V rámci komunikace se díky těmto informací mění rychlost odesílání RTP paketů, jejich velikost nebo použitý kodek a výsledný tok na druhé straně je tak co nejvíce plynulý.

Protože protokol RTP používá pouze čísla paketů a časové značky, ale již nezajišťuje kontrolu včasného doručení paketů nebo jejich doručení ve správném pořadí, spolupracuje s protokolem RTCP (Real-time Transfer Control Protocol). Oba dva protokoly jsou definovány ve společné normě RFC 3550. RTPC pakety jsou vyslány vždy jednou za určitý čas (v řádech sekund) a to vždy všem účastníkům RTP relace. RTPC tok tvoří zhruba 5 % celé komunikace, zbylých 95 % tvoří tok RTP paketů, které jsou vysílány v řádech milisekund. Úkolem RTCP je využívat data přenesená pomoci RTP, vyhodnocovat zpoždění, ztrátu paketů, zahlcení spojení a na základě těchto informací řídit na jedné straně odchozí datový tok a na straně druhé jej správně sestavit.

4 Útoky na VoIP

Bezpečnostní rizika vždy rostou s popularitou nové technologie a zvláště u technologií volně dostupných a používaných často neznalými uživateli je potřeba klást důraz na bezpečnost hned od začátku jejich masového nasazení. Nejinak je tomu i při vzniku sítí, pracujících s protokolem SIP. Hrozbou může být například vyřazení služby z provozu a znemožnění hovorů jednotlivci i velké skupině uživatelů [4]. Vážnějším porušením bezpečnosti je zneužití zařízení (identity uživatele) a následné hovory účtované na napadený účet. Další neméně závažnou hrozbou je odchytávání hlasových paketů, z kterých je možno zrekonstruovat hovor.

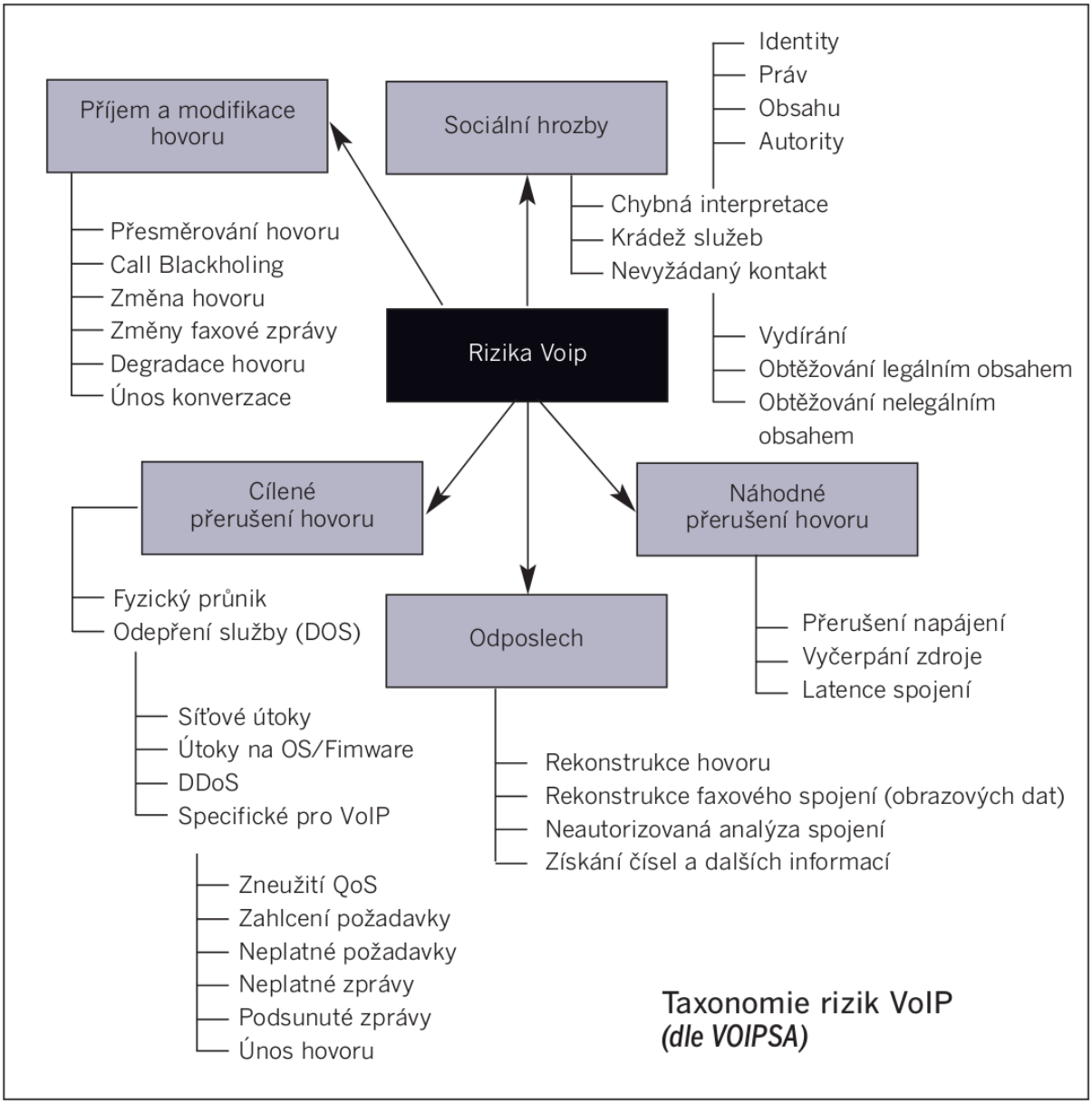
Při většině útoku je nutné vniknout do sítě a získat přístup k přenosovému médiu, což je pro útočníka určité riziko, neboť pokud je síť správně zabezpečena a monitorována, šance, že se mu povede připojit do sítě, je velice malá. Naopak hrozba, že bude odhalen a potrestán je velká. Vhledem k tomu, že velice záleží na znalostech a možnostech administrátora sítě, mohou nastat i krajní případy, kdy je síť zabezpečena špatně, nebo vůbec. Například pokud administrátor nepřenastaví heslo a uživatelské jméno nastavené při výrobě zařízení (admin/admin, root/root,...), lze velice snadno změnit nastavení zařízení. Ano, i takovéto případy mohou nastat a stávají se. Podobně elegantní a jednoduchý útok lze při troše štěstí provést přímo na server. Některá zařízení mají zveřejněné konfigurace přímo na internetových stránkách. Stačí pouze zadat správný dotaz do webového prohlížeče a přečíst si konfigurační soubory, včetně přístupových práv.

Bezpečností VoIP sítí obecně se zabývají mnozí odborníci z firem, které se podílejí na vývoji a zdokonalení technologie VoIP. Vznikla také organizace VOIPSA[7] (Voice over IP Security Alliance), která se touto problematikou zabývá a v roce 2005 vydala dokument VoIP Security and Privacy Threat Taxonomy[8] , který nastiňuje možnosti, jaké útoky lze provést na protokol SIP. Zobecněnější seznam útoků je na obrázku 5.

Nutné dodat, že při stejném útoku na různá koncová zařízení, mohou být výsledky odlišné.

7 http://www.voipsa.org/

8 http://www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf



Obr. 5 – Taxonomie rizik VoIP dle VOIPSA [5]

4.1 Mapování sítě

Podobně jako příslušníci armády neprovádí bezhlavý útok na neznámém území, ale snaží se jej co nejlépe zmapovat, i počítačový útok by měl být připravený. V první řadě musí útočník získat informace o síti a přítomných zařízeních. Tyto informace pak mohou napovědět, kde jsou slabá místa, která se dají zneužít nebo jak dále při útoku postupovat. Volně dostupnými programy lze zjistit, které IP adresy jsou používány, která zařízení jsou v daném okamžiku aktivní, na jakých portech zařízení pracuje, operační systém zařízení

a mnoho dalších informací, které mohou útočníkovi pomoci. Užitečným nástrojem je program Wireshark[9]. Jedná se o program, který zaznamenává veškerý datový provoz na zvoleném síťovém rozhraní, přehledně jej zobrazuje uživateli, vytváří statistiky atd. Velice podobnou funkci má i program Cain and Abel[10].

4.2 DoS a DDoS útoky

Pojem DoS (Denial of Service) [6] útok neznamená pouze jeden konkrétní útok, ale obecné značení, že útok pochází od jednoho uživatele (útočníka), který se snaží znemožnit dostupnost služby. Útočníkovi v určitých případech může stačit poslat pouze jeden jediný paket, který bude obsahovat takové informace, že se napadený systém zablokuje nebo zhroutí. Při tomto útoku se předpokládají velmi dobré znalosti systému, na který je útok prováděn. Další možností je zahlcení zprávami, které server zatěžují nejvíce. U serverů sítí SIP se může jednat například o snahu registrovat se. Server musí pro každou přijatou registrovací zprávu vypočítat pomoci algoritmu MD5 řetězec, který porovnává s přijatým. Právě výpočet pomoci algoritmu MD5 může být na pomalejších systémech náročný a mohou stačit nepřetržitě zasílané požadavky pouze od jednoho uživatele.

Naopak, DDoS (Distributed Denial of Service) útoky značí, že zprávy jsou posílány od velkého množství uživatelů. Může se jednat o stovky nebo desetitisíce stanic, zasílajících ve stejnou dobu požadavky na jediný server. Při takto velkém náporu se server přetíží a není schopen obsluhovat další požadavky. Pokud má cíl veřejnou IP adresu, tak pro provedení těchto útoků není nutné mít přístup do sítě.

4.2.1 Program SIPp

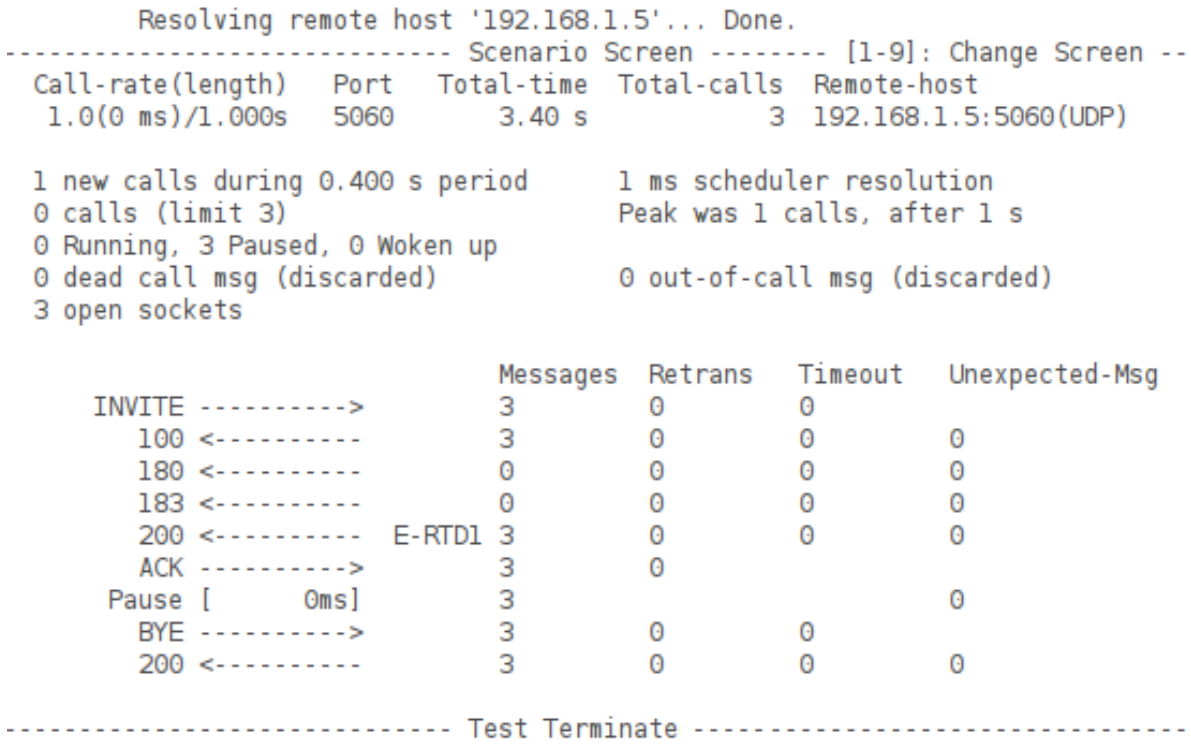
DoS útok lze provést například pomoci programu SIPp[11]. Není to program vytvořený pro provádění útoků, ale pro testování zatížení ústředen a správného scénáře zasílání zpráv. Scénářem se myslí přesně daná posloupnost signalizačních zpráv. Program sám obsahuje několik vytvořených scénářů, ale umožňuje také použít vlastní vytvořený scénář. Při vytváření scénáře přesně definujeme strukturu zprávy a použité hodnoty. To je možné využít pro zanesení úmyslné chyby do zprávy, která způsobí „zamrznutí“ testovaného zařízení. Já

9 http://www.wireshark.org/

10 http://www.oxid.it/cain.html

11 http://sipp.sourceforge.net/

však níže uvedu mnou vyzkoušený obecnější příklad, a to zaplavení testovaného zařízení zprávami INVITE.



Obr. 6 – Výstup programu SIPp

Na obrázku 6 je možné vidět příklad hlavní obrazovky programu, kde je znázorněn scénář zasílání zpráv, počet správně odeslaných a přijatých zpráv a počet zpráv, které přišly v jiném pořadí, než scénář předpokládá. K tomu dochází například právě při zahlcení zařízení mnoha zprávami. Dále program zobrazuje aktuální data o nově vytvářených hovorech. Počet vytvořených hovorů je možné zadat jako parametr při spuštění programu a dále je v reálném čase měnit a to pomoci kláves + a – o jeden hovor víc, respektive méně, nebo pomoci kláves * a /, kdy se počet hovorů zvýší, respektive sníží o 10. Pomoci numerických kláves 1 až 9 můžeme přepínat na další obrazovky, na nichž program zobrazuje další data. Mezi zajímavé statistiky patří rozdělení jednotlivých hovorů do intervalů podle času odezvy testovaného zařízení.

Při vytvoření 60 současných hovorů na ústřednu Asterisk bylo již pozorovatelné mírné zpoždění při vytváření hovorů mezi dalšími uživateli. Při vytvoření 150 hovorů ústředna „spadla“.
Příklad spuštění programu byl následující:

```
./sipp -sn uac 192.168.1.5 -i 192.168.1.6 -r 40 -s sipp
```

-sn uac – scénář „uac“ (klient) obsažený v programu
-192.168.1.5 – adresa testovaného zařízeních
-i 192.168.1.6 – adresa počítače, na kterém je program spuštěn
-r počet současně vytvořených hovorů za 1 vteřitnu
-s – název účtu, na který je hovor vytvářených

Obrana proti DoS a DDoS útokům je poměrně náročná. Pokud přichází data pouze z jedné adresy, je možné dočasně data z této adresy blokovat ať manuálně nebo automaticky programem. Je ale nutné si uvědomit, že daná adresa může patřit některému poskytovateli internetu. Pod jeho veřejnou adresou tedy mohou vystupovat desítky, stovky i tisíce uživatelů. Pokud tedy poskytujeme veřejné služby, znemožníme je i potencionálním uživatelům ze zablokované adresy. Pokročilejší metodou může být podrobnější zkoumání příchozích dat a jejich zahazování v případě, že mají nejen stejnou zdrojovou IP adresu, ale například i obsahují v signalizačních zprávách stále stejné kontaktní údaje.

4.3 Útoky na DHCP a DNS server

Protože sítě VoIP nejsou samostatné sítě, ale závislé na sítích s protokolem IP, nemusí být útok proveden pouze na zařízení zajišťující VoIP [6]. Při vyřazení z provozu zařízení, které obstarává správný chod sítě, přestane fungovat celá síť a s ní i služby VoIP. Důležitými službami jsou DHCP (Dynamic Host Configuration Protocol) server a DNS (Domain Name System) server.

DHCP server dynamicky přiděluje IP adresy určitého rozsahu zařízením, která se do sítě přihlásí (pokud nemají nastavenou IP adresu staticky). Vyřazením DHCP serveru neobdrží zařízení adresu a nemůže být přihlášeno do sítě. DHCP server může být vyřazen z provozu například vyhladověním adres – vyčerpá všechny adresy, které má k dispozici.

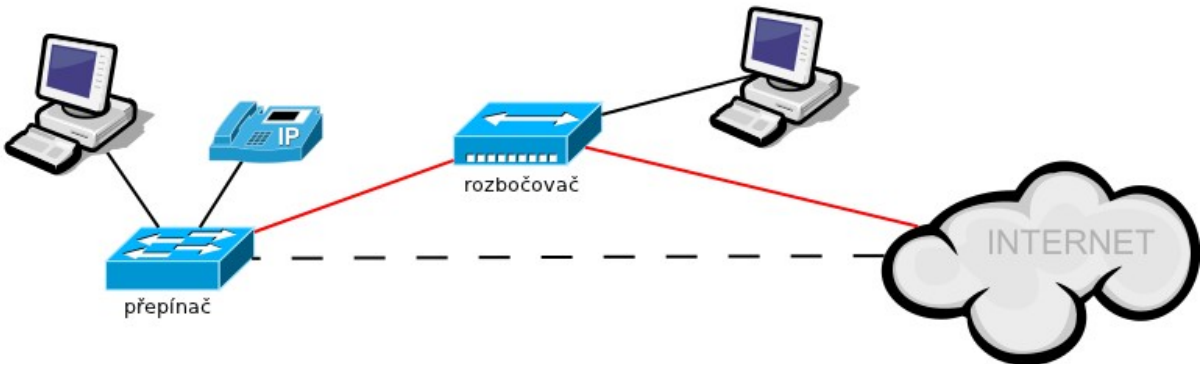
DNS server zajišťuje překlad doménových jmen na IP adresy. Bez jeho správné funkčnosti nejsou data zasílána na správnou adresu. Pokud má útočník přístup k síti, může zde umístit vlastní DNS server s upravenými záznamy, a pokud odpoví na dotazy zařízení rychleji než DNS server se správnými záznamy, budou data zaslána na jiné adresy.

4.4 Odposlech dat

Odposlechem dat se útočník může dozvědět velké množství informací o běhu sítě, které může použít na některý další útok, nebo může odposlouchávat přímo hlasové pakety a z nich sestavit hovor bez ztráty kvality zvuku. Z těchto hovorů je možno získat citlivé informace a například při ovládání bankovního účtu telefonem může útočník získat přístupový pin. Útočník také může použít program, který zjistí, jaké klávesy byly během hovoru stisknuté. To pro něj může být užitečné, pokud uživatel zadává pin pomoci stisknutí kláves. Odposlouchávání může být použito pro sběr telefonních čísel příchozích a odchozích hovorů a vytvoření databáze, která může být použita pro některé složitější útoky.

Výhodné může být i zaznamenávání TFTP komunikace, neboť mnoho IP telefonů po zapojení stahuje z TFTP serveru konfigurační soubory. Ty mohou obsahovat nastavení telefonu včetně přístupového hesla. Tyto informace lze využít k vniknutí do nastavení telefonu například pomoci služby telnet nebo přes webové rozhraní a přenastavit konfiguraci zařízení.

Aby měl útočník vůbec možnost data odposlouchávat, musí mít přístup k médiu. Nejsnadnější je odposlouchávat přenosy mezi zařízeními WiFi, která nemají nastavené šifrování. V takovém případě stačí útočníkovi pro odposlech například přenosný počítač s podporou WiFi. Pokud se chce útočník napojit na spoj vedený kabelem (klasický síťový kabel zakončený koncovkou RJ-45), vloží do tohoto spoje zařízení rozbočovač nebo zaplaví přepínač. Ukázka zapojení na obrázku 7. Rozbočovač se již v dnešních sítích nevyužívá, neboť jeho velkou nevýhodou bylo, že když přišla data na jeden port, byla odeslána na všechny ostatní porty. V dalším vývoji se tento princip ukázal jako nevyhovující. Pro útočníka je to ale stále dobrá pomůcka, jak se dostat k datům přenášeným po síti.



Obr. 7 – Odposlouchávání dat vložením rozbočovače

Stejně jako rozbočovač se chová v určitých případech i přepínač. Nemusí být tedy nutné fyzicky síť rozpojovat a vkládat do spoje rozbočovač. Přepínač obsahuje paměť, do které si postupně ukládá cílovou MAC (Media Access Control) adresu zařízení a port, přes který se mají data odeslat. Pokud tuto paměť dokážeme zaplnit a zahltit, začne se přepínač chovat úplně stejně jako rozbočovač.

Pro náročnější útoky nestačí mít pouze přístup k médiu a možnost odposlouchávat datový provoz, ale útočník musí zajistit, aby datový provoz procházel přes jeho zařízení. Existuje způsob, jak tohoto požadavku docílit, pokud se útočník připojí na přepínač. Pokud útočník změní záznamy v paměti přepínače (ARP tabulce), má vyhráno. Přitom tento úkol není těžký. Útočník zašle oběti paket, který říká, že má stejnou MAC adresu jako server, přes který musí jít veškerá komunikace ven ze sítě. Naopak serveru zašle paket, že má stejnou MAC adresu jako oběť. Přepínač si v paměti přepíše ARP tabulku tak, že data od oběti zašle přepínač nám, my vyplníme správnou MAC adresu (dosadíme adresu oběti) a data přepošleme na server. Takto si útočník zajistí přenos dat přes svůj počítač. Popsaný útok se nazývá otrávení tabulky (ARP Cache poisoning). Program, který velice snadno dokáže provádět tento útok a zobrazovat přenášená data v systému Windows se jmenuje Cain and Abel, pro linux je to například Ettercap[12].

12 http://ettercap.sourceforge.net/

4.4.1 Program Ettercap

Ettercap je linuxový program umožňující realizovat útok Muž uprostřed (Man In The Middle) několika způsoby, mimo jiné i výše popsaným útokem otrávení ARP tabulky (ARP cache poisoning). Uživatel zadává IP adresy zařízení, mezi kterými chce odposlouchávat datový provoz. IP adresy je možné zadávat jednotlivě (192.168.1.1) nebo skupinově (192.168.1.1-5 nebo 192.168.1.1-5, 10, 20). Uživatel má také možnost specifikovat port/y pro odposlouchávání, a to opět jednotlivě nebo skupinově jako u IP adres. Program je nutné spouštět jako superuživatel. Data jsou zobrazována v reálném čase na obrazovce nebo ukládána do souboru pro následnou analýzu. Aby nebyl datový provoz příliš velký a data byla přehledná, program umožňuje sledovaná data filtrovat podle IP adres a portů. Příklad spuštění programu je následující:

```
ettercap -T -M arp -i eth0 /192.168.1.10/5060 /192.168.1.1-9/5060
```

-T – program se spustí v konzolovém režimu (možné spustit také v grafickém režimu -G nebo v konzolovém grafickém režimu -C)

-M – uvádí typ útoku (v našem případě pomoci APR, možnosti jsou pomoci ICMP, DHCP,..)

-i – název síťového zařízení

// // - Mezi první dvě lomítka zapíšeme IP adresu/adresy prvního zařízení, mezi druhá lomítka adresu/y druhého zařízení. Čísla portů se píší za lomítka. Pokud chceme sledovat datový provoz celé sítě, napíšeme pouze // //.

Podrobný návod použití programu Ettercap se nachází na domovské stránce programu. Program je možné nainstalovat buď z domácích stránek nebo z repozitáře.

4.5 Odstranění registrace - Registration Removal

Jak již bylo zmíněno, všechny telefony podporující protokol SIP, po zapojení do sítě/elektřiny odešlou požadavek na registraci vůči serveru. Server díky této komunikaci ví, na jaké adrese má účastníka kontaktovat. Tato registrace musí být opakována (standardně v intervalu 3600s). Útok smazání registrace spočívá v zaslání zprávy pro odhlášení od SIP serveru. Ve zprávě je nutné nastavit údaje, aby se SIP server domníval, že zpráva pochází od

oběti. Do odhlašovací zprávy vložíme doménové jméno SIP serveru, uživatelské jméno oběti a její IP adresu. Zařízení oběti je po odhlášení ze serveru nedostupné (neplatí pro volání přímo přes IP adresu) až do vypršení intervalu, kdy se opět registruje. Pro dlouhodobější odstavení oběti je nutné tuto zprávu zasílat také v určitém intervalu. Zasílaná zpráva je uvedena níže.

```
REGISTER sip:10.1.101.99 SIP/2.0
Via:SIP/2.0/UDP 10.1.101.99:5060;branch=83c598e0-6fce-4414-afdd-11a
From: 4000 <sip:4000@10.1.101.99>;
tag=83c5ac5c-6fce-4414-80ce-de7720487e25
To: 4000 <sip:4000@10.1.101.99>
Call-ID: 83c5baaa-6fce-4414-8ff6-f57c46985163
CSeq: 1 REGISTER
Max-Forwards: 70
Contact: *
Expires: 0
Content-Length: 0
```

Tučným písmem jsou zvýrazněné položky, které slouží k odhlášení serveru, zvýrazněné položky je nutné zaměnit za údaje oběti.

Na internetu je volně dostupný program *erase_registrations*[13], který umí tento útok provést. Způsob užívání je následující:

```
./erase_registrations <EthernetInterface> <TargetUser> <TargetDomainIP> <DestinationIP> –h -v
```

EthernetInterface – název rozhraní přes které se zpráva odešle (např. eth0, wlan0,..)
TargetUser – uživatelské jméno oběti
TargetDomainIP – IP adresa SIP serveru
DestinationIP – IP adresa oběti
-h – výpis nápovědy, -v – spuštění podrobného výpisu

Příklad použití:

```
./erase_registrations eth0 3000 10.1.101.2 10.1.101.30
```

13 http://www.hackingvoip.com/sec_tools.html

Způsobů, jak se ubránit tomuto útoku je několik. První způsob je zavést nutnost autorizace uživatele. Pokud se uživatel neautorizuje, SIP server zprávu nepřijme. Druhá možnost využívá standardu, který specifikuje, že každé zařízení podporující protokol SIP musí pracovat jak s protokolem UDP, tak TCP. Pokud zařízení mezi sebou vytvoří stálé TCP spojení, tak se na jeho začátku vygeneruje náhodné číslo a se zasláním každého dalšího paketu se toto číslo zvýší o jedničku. Tato skutečnost útočníkovi znesnadňuje podvrhnutí zprávy. Dalšími možnostmi obrany je snížení intervalu obnovování registrace, použití SIP firewallu a zapojení SIP telefonů na zvláštní síť pomoci techniky VLAN.

4.6 Přidání účastníka - Registration Addition

Tímto útokem podvrhneme serveru novou kontaktní adresu. Ta se přidá do seznamu kontaktních adres (nebo přepíše stávající, například u ústředen Asterisk). Pokud přidáme kontakt na naše zařízení, tak v případě příchozího hovoru začne vyzvánět i naše zařízení. Pokud se útočníkovi podaří zvednout sluchátko dříve než oběti, příjme hovor on. Tato varianta útoku se dá využít pro zmatení uživatelů, a to přidáním velkého množství kontaktů. Při vyzvánění telefonu oběti tak může telefon vyzvánět dalším spolupracovníkům oběti.

V případě přepsání kontaktu za jiný, je možné přesměrovat hovor na útočníkem dostupné zařízení a oběť se o hovoru nedoví. Pokud nechce útočník hovor přijmout, ale jen znemožnit, nastaví neexistující kontakt a hovor se nikdy neuskuteční.

Na internetu je běžně dostupný nástroj k provedení tohoto útoku *add_registrations*[14]. Jeho nevýhodou je, že není schopen přidat registraci pokud ústředna požaduje po uživateli heslo. Způsob užívání tohoto programu je následující:

```
./add_registrations <EthernetInterface> <NewContactUser> <NewContactIP>
<TargetDomainIP> <DestinationIP>
```

EthernetInterface – název rozhraní přes které se zpráva odešle (např. eth0, wlan0,..)

NewContactUser – jméno nového kontaktu které se zobrazí druhé straně

NewContactIP – IP adresa zařízení nového kontaktu

TargetDomainIP – IP adresa SIP serveru

DestinationIP – IP adresa oběti

14 http://www.hackingvoip.com/sec_tools.html

-e – pro zachování původního kontaktu v případě, že se kontakt přemazává

-h – výpis nápovědy

-v – spuštění podrobného výpisu

Příklad použití:

```
./add_registrations eth0 clark 192.168.1.203 192.168.1.5 192.168.1.202
```

Pravé přihlášení uživatele Clark:

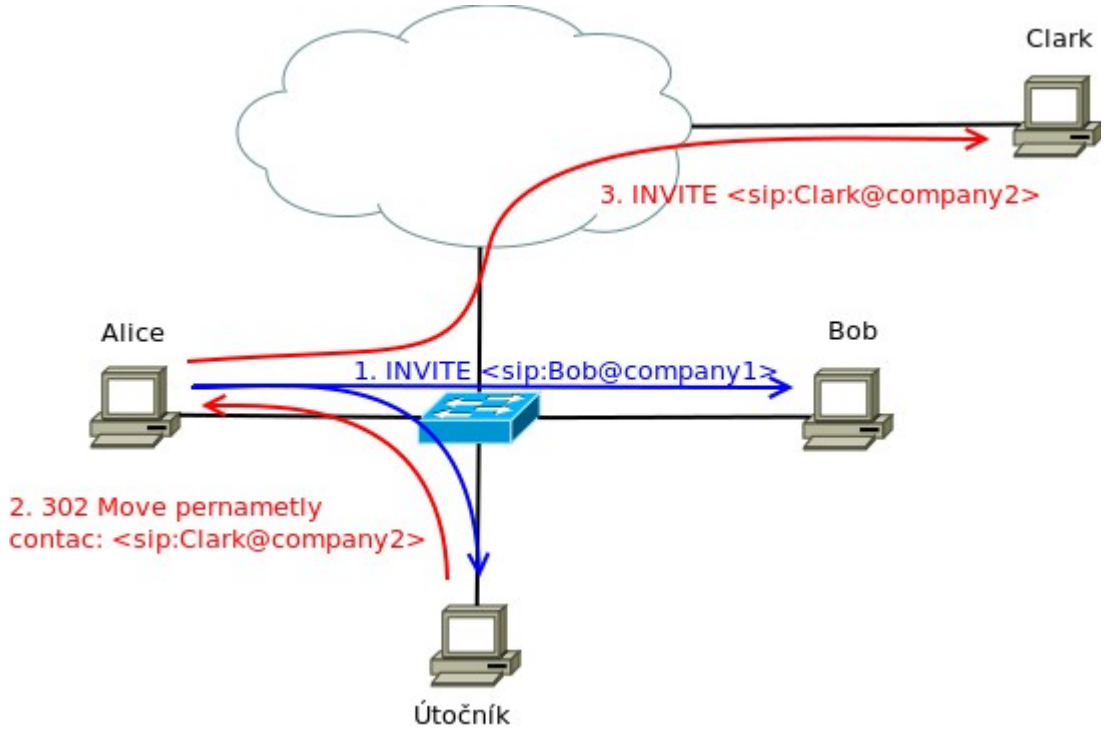
```
REGISTER sip:192.168.1.5 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.203;rport;branch=z9hG4bKxqphojpr
Max-Forwards: 70
To: "clark" <sip:clark@192.168.1.5>
From: "clark" <sip:clark@192.168.1.5>;tag=cujeg
Call-ID: saiklyxwfttxeww@debian3
CSeq: 726 REGISTER
Contact: <sip:clark@192.168.1.203>;expires=0
Allow:
INVITE,ACK,BYE,CANCEL,OPTIONS,PRACK,REFER,NOTIFY,SUBSCRIBE,INFO,MESSAGE
User-Agent: Twinkle/1.4.2
Content-Length: 0
```

Falešné přihlášení pomoci programu add_registration:

```
REGISTER sip:192.168.1.5 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.202;branch=6119b384-b824-44ed-9392-
74ca6702246e
From: clark <sip:clark@192.168.1.5>;tag=6119cb36-b824-44ed-a3a5-
8955c3d46934
To: <sip:clark@192.168.1.5>
Contact: <sip:192.168.1.203@192.168.1.202>
Call-ID: 6119e7c6-b824-44ed-9c23-a2218bb09137
CSeq: 100 REGISTER
Expires: 3600
User-Agent: Hacker
Max-Forwards: 16
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0
```

4.7 Přesměrování hovoru - Redirection attack

Jedná se o útok, kdy útočník nemodifikuje zprávy. Stačí, pokud může odposlouchávat přenos dat a zasílat data do sítě. V případě, že oběť zasílá někomu požadavek INVITE, útočník zašle oběti zprávu odpovědi 301 (přemístěn trvale) nebo 302 (přemístěn dočasně) a vloží kontakt, kam má být hovor přesměrován. Můžeme zvolit kontakt na útočníkem ovládané zařízení, na náhodně vybrané zařízení ze seznamu uživatelů nebo neexistující adresu. Aby byl útok úspěšný, musí oběť dostat odpověď od útočníka dříve, než odpověď volaného uživatele. Znázorněno na obrázku.



Obr. 8 – Redirection Attack

Ukázka zprávy INVITE vytvořené uživatelem Alice a odpovědi, kterou uživatel dostane od útočníka.

```
INVITE sip:Bob@company1 SIP/2.0
Via: SIP/2.0/UDP pc33.company1
To: Bob <Bob@company1>
From: Alice <alice@atlanta.com>
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Contact: <sip:alice@atlanta.com>
```

```
SIP/2.0 301 Moved Permanently
Via: SIP/2.0/UDP pc33.company1
To: Bob <Bob@company1>
From: Alice <alice@atlanta.com>
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Contact: <sip:Clark@company2>
```

Toto byla ukázka, jak přesměrovat odchozí hovory napadeného. Stejný princip lze uplatnit i na příchozí hovory. Níže uvedený program pro tento typ útoku nerozeznává příchozí a odchozí hovory.

Pro tento útok byl vytvořen program *redirectpoison*[15]. Způsob užívání tohoto programu je následující:

```
./redirectpoison <EthernetInterface> <TargetSourceIP> <TargetSourcePort>
<"Contact Information"> –h -v
```

EthernetInterface – název rozhraní přes které se zpráva odešle (např. eth0, wlan0,..)

TargetSourceIP – IP adresa oběti

TargetSourcePort – port

"Contact Information" – nový kontakt, kam bude hovor přesměrován

-h – výpis nápovědy

-v – spuštění podrobného výpisu

Příklad použití:

```
./redirectpoison eth0 10.1.101.30 5060 "<sip:6000@10.1.101.60>"
```

4.8 Programy SIPdump a SIPcrack

Tato dvojice programů společně hrubou silou prolamuje hesla použitá v signalizační zprávě REGISTER. Program Sipdum odposlouchává komunikaci na síťovém rozhraní a kdykoliv zaznamená zprávu REGISTER obsahující údaje k přihlášení, zapíše je do souboru a následně k nim přidá hash řetězec odeslaný uživatelem na server a další údaje potřebné pro přihlášení. Uložený soubor následně zpracovává program Sipcrack, který z dat uložených v souboru vytváří pomoci známého algoritmu MD5 řetězec hash. Jediný údaj, který není v uloženém souboru, je použité heslo. To získává program z takzvaného slovníku (většinou soubor, ale může být i vstup z klávesnice). Program postupně čte slova ze slovníku, dosazuje je jako heslo a vypočítává řetězec hash. Pokud se vypočtený řetězec shoduje s odposlechnutým řetězcem, pak bylo heslo úspěšně nalezeno.

15 http://www.hackingvoip.com/sec_tools.html

Pokusně byl použit slovník s více než 11 000 000 slovy (platné heslo bylo jako poslední ve slovníku) a na dvoujádrovém procesoru o taktu 1,8 Ghz od firmy Intel trvalo nalezení hesla 12 sekund.

Spuštění programů:

```
sipdump -i <interface> <target file>
sipcrack -w <passwordlist> <target file>
```

-i	rozhraní, na kterém bude program poslouchat
-target file	soubor, kam se ukládají odposlechnuté informace
-w	slovníkový soubor (jedno slovo na řádek)

Ukázka jednoho záznamu zachycené komunikace pomoci programu Sipdump. Jednotlivé položky jsou odděleny uvozovkami v pořadí:

```
192.168.1.102"192.168.1.50"bob"asterisk"REGISTER"sip:192.168.1.50"6244a364"
"""MD5"5cca5ff357c45c6e3cc5a772275edbe8
```

Obecnou ochranou proti prolomení hesla je použití bezpečného hesla a dodržení zásad jeho bezpečnosti. To znamená používat hesla o minimálně 8 znacích, použít velká písmena, číslice nebo speciální znaky. Další ochranou je použití šifrování. Útočník tedy nebude mít možnost číst odposlechnutou komunikaci, a tím ani údaje potřebné pro prolomení hesla.

4.9 Ostatní útoky a programy

Popsal jsem pouze pár útoků. Existuje mnoho dalších, někdy velice si podobných útoků. Pro tyto útoky lze na internetu nalézt velké množství programů, které bývají jednoduché a slouží pouze pro jeden jediný útok. Ne vždy je snadné je nainstalovat a zprovoznit. Výjimkou je program Cain and Abel, který zvládá více útoků, a to nejen na službu VoIP. Avšak nástroj, který by zvládal více penetračních testů, volně k dispozici není. Velké množství programů je volně ke stažení na webové stránce

http://www.hackingvoip.com/tools.html a také na stránkách VOIPSA http://www.voipsa.org/Resources/tools.php#Tool%20Tutorials%20and%20Presentations.

5 Ochrana komunikace

I když je většina útoků na VoIP zařízení cílená a využívá jiné techniky, v základu je problém, že útočník vidí probíhající komunikaci a dokonce je schopen s touto komunikací určitým způsobem manipulovat či ji vynutit podvrhnutými zprávami. Pro lepší bezpečnost před většinou útoků tedy musíme zamezit přístupu k datovému toku. Velice často uváděnou ochranou je použití takzvaných VLAN (Virtual LAN) sítí. Jedná se o virtuální sítě vedené v rámci již vybudované síťové infrastruktury s přepínači, které tuto techniku ovládají. Přepínače vytvoří dvě virtuální sítě, které po páteřních spojích mohou vést jediným spojem, ale na přepínači se každému portu přidělí jedna z virtuálních sítí. Tím tedy vytvoříme samostatnou virtuální síť pouze pro VoIP zařízení a samostatnou virtuální síť pro zbytek síťových zařízení bez nutnosti změny kabeláže. Tento způsob ochrany je ale možný pouze v LAN sítích a data stále proudí v čitelné podobě.

Pro větší zabezpečení je možné využít například již známé šifrovací techniky a protokoly. Jedná se o protokoly SRTP[16] a ZRTP[17] pro data a IPsec[18], TLS[19] nebo SMIME pro signalizaci. Sama norma však před jeho používání varuje. TLS je funkční pouze v případě, že všechna zařízení SIP toto zabezpečení podporují. Pokud jediné zařízení tuto metodu šifrování nepodporuje, šifrování se nepoužije. A protože nikdy nevíme (nepoužijeme-li přímo směrování) kudy data projdou, nemůžeme si být jistí použitím šifrování. Stejně tak norma varuje před použitím SMIME, protože mohou být po trase zařízení, které tento protokol nepodporují.

5.1 Způsoby ochrany

- **VLAN** – jedná se o techniku, kterou umí některé přepínače. VLAN dovolují seskupit zařízení do virtuálních sítí bez ohledu na jejich fyzické umístění. Zařízení v jedné virtuální síti poté spolu komunikují jako by byly na jediné fyzické síti.

16 http://www.ietf.org/rfc/rfc3711.txt

17 http://www.ietf.org/rfc/rfc6189.txt

18 http://www.ietf.org/rfc/rfc2401.txt

19 http://www.ietf.org/rfc/rfc2246.txt

- **SRTP** – Tento protokol je bezpečnější alternativou k protokolu RTP. Šifruje data obsažená v užitečném zatížení. Tím se zamezí jejich čtení při zachycení a následnému sestavení zvukového nebo obrazového toku. Autentizaci podléhají všechna data patřící k protokolu RTP a navíc mohou být přidána dvě nepovinná pole. Jedno z nich slouží pro výměnu hlavního šifrovacího klíče. Pro jejich výměnu je možné použít protokol SDP, ale ten data nijak nezabezpečuje. Proto se doporučuje zabezpečit je protokolem TLS nebo IPsec. Druhé pole slouží pro uložení šifrovaného kontrolního součtu hlavičky a těla RTP paketu. Pokud je toto pole použito, chrání před neautorizovanou změnou dat v paketu.
- **ZRTP** – jedná se o rozšíření protokolu SRTP mechanizmem pro počáteční výměnu symetrického klíče, není tedy potřeba využívat jiných zabezpečovacích protokolů.
- **IPsec** – jedná se o známý způsob šifrování na síťové vrstvě. To znamená, že je nezávislé na šifrovaných datech. Šifrování však musí podporovat všechna zařízení pracující s protokolem SIP nebo SRTP.
- **TLS** – jedná se o další známý způsob šifrování, tentokrát na transportní vrstvě. Pro použití tohoto šifrování je již potřeba podpora aplikací. Opět je podmínkou, aby šifrování pomoci TLS podporovala všechna zařízení pracující se zašifrovanými daty.
- **SMIME** – jde o zabezpečovací mechanizmus používaný jako standard pro šifrování a podepisování elektronické pošty.

6 Nástroj pro testování bezpečnosti VoIP a jeho testování

Po nastudování teoretické části problémů protokolu SIP a vyzkoušení několika programů jsem začal pracovat na nástroji, který by testoval základní zabezpečení a správnou funkčnost zařízení. Hlavním cílem bylo vytvořit nástroj s otevřeným kódem a možností přidávání testů. V současné podobě má program čtyři moduly, kdy každý z nich představuje jeden typ útoku. Typy útoků, princip fungování programu a jeho spouštění bude detailněji popsáno v následujících podkapitolách. Program byl napsán v jazyce C++ a je určen pro systém Linux.

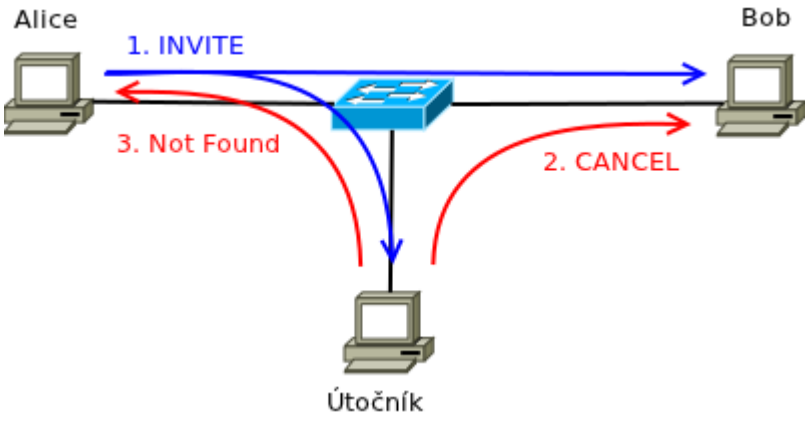
6.1 Zahození hovoru – podsunuté zprávy

6.1.1 Popis útoku

Zahození hovoru je alternativa ke známému útoku Teardown [1] pomoci zprávy BYE nebo výše popsanému útoku přesměrování hovorů. Útok Teardown probíhá tak, že útočník musí odposlechnout údaje předávané během navázání hovoru. Sám si vytvoří zprávu BYE a vloží do ní odposlechnuté údaje. Odeslání zprávy jednomu z účastníků má za následek ukončení již probíhajícího hovoru.

Vytvořená aplikace se pokouší znemožnit komunikaci mezi zařízeními. S využitím knihovny pcap program přepne síťové rozhraní do promiskuitního režimu a naslouchá datům na portu 5060, tedy portu určenému pro signalizační protokol SIP. Při zachycení žádosti (ne všech, viz dále) program okamžitě zasílá nazpět odpověď 404 Not Found a spoléhá na to, že v odpovědi bude rychlejší než účastník, kterému byla zpráva určena. Tím pádem strana, která žádost odeslala, dostane mylnou zprávu, že dané zařízení neexistuje, a když následně dostane odpověď od druhého účastníka, ignoruje ji. Pokud je zaslanou žádostí vytvoření hovoru, nejsme schopni zabránit vyzvánění na straně druhého účastníka. Toto vyzvánění tedy okamžitě ukončuji odesláním žádosti CANCEL volanému účastníkovi. Ten může slyšet jen krátké zazvonění a zobrazí se mu záznam o nepřijatém hovoru. Program neodpovídá na

žádosti ACK, SUBSCRIBE a CANCEL. Jde především o test znemožnění registrace a navázání hovoru.



Obr. 9 – Příklad zapojení a zasílání zpráv při útoku zahození hovoru

6.1.2 Zabezpečení

Jak již bylo zmíněno, mnoho útoků si je velice podobných a jedním opatřením lze předejít více nepříjemnostem. Problémem tohoto útoku, kromě přístupu k nešifrované signalizaci, je i akceptování odpovědi zařízení, které není účastníkem hovoru. Obranou tak může být používání transportního protokolu TCP namísto standardně užívaného UDP nebo oddělení datové a hlasové komunikace pomoci VLAN sítí [6]. Nejlepší obranou je ovšem zamezení přístupu k signalizačním zprávám šifrováním pomoci TLS. Při tomto útoku se předpokládá, že útočník má možnost odposlouchávat data pouze na místní síti. V dnešní době není až tak těžké pořídit si ústřednu, která šifrování TLS podporuje, a stejně tak koncová zařízení a šifrovat hovory pouze uvnitř firmy. Problém může nastat při hovorech mimo firmu.

6.1.3 Použití

Formát spuštění programu je následující:

```
sudo ./sipTool -a <číslo útoku> -i <interface> -f <filtr> -w <soubor>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-f – pro filtrování odposlouchávaných dat je možné definovat filtr například jako u programu TCPdump, pouze mezery je nutné nahradit podtržítkem „_“

-w – soubor pro záznam výstupu programu

Výstupem programu je výpis zaznamenaných a vykonaných akcí. Tedy výčet toho, která zpráva byla zaznamenána, z které IP adresy na kterou IP adresu byla zaslána, od kterého uživatele kterému a dále které zprávy a kam program odeslal.

Příklad spuštění a výstupu programu:

```
sudo ./sipTool -a 1 -i wlan0
Catched request: INVITE 192.168.1.201->192.168.1.5
From: sip:bob@192.168.1.5 To: sip:203@192.168.1.5
Send CANCEL to: 192.168.1.5
Send 404 Not Found to: 192.168.1.201
Catched request: REGISTER 192.168.1.202->192.168.1.5
From: sip:dan@192.168.1.5 To: sip:dan@192.168.1.5
Send 404 Not Found to: 192.168.1.202
```

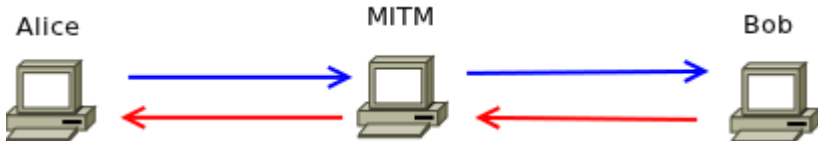
6.2 Man in The Middle – únos hovoru

6.2.1 Popis útoku

Tento útok je velice nebezpečný [6]. Pokud se útočník dostane do komunikace, která přes něj prochází, může útočník plně sledovat a řídit signalizační zprávy. Přesměrovat tok dat přes útočníkovo zařízení lze například výše zmíněným útokem otrávení ARP tabulky, nebo přesměrovat všechny hovory na útočníkovo zařízení a zprávy z něj dále přeposílat.

V mém programu není implementována funkce jak tohoto útoku docílit, program pouze emuluje chování SIP proxy serveru. Nejprve musí klient zaslat registrační zprávu, program si vytvoří záznam, z které IP adresy a portu se uživatel hlásí, a požadavek přepošle ústředně. Při obdržení zprávy od ústředny prochází pole záznamů, a podle uživatelského jména (pokud jde o odpověď hledá uživatele v položce From, pokud o žádost, tak v položce To) zjistí, kam má zprávu přeposlat.

Před přeposláním je vždy potřeba zprávu upravit, přepsat IP adresy a porty. Protože protokol SIP je protokol textový, přepisování se provádím prostým vyhledáním řetězce a jeho přepsáním. Pokud zpráva směřuje na server, musí program změnit útočníkovu adresu na adresu serveru (stejně tak porty). a poté přepsat adresu klienta na adresu útočníka. Jedná se o adresy v hlavičce zprávy, v polích Via, From, To a Contact. Pokud se jedná o registraci s autorizací, tak adresa uvedená pro autorizaci se přepisovat nesmí. Stejně tak pro vytvoření hovoru musejí zůstat zachované adresy a porty v protokolu SDP.



Obr. 10 – ukázka útoku Man in The Middle

6.2.2 Zabezpečení

Pro ochranu sítě před útoky Man in The Middle je třeba útočníkovu odepřít přístup k síti nebo znemožnit přístup k datům a jejich zneužití. Odepřít přístup k síti můžeme například provozem VoIP zařízení na oddělené síti. Toto opatření útočníkovi práci ztíží, ale nemusí síť ochránit. Bezpečnější je odepřít útočníkovi přístup k datům, například šifrováním. Pro opravdovou bezpečnost je potřeba ověřovat pravost certifikátů.

6.2.3 Použití

Formát spuštění programu je následující:

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -w <soubor>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu ústředny, za kterou se vydáváme a na kterou data zasíláme

-w – soubor pro záznam výstupu programu

Výstupem programu je výpis zpráv, které přes program přeposílá a chybové zprávy.

Příklad spuštění a výpisu programu:

```
./sipTool -a 2 -i wlan0 -s 192.168.1.5
192.168.1.201->192.168.1.5
REGISTER sip:bob@192.168.1.9->sip:bob@192.168.1.9
192.168.1.5->192.168.1.201
401 Unauthorized sip:bob@192.168.1.5->sip:bob@192.168.1.5
192.168.1.201->192.168.1.5
REGISTER sip:bob@192.168.1.9->sip:bob@192.168.1.9
192.168.1.5->192.168.1.201
200 OK sip:bob@192.168.1.5->sip:bob@192.168.1.5
```

6.3 Zaslání REGISTER – získání informací

6.3.1 Popis útoku

Při tomto útoku program zasílá na zadanou adresu (ústřednu) žádost REGISTER. Z odpovědi může být patrné, zda má uživatel na ústředně vytvořený účet nebo ne. Díky této informaci lze poté provádět útoky na konkrétní uživatele. Proto je vhodné se těmto průzkumným útokům bránit.

Program má předem vytvořenou žádost REGISTER, kterou odesílá, a útočník je schopen zadat uživatele nebo seznam uživatelů, kterým se má zpráva odeslat. Může tak učinit buď pomocí seznamu v textovém souboru, kdy každý řádek představuje jednoho uživatele, nebo zadáním rozmezí telefonních čísel. Rozmezí zadá pomoci prvního a posledního čísla. Pokud bylo zadáno pouze první číslo, zpráva se odešle jednomu uživateli. Program po spuštění bere seznam těchto uživatelů a zasílá zprávy ihned za sebou. Pro příjem a vyhodnocení odpovědí si program vytvoří nové vlákno, které pouze zprávy přijímá a vypisuje na obrazovku údaje, zda má uživatel na ústředně vytvořený profil nebo ne a zda je potřebná autorizace. Pro případné zpoždění příjmu zprávy je nastavena prodleva na půl vteřiny mezi příjmem zpráv.

Tento útok může při dostatečně dlouhém seznamu uživatelů nahradit programy pro DoS útok, protože zprávy zasílá okamžitě po sobě. Nejen proto byla přidána možnost zadat dobu čekání mezi odesláním zpráv, a to v milisekundách. Další důvod k tomuto kroku byl, že některé ústředny (hlas.802.cz a iptel.org) reagují na signalizační zprávu REGISTER pouze jedenkrát za vteřinu. Pokud bylo na ústřednu zasláno během jedné vteřiny zpráv REGISTER více, odpovídala ústředna stejným počtem zpráv, ale opakovala se vždy odpověď na první žádost. Po vteřině se zpráva odpovědi změnila, ale opět po dobu jedné vteřiny byla stejná.

6.3.2 Zabezpečení

Při testování bylo dále zjištěno, že ústředny patřící poskytovatelům VoIP (802.cz a iptel.org) jsou chráněny proti zjišťování vytvořených účtů. Při snaze přihlásit každého vygenerovaného uživatele ústředna odpověděla, že uživatel existuje a potřebuje být registrován. Nevíme tedy, zda uživatel má opravdu vytvořený účet nebo ne. Tím se zamezí přesně cíleným útokům. Ústředny také reagovali na zaslání zprávy pouze jedenkrát za vteřinu. Při zaslání dalších požadavků na registraci během této vteřiny zasílaly stále odpověď

na první žádost. Po vteřině se odpověď změnila, ale opět se zasílala na všechny žádosti po dobu další vteřiny.

6.3.3 Použití

Formát spuštění programu je následující:

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -e <první číslo> -E <poslední číslo> -S <prodleva> -u <soubor> -v <detailnější výpis>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu, kam bude zpráva zaslána

-e – první číslo z rozmezí. Pokud je použit pouze tento přepínač, bude zpráva zaslána pouze na toto jediné číslo

-E – poslední číslo z rozmezí

-u – soubor s uživateli, kterým bude zpráva zaslána

-s – prodleva mezi zasláním zpráv. Zadává se v milisekundách (1s = 1000ms)

-v – detailnější výpis. Vypíše číselný i řetězcový název všech přijatých odpovědí a na závěr vypíše seznam uživatelů, kteří existují.

Výstupem programu je výpis říkající, zda má uživatel vytvořený účet a zda je potřeba heslo pro registraci k účtu.

Příklad spuštění a výpisu programu:

```
./sipTool -a 3 -i wlan0 -s 192.168.1.5 -u examples/users
User: alice   exist and need to be authorized
User: bob     exist and need to be authorized
User: clark   exist and need to be authorized
User: dan     exist and need to be authorized
User: jane    does not exist
```

6.4 Zaslání libovolné zprávy – útok na software

6.4.1 Popis útoku

Tato část programu funguje stejně jako výše uvedené zaslání zprávy REGISTER, jen nemá zprávu předpřipravenou programem, ale čte ji ze souboru. Opět je možné zadat seznam

uživatelů ze souboru nebo rozmezím čísel a nastavit prodlevu mezi zasíláním zpráv. Proč vlastně byla vytvořena předchozí část programu, a k čemu je to dobré? Předchozí část byla vytvořena především jako testování vytvořených účtů na ústředně a pro pohodlnější užívání stačí předpřipravená zpráva. Tato část programu by měla sloužit pro testování, jak se zařízení chová vůči neúplným nebo nějakým způsobem deformovaným zprávám. Některá zařízení mohou zprávu odepřít a nezpracovat, jiná zase mohou bez problémů pracovat i s neúplnou či deformovanou zprávou a jiná se mohou zaseknout.

Tím, že lze vytvořit libovolnou zprávu, můžeme prakticky provést jakýkoliv útok (například zahlcení požadavky, podsunutí zprávy, přesměrování hovoru atd.). Pokud například víme, že probíhá určitý hovor a známe data ze zprávy INVITE, můžeme zaslat zprávu BYE, a tím hovor ukončit. Jen to není zautomatizovaný proces, musí se dělat ručně a trvá delší dobu. Dále můžeme opět vyzkoušet DoS útok. Tentokrát ale vylepšený o neúplnou nebo deformovanou zprávu, což by mělo ústředně zabrat více času na zpracování, a tím zlepšit účinnost útoku. Ověření této domněnky testováním neproběhlo. Ani testování chování různých zařízení na různě deformované zprávy neproběhlo nijak intenzivně. Je to testování časově náročné a vždy se vztahuje pouze k danému zařízení.

Způsob zápisu zprávy do souboru je takový, že se napíše textová zpráva SIP, a na každý řádek program automaticky doplní potřebnou sekvenci znaků pro ukončení řádku (\r\n). Pokud chceme zaslat více typů zpráv, musejí být oddělené sekvencí znaků /*/, a to na samostatném řádku. Nesmíme zapomenout před tuto sekvenci dát prázdný řádek, který udává konec zprávy. Nic nám nebrání zasílat zprávu SIP včetně dat z protokolu SDP. V tomto případě mezi zprávou SIP a daty protokolu SDP necháme volný řádek. Zpráva je ukončena koncem souboru nebo sekvencí znaků pro oddělení zpráv.

Způsob zasílání více zpráv probíhá tak, že se nejdříve přečte zpráva, a poté se zašle všem zadaným uživatelům. Program pak zkusí načíst ze souboru další zprávu a opět ji zaslat všem uživatelům.

6.4.2 Zabezpečení

Protože tento modulu nebyl vytvořen pro provádění útoků, ale pro ověření korektního chování zařízení, je zabezpečením i používání korektně chovajících se zařízení. Stejně tak tento modul může posloužit pro testování korektního chování při vývoji nového VoIP telefonu.

6.4.3 Použití

Formát spuštění programu je následující:

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -e <první číslo> -E <poslední číslo> -S <prodleva> -u <soubor> -m <soubor> -v <detailnější výpis>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu, kam bude zpráva zaslána

-e – první číslo z rozmezí. Pokud je použit pouze tento přepínač, bude zpráva zaslána pouze na toto jediné číslo

-E – poslední číslo z rozmezí

-u – soubor s uživateli, kterým bude zpráva zaslána

-m – soubor, kde je napsána zpráva nebo zprávy

-s – prodleva mezi zasláním zpráv. Zadává se v milisekundách (1s = 1000ms)

-v – detailnější výpis. Vypíše všechny přijaté zprávy se všemi položkami.

Příklad spuštění a výpisu programu:

```
./sipTool -a 4 -i wlan0 -s 192.168.1.5 -e 200 -E 202 -m examples/messages
404 found
404 found
404 found
503 error
503 error
503 error
```

6.4.4 Testování a srovnání

Telefonní ústředna Asterisk je pravděpodobně nejoblíbenější a nejflexibilnější volně dostupný projekt v oblasti telefonních poboček [7]. Její oblibě napomáhá široká škála podporovaných protokolů (např. SIP, H.323, MGCP, SCCP, IAX), možnost nahrávání hovorů, vzájemné propojování více poboček Asterisk, připojení do běžných telefonních sítí (např. GSM, PSTN) a značné možnosti při vytváření plánů vytáčení. V nových verzích bylo zavedeno například použití protokolu SRTP nebo propojení s produkty jako je Exchange server.

Základní správu číselných plánů by měl zvládnout snad každý administrátor, a to v textovém režimu. Jsou dostupné moduly, které umožňují nastavení i přes webové rozhraní, ne vždy je to ale rychlejší, přehlednější a pohodlnější. Problémem může být i bezpečnostní chyba v takovémto modulu.

S oblibou uživatelů nasazovat tuto ústřednu roste i snaha najít chyby a zneužít je za účelem pouze provozovatele poškodit nebo se obohatit na jeho úkor. I proto jsem Asterisk vybral jako hlavní testovací ústřednu a to instalovanou z repozitáře v systému Debian Squeeze v základním nastavení. Dále jsem testoval na ústřednách dostupných na doménových adresách hlas.802.cz a iptel.org.

Vytvořený program byl testován na systému Linux Ubuntu 11.04. Jako koncová zařízení jsem využil softwarové telefony Ekiga, Twinkle, SJphone a hardwarový telefon Well 3130IF. Na všech těchto zařízeních probíhalo i testování popsaných programů.

6.4.5 Projekt SPT - srovnání

Dále jsem na uvedených ústřednách vyzkoušel penetrační testy z projektu SPT (SIP Penetration Tests) Cesnetu dostupném na adrese https://pentest.cesnet.cz. Zde mi byl zřízen přístup pro vykoušení webové aplikace SPT, která využívá k testování volně šířené nástroje [8]. Aplikace se zabývá prohledáváním otevřených portů pomocí programu Nmap[20] a existujících účtů pomocí SIPVicious[21]. Dále testuje odolnost proti DoS útoku programy Udpflood[22] a Inviteflood[22]. Poté testuje bezpečnost proti manipulaci při registraci, a to programem Reghijacker[22]. Tato aplikace testuje také odolnosti proti spamu, a to programem SPITFILE, který vytvořil stejný tým jako webovou aplikaci SPT. Uživatel má možnost vybrat a nastavit jeden až všechny čtyři testy na jednu ústřednu. Výsledek je poté zaslán na uživatelem zadanou emailovou adresu v textové podobě.

Můj program má s touto testovanou aplikací společné testování vytvořených účtů a v obou je možné provádět DoS útok (i když můj program na tento útok není cílen). Nevýhodou webové aplikace je nutnost vytvoření účtu, což nelze pro širokou veřejnost ale jen pro instituce spolupracující s Cesnetem. Další nevýhodou je možnost testovat pouze ústředny s veřejnou IP adresou. Například ústřednu pro menší firmu bez veřejné IP adresy, která používá připojení od VoIP operátora, není možné otestovat. Můj program je oproti této

20 http://nmap.org/

21 http://code.google.com/p/sipvicious/

22 http://www.hackingvoip.com/sec_tools.html

aplikaci volně dostupný i s kódem. Je možné s ním testovat kterékoliv zařízení na internetu i v místní síti.

Další program, kterým jsem se nechal inspirovat, je již popsaný program SIPp. Jak bylo zmíněno, dokáže procházet scénáře zaslaných zpráv a očekávaných reakcí, které si uživatel předem vytvoří. Můj program toto neumí a je zde velké pole pro vylepšení. Možnou nevýhodou může být, že zpráva, která se má podle scénáře odeslat, je napsána v jazyku XML a je plná proměnných, takže je složitější se naučit takový scénář napsat. Můj program sice umí pouze zprávu odeslat a zaznamenat příchozí odpověď, ale zase využívá toho, že protokol SIP je textový protokol. Tedy zprávu napíšeme jako text. Tento text můžeme napsat podle vlastních vědomostí, stáhnout z internetu a nebo použít reálné zprávy odchycené pomoci některého programu, například Wireshark.

Program	Útok	Zdroj
Udpflood Inviteflood	DoS	http://www.hackingvoip.com/sec_tools.html
SIPp	DoS Ověření korektního chování	http://sipp.sourceforge.net/
Erase registrations Add registrations Redirectpoison	Manipulace zpráv	http://www.hackingvoip.com/sec_tools.html
SIPdump SIPcrack	Prolamování hesel	http://voipsa.org/Resources/tools.php
Ettercap	MiTM	http://ettercap.sourceforge.net/
SPT	Vyhledávání existujících účtů Manipulace zpráv DoS Spam	https://pentest.cesnet.cz
sipTool	Vyhledávání existujících účtů Manipulace zpráv MiTM Ověření korektního chování DoS	

Přehled vyzkoušených nástrojů pro penetrační testování

7 Závěr

V oblasti internetové telefonie je protokol SIP pravděpodobně nejoblíbenější a nejrozšířenější. Je také označován jako budoucnost VoIP telefonie. I přesto, že novější norma byla vydána již před deseti roky, až v posledních několika letech zažívá SIP masivní rozšíření i mezi běžnými uživateli a malými firmami.

V mé práci jsem se zabýval popisem základních principů protokolu SIP. Po nastudování těchto základů jsem hledal a studoval známé útoky na tento signalizační protokol, včetně způsobů obrany. Některé z nich jsem pomocí volně dostupných programů vyzkoušel.

Na základě zkušeností s některými programy jsem navrhl vlastní program pro penetrační testování, který není zaměřený pouze na jeden typ útoku. Vytvořený program umí ověřit odolnost proti zasílání zpráv třetí strany, prohledávat vytvořené účty na ústřednách a testovat korektnost chování po zaslání správné nebo deformované zprávy SIP.

Jak jsem sám testováním vytvořeného programu zjistil, v praxi se provozovatelé VoIP telefonie (hlas.802.cz, iptel.org) snaží útočníkům znemožnit napadení účtů na ústředně. A to například tím, že chráněné ústředny na zprávu REGISTER od jakéhokoliv uživatele vždy odpoví, že uživatel existuje a požadují autorizaci. Nelze tedy zjistit, kteří uživatelé mají na ústředně opravdu vytvořený účet. Ústředny také odpovídají na tuto zprávu pouze jedenkrát za vteřinu, což značně zpomaluje například pokus o uhodnutí hesla.

Při dalším vývoji by mohlo dojít k rozšíření počtu útoků, které by program uměl provádět, a také vylepšení stávajících útoku jejich zautomatizováním. Především možnost provést vybrané útoky jedinou konfigurací (jako u projektu SPT [8]), nenechávat vyhodnocení úspěšnosti na uživateli podle výstupu programu nebo možnost dalšího zaslání předem vytvořené zprávy na základě přijaté odpovědi.

V dnešní době se již používají protokoly, které zabrání odposlechnutí hovoru. Zneužití volání a tím finanční poškození určité osoby je ale stále aktuální. Volně dostupný prostředek pro komplexní testování zahrnující několik základních útoků však stále chybí.

8 Literatura

[1] COLLINS, D. *Carrier grade voice over IP. Second edition.* New York: McGraw-Hill, 2003, 522 s. ISBN 00-714-0634-4.

[2] ROSENBERG, J., SCHULZRINNE, H., CAMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M., SCHOOLER, E. *SIP: Session Initation Protocol*, RFC 3261, June 2002.

[3] SCHULZRINNE, H., CANSER, S., FREDERICK, R., JACOBSON, V. *RTP: A Transport Protocol for Real-Time Applications*, July 2003.

[4] WWW stránky: *Voice over IP Security Alliance (VOIPSA)*, URL http://www.voipsa.org, květen 2012.

[5] DOČKAL, J., MALINA, R., MARKL, J., VANĚK, T. *Bezpečnost internetové telefonie*. *DSM*. 2006, roč. 2006, č. 6, s. 36-42. Dostupné z:http://www.nextsoft.cz/~malina/cs/articles/voip/clanek_Bezpecnost.pdf, květen 2012.

[6] ENDLER, D., COLLIER, M. D. *Hacking exposed VoIP*: *voice over IP security secrets & solutions*. New York: McGraw-Hill, 2007, 539 s. ISBN 978-0-07-226364-0.

[7] WWW stránky: Asterisk – The Open Source Telephony Projects, URL http://www.asterisk.org/ květen 2012.

[8] ŘEZÁČ, F. *Penetrační testy v IP telefonii*. [online]. [cit. 6.5.2012]. Dostupné z: www.cesnet.cz/akce/2011/multimedia/p/Rezac.pdf

Seznam zkratek

ARP	Address Resolution Protocol
CRLF	Carriage return line feed
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IAX	Inter-Asterisk eXchange
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security
ISDN	Integrated Services Digital Network
ISO-OSI	International Organization for Standartization – Open Systems Interconnection
MAC	Media Access Control
MD5	Message-Digest Algorithm
MGCP	Media Gateway Control Protocol
PSTN	Public Switch Telephone Networks
RFC	Request for Comments
RTCP	Real-time Transfer Control Protocol
RTP	Real-time Transfer Protocol
SCCP	Skinny Call Control Protocol
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SIP	Session Initation Protocol
SMIME	Secure Multipurpose Internet Mail Extensions
SRTCP	Secure Real-time Transfer Control Protocol
SRTP	Secure Real-time Transfer Protocol
TCP	Transmission Control Protocol
TFTP	Trivial File Transport Protocol
TLS	Transport Layer Security
TTL	Time To Live
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
UTF-8	Universal Character Set Transformation Format – 8-bit
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
XML	Extensible Markup Language

Manuál

Použití programu sipTool:

1) Zahození hovoru:

```
sudo ./sipTool -a <číslo útoku> -i <interface> -f <filtr> -w <soubor>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-f – pro filtrování odposlouchávaných dat je možné definovat filtr například jako u programu TCPdump, pouze mezery je nutné nahradit podtržítkem „_“

-w – soubor pro záznam výstupu programu

Příklad:

```
sudo ./sipTool -a 1 -i wlan0
```

2) Man in The Middle:

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -w <soubor>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu ústředny, za kterou se vydáváme a na kterou data zasíláme

-w – soubor pro záznam výstupu programu

Příklad:

```
./sipTool -a 2 -i wlan0 -s 192.168.1.5
```

3) Zaslání REGISTER :

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -e <první číslo>
-E <poslední číslo> -S <prodleva> -u <soubor> -v <detailnější výpis>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu, kam bude zpráva zaslána

-e – první číslo z rozmezí. Pokud je použit pouze tento přepínač, bude zpráva zaslána pouze na toto jediné číslo

-E – poslední číslo z rozmezí

-u – soubor s uživateli, kterým bude zpráva zaslána

-s – prodleva mezi zasláním zpráv. Zadává se v milisekundách (1s = 1000ms)

-v – detailnější výpis. Vypíše číselný i řetězcový název všech přijatých odpovědí a na závěr vypíše seznam uživatelů, kteří existují.

Příklad:

```
./sipTool -a 3 -i wlan0 -s 192.168.1.5 -u examples/users
```

4) Zaslání libovolné zprávy :

```
./sipTool -a <číslo útoku> -i <interface> -s <adresa> -e <první číslo> -E
<poslední číslo> -S <prodleva> -u <soubor> -m <soubor> -v <detailnější výpis>
```

-a – určuje typ útoku

-i – použité síťové zařízení

-s – určuje adresu, kam bude zpráva zaslána

-e – první číslo z rozmezí. Pokud je použit pouze tento přepínač, bude zpráva zaslána pouze na toto jediné číslo

-E – poslední číslo z rozmezí

-u – soubor s uživateli, kterým bude zpráva zaslána

-m – soubor, kde je napsána zpráva nebo zprávy

-s – prodleva mezi zasláním zpráv. Zadává se v milisekundách (1s = 1000ms)

-v – detailnější výpis. Vypíše všechny přijaté zprávy se všemi položkami.

Příklad:

```
./sipTool -a 4 -i wlan0 -s 192.168.1.5 -e 200 -E 202 -m examples/messages
```

Obsah CD

bakalarska_prace/bakalarka.odt	- zdrojový tvar písemné zprávy
bakalarska_prace/bakalarka.pdf	- písemná zpráva ve formátu PDF
sipTool/ functions.cpp	- zdrojový kód často používaných funkcí
sipTool/ functions.h	
sipTool/ main.cpp	- hlavní program
sipTool/Makefile	- Makefile
sipTool/ mitm.cpp	- modul MiTM
sipTool/ mitm.h	
sipTool/ sendmsg.cpp	- modul zaslání libovolné zprávy ze souboru
sipTool/ sendmsg.h	
sipTool/ sendreg.cpp	- modul zaslání REGISTER
sipTool/ sendreg.h	
sipTool/ teardown.cpp	- modul zahození hovoru
sipTool/ teardown.h	
sipTool/examples/messages	- ukázkový soubor se zprávami
sipTool/examples/users	- ukázkový soubor s uživately