ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

Testování odolnosti SIP PBX

leden 2015

Diplomant: Bc. Ondřej Kovář

Vedoucí práce: Ing. Ján Kučerák

Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 5. 1. 2015

……

podpis diplomanta

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Kovář Ondřej**

Studijní program:
Obor: Sítě elektronických komunikací

Název tématu: **Testování odolnosti SIP PBX**

Pokyny pro vypracování:

Navrhňete a realizujte testovací prostředí pro simulaci provozu SIP pobočkové ústředny za účelem otestování limitů její zatížitelnosti a odolnosti proti útokům. Zpracujte metodiku testování vhodnou pro různé třídy pobočkových řešení. Otestujte pomocí navržené metodiky SIP část konkrétní instalace Avaya Aura, zaměřte se na hodnocení schopnosti instalace plnit alespoň částečně běžné úkoly i během probíhajícího útoku.

Seznam odborné literatury:

[1] Schulzrinne, H. et. al. RTP: A Transport Protocol for Real-Time Applications. RFC3550. July 2003.
[2] Rosenberg, J. et. al. SIP: Session Initiation Protocol. RFC3261. June 2002
[3] Schulzrinne, H. et. al.: Real Time Streaming Protocol (RTSP), RFC2326. April 1998.

Vedoucí: Ing. Ján Kučerák

Platnost zadání: do konce zimního semestru 2014/2015

prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 11. 2013

Poděkování

Chtěl bych poděkovat svému vedoucímu, panu Ing.Jánu Kučerákovi, že mi pomohl objasnit problematiku VoIP komunikace, za jeho cenné rady a za trpělivost při vypracovávání diplomové práce. Dále bych chtěl poděkovat rodině.

Datum: 5. 1. 2015

……

podpis diplomanta

Obsah

Anotace:

Cílem této práce je analýza možností zabezpečení VoIP (*Voice Over IP*) a následná praktická zkouška bezpečnosti a nastavení softwarové ústředny v závislosti na typu útoku. Teoretická část je rozdělena na tři části, kde první bude pojednávat o VoIP obecně. Druhá část popisuje VoIP řešení konkrétního výrobce (Avaya). V třetí části je popsán útok DoS generovaný za použití nástroje SIPp. Cílem praktické části bude zhodnocení provozu ústředny za běžného provozu a její funkčnost při útoku DoS.

Klíčová slova: VoIP, SIP, RTP, Avaya Aura, Sessiona Manager, Session Border Controller, SIPp DoS, flooding

Summary:

The aim of this study is analysis of possibilities of security of Voip (*Voip Over IP*) followed by practical test of safety and settings of PBX software depending on type of attack. Theoretical part is divided into three parts. The first part will deal of VoIP itself. Second part describe Avaya components. Final part of theoretical analysis will foreshadow method of attack DoS with using SIPp generator. The aim of practical part is to evaluate the PBX operation during normal operations and its functionality in DoS attack.

Index Terms: VoIP, SIP, RTP, Avaya Aura, Session Manager, Session Border Controller, SIPp, DoS, flooding

1. Úvod

Práce Testování odolnosti SIP PBX pojednává o problematice zabezpečení pobočkových ústředen PBX. Práce je věnována především DoS útokům aplikovaných na pobočkové ústředny založených na VoIP komunikaci. Teoretický rozbor protokolu VoIP a útoků je popsán v kapitole 2 respektive 4. Záměrem práce se navrhnutí testovací metodiky pro pobočkové ústředny PBX, které slouží ke zjištění bezpečnostních mezer díky nímž se může navrhnout lepší a bezpečnější řešení. Důležitou částí práce je otestování SIP řešení konkrétního výrobce. V našem případě bude testování řešení Avaya Aura®, které je popsáno v kapitole 3. Testování architektury Avaya Aura® je následně popsáno v kapitole 5. Výstupem práce bude návrh řešení testování pobočkových ústředen, který může sloužit jako návod pro otestování vlastní pobočkové ústředny PBX.

2. VoIP

V posledních letech zaznamenalo telekomunikační odvětví velký pokrok. S rozvojem datových sítí a zejména sítě Internet se telefonní komunikace setkala s velkou popularitou jednak využitím v podnikatelském sektoru tak i u veřejnosti. Pro komunikaci VoIP (*Voice Over IP*) jsou využívány stávající datové sítě. Použití stávajících sítí snížilo náklady na propojení a zřízení telekomunikační sítě. Přispěly k ušetření nákladů v provozu, kdy nejsou využívány přepojované okruhy ale datové sítě. VoIP technologie byla vytvořena jako alternativa k PSTN (*Public Switched Telephone Network*) zahrnující nynější telekomunikační síť. Využívání VoIP přináší nové možnosti v komunikaci. Nabízí využívání konferenčních hovorů, video hovorů a v neposlední řadě velice oblíbený Instant Messenging. Bohužel daň za využívání datových sítí je vybrána v oblasti zabezpečení. Vzhledem k užívání sítí zařazených do globální sítě Internet, je zde pravděpodobnost bezpečnostních rizik. Jelikož VoIP síť je založena na sítích IP, vztahují se na VoIP telekomunikaci klasické útoky, se kterými jsme se mohli setkat pouze u datových sítí donedávna využívaných pouze k ryze datovým přenosům. K zabezpečení VoIP komunikace existují mnohé postupy, avšak řada z nich je velice nákladná nebo složitá k aplikaci. Z důvodu bezpečnostního rizika nebyly VoIP tolik využívány, ale postupem času se dostáváme do fáze, kdy pomalu ale jistě vyzařují klasické PSTN využívající uzavřených sítí s přepojovanými okruhy.

Jak už bylo napsáno, VoIP umožňuje přenášení zdigitalizovaného hlasu za využití datových sítí. Mezi ně patří např. síť Internet, menší uzavřená firemní síť, Intranet nebo jiná datová spojení. Digitální informace, přenos hlasu nebo videa, je přenášena v reálném čase. Využívány jsem různé komunikační a signalizační protokoly. Dva nejznámější a nejpoužívanější jsou H.323 a SIP.

2.1. VoIP protokoly

Protokoly VoIP jsou uzpůsobeny k realizaci přenosu hlasu za použití IP sítí. Realizována je telefonie a hlasová komunikace sítí Internetového Protokolu (IP), fungující na principu

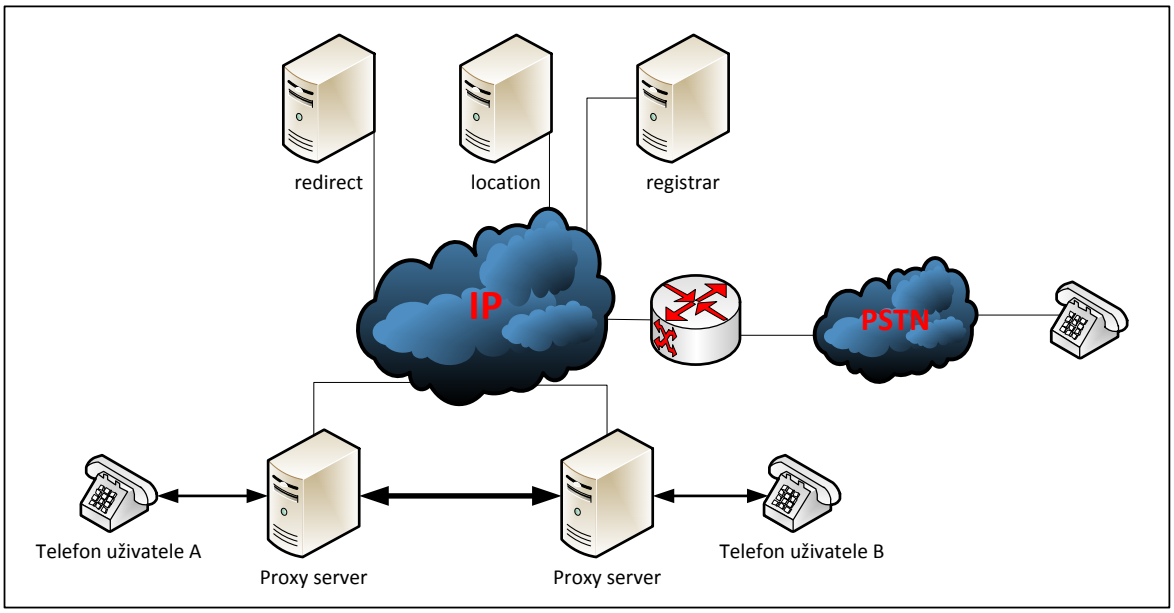
přenosu dat v podobě datagramů. Existují dva typy VoIP protokolů. První skupina, signalizační protokoly, starající se o inicializaci, správu, vedení a ukončování hovoru. Do skupiny signalizačních protokolu patří dva neznámější z nich, H.323 a SIP, které jsou používány v drtivé většině VoIP zařízeních umožňující přenos hlasu v IP síti. Druhou skupinou protokolů VoIP jsou protokoly komunikační. V souvislosti s přenosem hlasu je dobré zmínit protokol RTP (*Real-time Transport Protocol*), patřící mezi komunikační protokoly, který umožňuje právě přenos mediální informace.

2.1.1. SIP

Protokol SIP (*Session Initiation Protocol*) je signalizační protokol textového formátu starající se o správu telefonního hovoru VoIP (zahájení, vedení a ukončení hovoru). Vzhledem k tomu, že SIP protokol je textový protokol, má podobné vlastnosti jako SMTP (*Simple Mail Transfer Protokol*) nebo HTTP (*HyperText Transport Protocol*). Struktura protokolu SIP: hlavička, obsahující informace potřebné ke správnému doručení. Hlavička SIP zprávy by měla obsahovat [1]:

- To: Adresa volaného
- From: Adresa volajícího
- Via: Cesta, kudy prošel požadavek a kudy se bude vracet odpověď
- Call-Id: Identifikace volání
- Contact: Aktuální skutečná adresa klienta
- Record-Route: Databáze adres serverů dostávající veškerou komunikaci náležící k hovoru
- Request-URI (Uniform Resource Identifier): Aktuální adresát požadavku

Pro přenos zpráv SIP prokolou v IP síti se využívají protokoly UDP (*User Datagram Protocol*), TCP (*Transmission Control Protocol*) nebo TLS (*Transport Layer Security*), protokol, který zajišťuje zabezpečený přenos. Schéma funkčnosti volání založeném na protokolu SIP je vyobrazeno na obrázku 2.1.



Obr. 2.1: Architektura sítě SIP

Struktura sítě pracující se SIP protokolem je:

- UAC (User Agent Client) - terminálový klient, starající se o započetí SIP.
- UAS (User Agent Sever) - terminálový server odpovídající na signalizaci, která přichází z UAS.
- UA (User Agent) - koncové zařízení umožňující generovat a ukončovat SIP žádosti. Jedná se například o IP telefon, analogový telefon, softphone[1] nebo instant messenger[2] (IM).
- SIP Proxy - přijímá žádosti od UA nebo jiných proxy serverů a následně vytváří nová spojení. Jestliže není volaný účastník ve stejné doméně, dojde k posunutí zprávy na další proxy server. Zpráva se přeposílá do té doby, dokud nedojde k nalezení volaného účastníka.
- SIP Redirect Server - přijímá žádosti jiných SIP zařízeních nebo proxy serverů a následně posílají zpět odpověď se zprávou s místem zaměření žádosti.
- Location Server - databázový server uchovávající umístění klientů a dalších SIP proxy serverů.
- Registrar server - databáze připojených koncových zařízeních, aktualizující se žádostmi s registrací od UA.

1 Softphone je odvozen od spojení software a phone (telefon) a jedná se tedy o softwarové řešení SIP telefonu. Softphone je v podobě klientů např. X-Lite, DialX, Zoiper, a pro všechny známý Skype, ale ten není typickým řešením SIP.

2 Instant messenger slouží ke komunikaci v reálném čase, kdy výměna zpráv probíhá v řádech milisekund (v závislosti na kvalitě internetového připojení) od momentu odeslání. U nových klientů se setkává s pokročilejšími funkce obsahující např. posílání souborů, sdílení vzdálené plochy nebo videohovory.

Během hovoru dochází ke stálým výměnám požadavků a odpovědí mezi klientem a serverem (SIP protokol je typu klient-server, kdy se tyto role střídají). Textový charakter protokolu napomáhá jednoduché implementaci protokolu a rozšíření. [1] Princip procesu navazování, vedení a ukončení hovoru je znázorněn na obrázku 2.2 a obrázku 2.3.

Obr. 2.2: Navazování spojení v SIP [2]

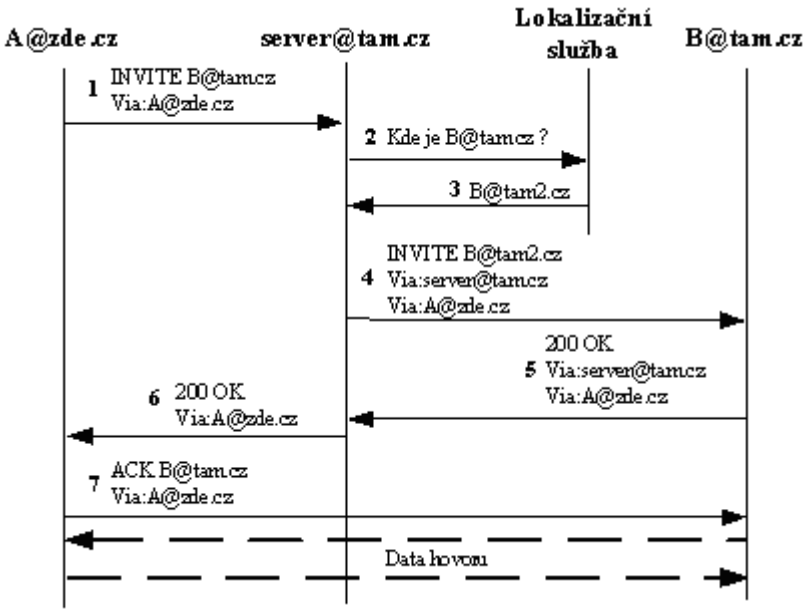
Komunikace protokolu SIP je realizována výměnou SIP zpráv tvořících záznamy o proběhlých relacích. Jedná se vlastně o transakční protokol starající se o doručování finálních odpovědí, potvrzovacích nebo chybových hlášeních v závislosti na typu požadavku a stavu komunikace. Záznam o spojeních skrývá několik podružných procesů, které spojení předchází. Zprávy a hlášení jsou posílány v textové podobě a tudíž je snadné jejich procházení, aniž bychom potřebovali využít analyzátoru nebo jiných nadstandardních aplikací. Zpráva je složena z požadavku (REQUEST) a hlavičky. Nejznámější požadavky jsou:

- INVITE – iniciátor spojení na základě požadavku o spojení
- CANCEL – ukončení relace v rámci INVITE žádosti
- ACK – potvrzení o zahájení relace (použití v rámci INVITE žádosti)
- REGISTER – registrování nebo zrušení registrace adresy telefonu nebo jiného SIP zařízení
- BYE – ukončení relace
- SUBSCRIBE – získání aktuálního stavu
- NOTIFY – upozornění na zpracování metody SUBSCRIBE obsahující informace o stavu
- OPTIONS – sondování ostatních SIP uživatelů a serverů, pro zjišťování podpory, stavu ostatních UA (User Agent) a možností volajícího

S REQUEST žádostmi souvisí stavové kódy, které charakterizují odpovědi na REQUEST žádosti. Použití je stejné jako u protokolu HTTP. Hlášení je udáváno v závislosti na typu odpovědi:

- 1xx – informační zprávy
- 2xx – úspěch (OK)
- 3xx – přesměrování (zaslání požadavku na místo)
- 4xx – chyba klienta (žádost klienta nelze zpracovat)
- 5xx – chyba serveru (chyba na straně příjemce)
- 6xx – globální selhání (žádost nelze provézt)

Názorná ukázka principu komunikace SIP protokolu je zobrazena na obrázku 2.3. Příprava na telefonní hovor začíná posláním požadavku INVITE (1) serveru SIP, který odpovídá adrese volaného. Potom se server pokouší zjistit, kde se volaný nachází (2). Většinou je poloha volaného obsažena databází kontaktů. Server SIP se může příslušné databáze dotázat na adresu volaného. Jakmile je poloha klienta B obdržena (3), zasílá server SIP tuto informaci zpět klientovy A. V tomto případě je server v režimu *redirect*. Klient A buď může zahájit spojení nebo server SIP upraví zprávu INVITE a přepošle ji klientovy B (4). V případě, že je zpráva přeposlána klientovy B, jedná se o proxy mód. Hláškou 200 OK, která projde přes server a následně k volajícímu, dojde k povolení hovoru. Následně proběhne potvrzovací ACK a další přenos zpráv probíhá mezi klienty A a B napřímo (7).



Obr. 2.3: Komunikace v SIP po úspěšné registraci [2]

2.1.2. H.323

Protokol H.323, neboli norma pro komunikaci, byla vytvořena organizací ITU-T (*Telecommunication Standardization Sector of ITU*). Definuje procedury, protokoly a komponenty důležité a nezbytné k uskutečnění telefonního hovoru nebo započetí vizuální komunikace v rámci lokální sítě. Umožňuje přenos informace v reálném čase (*real-time* komunikace). Lze přenášet jak audio tak video nebo jiné datové informace. Norma H.323 může být taktéž využita k vícebodové multimediální komunikaci a byla určena pro snadný přechod mezi technologiemi PSTN a VoIP. [3]

Správa protokolu zahrnuje vše od započetí spojení až po jeho ukončení. [4] Bez ohledu na topologii sítě jsou H.323 terminály schopny komunikovat v rámci celé sítě za použití přepínačů, rozbočovačů a mostů. Nejhojnějšího využití nachází protokol H.323 v Internetu, kde je používán výhradně k telefonii a videotelefonii.[5] Využívání protokolu H.323 se však snižuje kvůli nástupu protokolu SIP, který je jednodušší pro vývojáře a jeho implementace do systému je podstatně snadnější než je tomu u H.323. Zjednodušení implementace H.323 mělo pomoci zavedení open source implementace protokolu OpenH.323. [6]

2.1.3. RTP

Real-time Transport Protocol (RTP) je protokol, který se používá k samotnému přenášení jednak hlasu, ale i videa, ke koncovým uživatelům. Jedná se o *real-time* protokol a dochází tedy k přenosu dat v reálném čase. Nejčastěji je přenášeno audio a video. Přenos dat v reálném čase je přenášen za využití procedur, které protokol RTP obsahuje. RTP protokol využívá služeb, které nabízejí určování nesoucích dat, číslování sekvencí, sledování přenosu a časové známkování (*timestamping*). RTP protokol navíc podporuje *multiplexing* a protokol RTCP (*RTP Control Protocol*), poskytující kontrolní služby, které mají zajistit určitou kvalitu poskytovaných služeb, identifikaci odhadu velikosti relace a s tím spojenou kontrolu proti zahlcení. Pomocí podpůrného protokolu RTCP jsou přenášeny statické informace o přenosu RTP nesoucí informace o kvalitě, zpoždění přenosu a *jitteru*. Nejčastěji používanými protokoly pro přenos informace jsou protokoly UDP a TCP, které nemají pevně definované komunikační porty. Omezující faktor je dynamické přidělování portů, který způsobuje špatnou prostupnost přes firewall nebo NAT[3]. RTP protokol využívá *multicasting,* čímž může zasílat data do více koncových bodů za předpokladu, že je *multicasting* podporován sítí. Vše o RTP protokolu je definováno v RFC3550. [21]

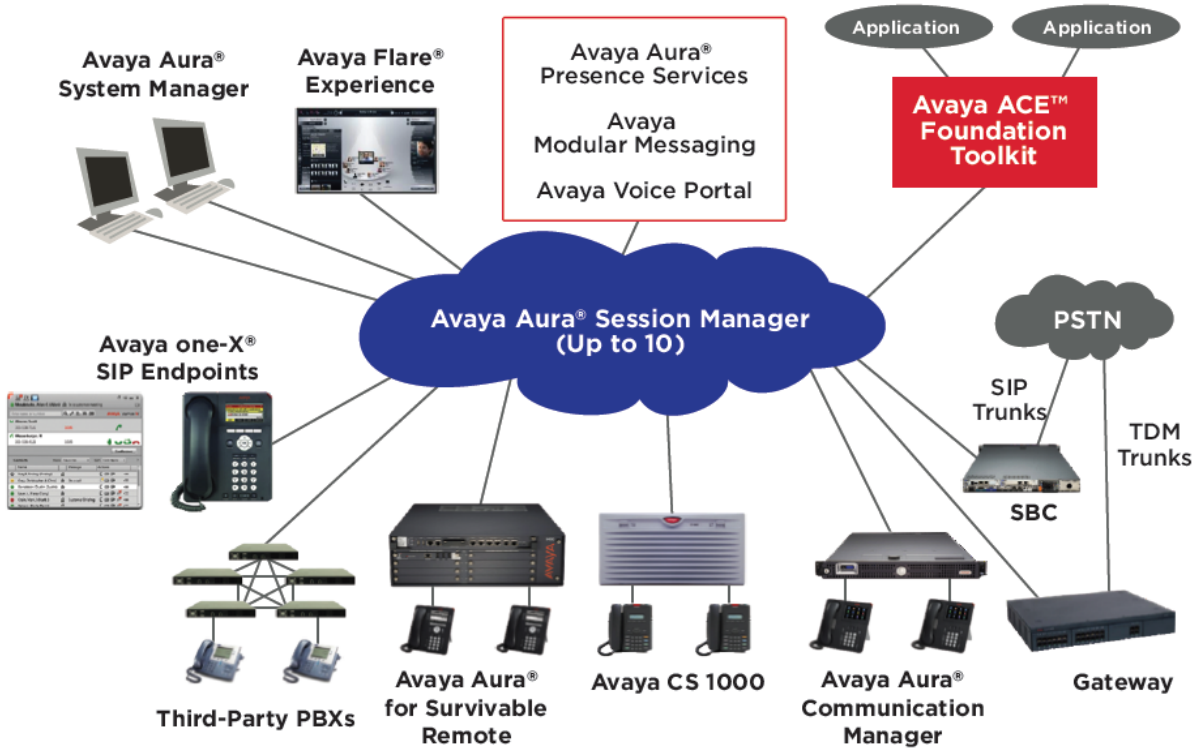
[3] U použití NAT dochází k problému s prostupností skrze NAT. To je vyřešeno síťovým protokolem STUN (Simple Traversal of UDP though NATs), který je nejčastěji umístěn na stejné zařízení monitorující NAT (jeho veřejnou adresu a číslo portu). Klient má tak dostatek informací potřebných k snazšímu projití přes server NAT.

3. Avaya Aura

Avaya Aura® je sada produktů určených ke komunikaci, která se snaží utvořit pro firmy unifikovaný komunikační nástroj. Platforma Avaya Aura® je jádrem komunikační platformy přinášející celé společnosti, mezilidské spolupráci podporu plně unifikovaných komunikačních a kontaktních center od středních až po velké společnosti. Platforma Avaya Aura zlepšuje SIP architekturu, zjednodušuje komplexní komunikační sítě, redukuje náklady na infrastrukturu a rychle doručuje zvuk, video, zprávy a webové aplikace uživatelům ze všech míst nacházejících se kdekoliv.

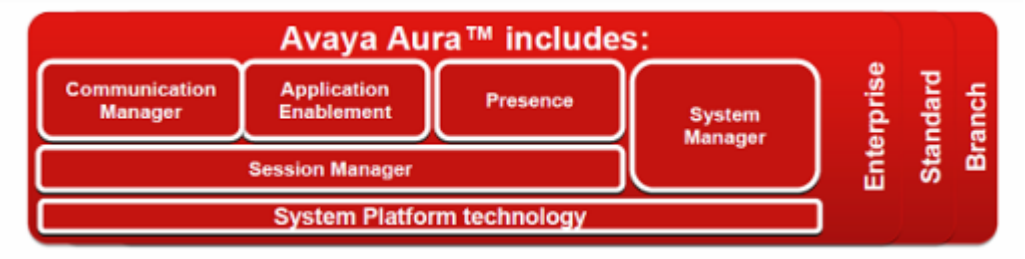
Avaya Aura® nabízí plnou podporu třetích stran. Jednotná komunikace, možnosti kontaktního centra a spolupráce Avaya Aura® může být přístupna řešením třetí strany pomocí sad standardů API (*Application Programming Interface*) a Webové podpory poskytované řešeními Avaya Aura® Collaboration Environment, Avaya Agile Communication a Avaya Aura® Application Enablement Services.

Platforma Avaya Aura® se skládá z těchto pěti částí (serverů): Communication Manager, Application Enablement, Presence Service, System Manager a Session Manager. Topologie architektury Avaya Aura® je znázorněna na obrázku 3.1. [27]



Obr. 3.1: Topologie architektury Avaya Aura® [27]

Schéma částí Avaya Aura® je znázorněno níže na obrázku 3.2.



Obr. 3.2: Diagram částí obsahující Avaya Aura® [27]

3.1. Session Manager

Session Manager je mozkem a nervovým systémem komunikačních systémů Avaya Aura® pro podniky. Úlohou Session Manageru je směrování SIP protokolu. Nabízí lepší schopnost centralizovaného řízení a výrazné zlepšení rozsahu a redundance, které umožňují zefektivnění nákladů a lepší distribuci do podnikového segmentu. Avaya Aura® Session Manager vylepšuje infrastrukturu existujících PBX, zajišťující pokrok vpřed, chránící a ubezpečující investice do dnešních Avaya systémů a softwarů. [22] [26]

Session Manager nabízí:

- Nižší celkové náklady.
- Centralizovanou infrastrukturu a management.
- Snížení provozních nákladů se samostatným směrováním a kontrolním dial plánem.
- Unifikovanou architekturu kombinující to nejlepší z Avaya Communication Server 1000 a Avaya Aura® Communication Manager.
- Integraci zařízení třetích stran.
- Náhradu a upgrade starších Network Routing Serverů (NRS).
- Rychlé nasazení dílčích aplikací bez nutné úpravy stávající PBX.

3.1.1. Communication Manager

Communication Manager je klíčovou komponentou struktury Avaya Aura® nabízející telefonii, mobilní funkce, integrovaný *messaging*, integrovaná telekomunikační a hlasová centra a nabízející integraci aplikačních funkcí. Avaya Aura® Communication Manager je otevřená, plně rozšiřitelná platforma IP telefonie, která může být využita buď jako IP PBX, server podporující pouze služby SIP nebo server podporující jak SIP služby, tak služby mimo SIP. Vyniká vysokou spolehlivostí a velmi dobrou rozšiřitelností a poskytuje podporu pro více protokolů, nabízí více jak 700 funkcí a pokročilé funkce pro produktivitu a mobilitu. Součástí jsou aplikace konferenčních a kontaktních center obsažených v systému. Organizuje a zabezpečuje přenos hlasu, dat, obrázků a videa. Může se připojit k soukromé i veřejné

telefonní sítě, do sítě Internet a Intranet. Umožňuje spravovat distribuované sítě, servery, a serverové brány. Navíc je podporována široká škála analogových a digitálních zařízeních, a dalších zařízeních IP komunikace. [28]

Communication Manager si klade za cíl splnění obchodních problémů s optimalizací hlasové komunikace a integrací aplikací s vysokou přidanou hodnotou. Communication Manager poskytuje funkce pro správu uživatelů a systému, inteligentní směrování hovorů, integraci aplikací s možností rozšíření a propojení firemní komunikace. [28]

Communication Manager poskytuje následující funkce virtuálního podniku:

- Výkonné zpracování hlasových a video hovorů.
- Zvýšení produktivity zaměstnanců a sofistikovaných mobilních funkcí.
- Integrované aplikace pro kontaktní a konferenční centra.
- Centralisované operace pro hlasové zprávy.
- Široká škála analogových, digitálních a IP zařízeních.
- Podpora protokolů H.323, SIP a mnoho dalších standartních komunikačních protokolů.
- Vysoká dostupnost, spolehlivost a nezávislost.

3.1.2. Application Enablement

Avaya Aura® Application Enablement Services poskytuje rozšíření v podobě uživatelsky jednoduchého rozhraní se stovkami jak integrovaných komunikačních a podnikových aplikací, např. Microsoft Office Communicator a IBM Lotus Sametime, tak širokou škálu kontaktních center, hlasové nahrávání a aplikace *click-to-dial*[4].

3.1.3. Presence Service

Presence Service je základním prvkem pro sjednocení komunikace. Má za funkci poskytovat detailní informace o dostupnosti daného uživatele, jeho aktuální aktivitě, preferovaném způsobu komunikace a pozici, umožňující uživatelům spojit se s tím správným klientem ve správný čas a správným komunikačním prostředkem. Navíc tyto informace jsou doručeny tak, aby byly k dispozici nejen uživateli, kterému byly určené, ale také pro aplikace využívající informaci ke kontaktování uživatele v případě kritické komunikace během obchodního procesu.

[4] *Click-to-dial* je nástroj pro tvorbu odkazů na propojení čísel, např. ze sítě Internet, s klientským telefonem, softonem nebo jiným zařízením podporující volání a zároveň *click-to-dial* funkci. Tato funkce umožňuje okamžité volání při kliknutí na odkaz vytvořený z čísla.

3.2. System Manager

Avaya Aura® System Manager je jedna z klíčových komponent architektury Avaya Aura. Avaya Aura® poskytuje bohaté schopnosti v oblasti hlasové a video komunikace, poskytuje odolnou, pružnou a distribuovanou síť pro analogové, digitální a na IP založená zařízení. System Manager je konfigurační bránou pro všechny služby, kde se dají nastavovat a spravovat skupiny, uživatelé, dostupná rozšíření a změny preferencí. Avaya Aura® System Manager je intuitivní nástroj pro správu a řízení, který přináší lepší možnosti do běžného života. [29]

System Manager nabízí integrované řešení, které využívá uživatelská data a plánování napříč strukturou Avaya Aura® pomáhající podnikům implementovat a spravovat optimální výsledky pro vlastní řešení komunikace a obchodní spolupráce. [29]

Možnosti řízení a správy: [29]

- Zajištění jednoduché správy a programovacího rozhraní aplikací systému Avaya Aura®
- Jednoduchá, intuitivní a konzistentní.
- Umožňuje zadat uživatelská data pouze jedenkrát a ty jsou následně sdílena v rámci všech aplikací Avaya Aura®.
- Pevná integrace do podnikové IT struktury zahrnující správu identity, zabezpečení a podnikové adresáře.

3.3. Session Border Controller

Avaya Aura® Session Border Controller, vyvíjen ve spolupráci Acme Packet, zabezpečuje IP síť pro *real-time* interaktivní komunikaci proudící mimo vnitřní síť intranet a slouží k zabezpečení SIP *trunků* na hranici veřejné sítě. Funkce Avaya Session Border Controller for Enterprises je kompletní zabezpečení architektury na úrovni aplikační vrstvy v jediném zařízení. Nabízí SIP firewall, Session Border Controller, Intrusion Detection System a Intrusion Prention System (IDS/IPS), Access Controller, Authentication, Unified Communication Proxy a Policy Enforcement pro všechny aplikace unifikované komunikace v reálném čase. [30]

Avaya Aura® Session Border Controller je standartním elementem Avaya Aura® komunikační architektury nabízející:

- Podporu zabezpečených relací SIP.
- Ochranu signalizace SIP proti bezpečnostním hrozbám a přetížení.
- Eliminování signalizace a problémů spojených s interoperabilitou.
- Zachování kvality relací v zatíženém režimu a za nepříznivých podmínek.
- Podporu dodržování předpisů.

Tři důvody nasazení SBC v síti:

1. **Real time IP komunikace vyžaduje rozdílné bezpečnostní služby.**
    - *Real-time* komunikace je kontinuální.

- U *real-time* komunikace je potřeba zprostředkovat problémy interoperability od různých dodavatelů.

2. **Dnešní řešení zabezpečení zaměřená na data nejsou dostatečně chráněná.**
    - Postrádání schopnosti dynamického řešení potíží s připojením VoIP řešení.
    - Nezpůsobilost provádět hloubkovou kontrolu VoIP signalizace a média.
    - Nezpůsobilost řešení problémů interoperability protokolů.

3. **Bezpečnost především.**
    - Multiprotokolová a *real-time* povaha IP telefonie a unifikovaná komunikace vyžadující sofistikované řešení zabezpečení.
    - Od dob, kdy je jednodušší aplikace signalizačních útoků, je potřeba být připraven s dostatečným řešením zabezpečení.

Zabezpečení sítě založené na SIP protokolu proti:

- Útokům odmítnutí služby (DoS).
- Spoofingu.
- Zcizení telefonního hovoru .
- Toll-fraud blocking.
- Odposlechu a zcizení citlivých informací.

4. Bezpečnost VoIP

Paketově orientované sítě jsou stále více využívány k VoIP telefonování, videotelefonii a jiné multimediální činnosti. Díky textové podstatě protokolu SIP a podobnosti s protokoly SMTP a HTTP je jednodušší jeho implementace a použití ve stávajících sítích. Tím se nám však vyskytuje určité riziko útoků, které byly použitelné k napadení internetových protokolů a tím pádem i signalizačního protokolu SIP. Vzhledem k závažnosti internetového telefonování je nezbytností zabezpečení dat přenášených při VoIP hovoru. Vzhledem k relativně krátké doby nasazení VoIP do „ostrého" provozu, může se ještě vyskytnout několik míst, kde je nedoladěné zabezpečení spojené s inovací technologie. V konverzaci účastníků telefonního hovoru se můžou vyskytovat citlivější data v podobě čísel platebních karet nebo jiných, lehce zneužitelných informací. Z toho důvodu je důležité zabezpečení VoIP systému jako celek a potom i dílčích zařízení jako jsou zařízení koncová, směrovače, rozbočovače a jiné podpůrné systémy související s VoIP komunikací.[8] Zabezpečení signalizačních a komunikačních protokolů by měli vykazovat určitý stupeň utajení, autorizace a integrity. K informaci by měl se zabezpečením, s určitým stupněm autorizace, přistupovat pouze volající nebo volaný. Neposlední podmínkou zabezpečení hovoru je integrita, která zajišťuje, aby byla informace přijata ve stejné podobě, v jaké byla vyslána. [9]

Zavedení VoIP řešení na jednu stranu nabízí různá řešení a přizpůsobení a tím i plynoucí snížení nákladů jak s implementací, tak v řízení a vedení, ale přináší i mnohé zneužitelné bezpečnostní mezery. Problémem VoIP řešení je instalace systému do již vytvořené sítě, která obsahuje různá zabezpečení, které nemusí vždy odpovídat standardům potřebných k bezpečnému provozu VoIP systému. Důležitou součástí zabezpečení telekomunikačního systému je i pravidelný dohled a stálá inovace bezpečnostních prvků jak aktivních, tak i pasivních. Vývoj bezpečnostních prvků a možností je nezbytný, vzhledem k stále se vyvíjejícím možnostem zneužití.

4.1. Cíle útoků

V prostředí SIP můžou být útoky směřovány na dva typy komponent:

SIP server – nejdůležitější komponenta SIP architektury. Útoky jsou často směřovány na SIP server, protože v případě úspěšného útoku dojde k vyřazení celé SIP služby.

SIP klient – útok na SIP klienta není až tak závažný, protože dojde „pouze" k vyřazení koncové stanice, která potom není dostupná. Důsledek není tak fatální jako útok na SIP server.

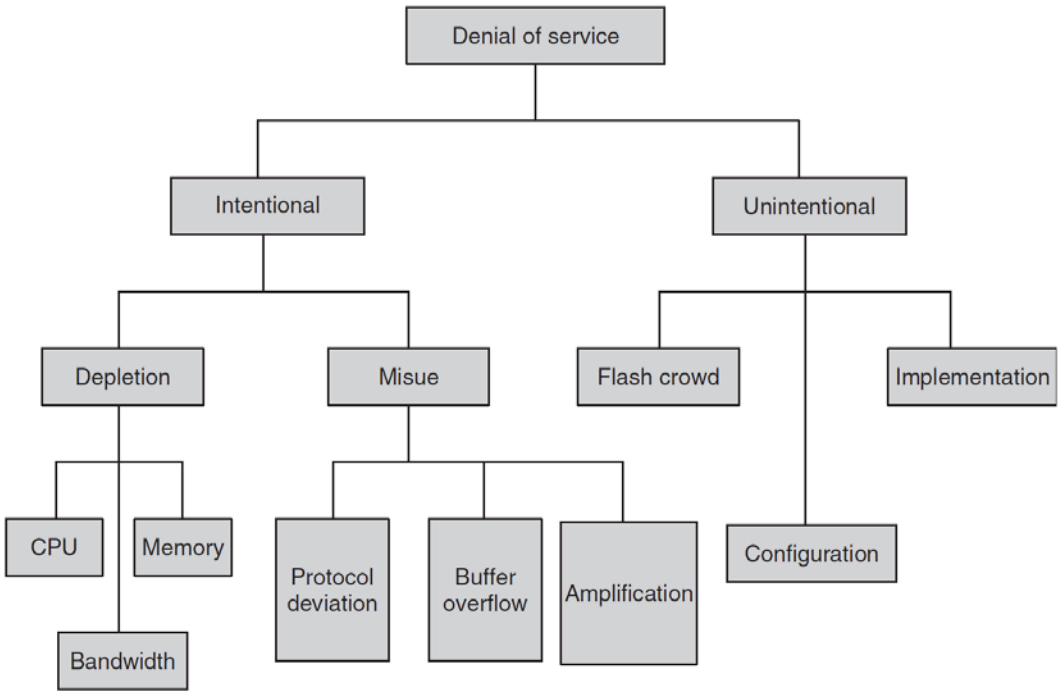
SIP servery jsou nejčastěji umísťovány na rozmezí privátních sítí a sítě Internet z důvodu dostupnosti volání nejen ve vlastní síti, ale i skrze proxy i do jiných sítí v síti Internet. Útočník proto směřuje útoky na SIP server, který je spojený s oběma sítěmi a je důležitější komponentou v SIP prostředí.

4.2. Odepření služby (DoS)

Útoky typu odepření služby, neboli DoS (*Denial of Service*), jsou všechny útoky, které jsou směrovány na účastníky hovoru nebo přímo na ústřednu a jejich záměrem je využití prostředků k odstavení služby nebo určité části sítě. Záměrem útoku je směrování velkého množství bezvýznamných paketů, které vyčerpají jeden nebo více zdrojů hosta, jako například šířku pásma, paměť nebo výpočetní jednotku CPU a tím pádem host nemá dostatek prostředků pro zpracování dalších požadavků. Ve VoIP síti bude útok DoS nejčastěji směrován na server proxy, který zprostředkovává přístup do sítě internet. Jak jsme si řekli, VoIP hovory a tudíž přenos dat je realizován v reálném čase, v určitých případech může dojít důsledkem DoS útoku k ukončení probíhající spojení. DoS útoky jsou prakticky realizovány s využitím vyššího výpočetního výkonu ke generování většího množství podvržených informací.

Rozdělení útoků odepření služby můžeme kategorizovat z různých pohledů. Útoky DoS jsou zaměřené na protokoly zpracovávající funkce na různých vrstvách OSI architektury. Útoky mohou být prováděny na vrstvách transportní, síťové a aplikační. Z toho pohledu můžou být DoS útoky rozděleny podle obrázku 4.1. U toho typu rozdělení rozlišujeme útoky úmyslné, neúmyslné nebo škodlivé. Neúmyslné útoky jsou útoky, které vyplívají z chyb v implementaci nebo konfiguraci a škodlivé útoky jsou záměrně inicializované útočníkem. Úmyslné útoky můžeme dále rozdělit na útoky zaplavením a útoky zneužitím. Útokem zaplavením dojde k vyčerpání nebo snížení jednoho nebo více prostředků (CPU – *Central Processing Unit*, přenosová rychlost, paměť), což činí nemožnost dalšího využívání prostředků pro úkony spojené s provozem služby, příjem a zpracování požadavků nebo přímo vyřazení celé služby z provozu. [12]

Další způsob, jak DoS útoky rozdělit, je podle počtu zdrojů útoku. Buď se jedná o útok s jedním zdrojem nebo distribuovaný útok odepření služby (DDoS – *distributed denial-of-service*) s více zdroji. Typické pro DDoS útok je, že je produkovaný ze zařízení nazvaných *botnets*, sad počítačů sloužících útočníkovi k provedení útoku. Často se jedná o servery a počítače klasických uživatelů nebo firem, ke kterým útoční získá přístup infikací virem. Útočník tak získá kontrolu na více počítačích, které využije k útoku a může využít většího výpočetního výkonu pro generování více paketů potřebných k úspěšnému útoku zaplavením. [13]



Obr. 4.1: Klasifikace DoS útoků [12]

4.2.1. Bandwidth Depletion Attack

Přístupová linka k síti Internet je omezena velikostí přenosového pásma. Velikost může být od několika desítek kilobitů po několik gigabitů za vteřinu v závislosti na typu úkonu prováděném hostem. Právě na přenosové pásmo jsou zaměřené *bandwidth depletion attacks.* Jedná se o útoky zaplavením, které jsou založeny na jednoduchém principu přetěžování serveru posíláním většího počtu paketů, než je server schopný zpracovat. Oblíbeným útokem patřící mezi *flooding attack* je *UDP flood attack. User Datagram Protocol* (UDP) paket může být poslán příjemci bez předchozího spojení a od příjemce je požadován příjem a zpracování onoho paketu. Posláním velkého množství UDP paketů útočník může zahltit linku a vyčerpat celé přenosové pásmo k oběti. Po přijetí je rozhodnuto, zda-li je paket určen pro další aplikaci a přesměrován dále nebo je zahozen. V případě zahození paketu je odesílateli poslána ICMP (*Internet Control Message Protocol)* zpráva. V obou případech se vyžaduje zpracování množství paketů procesorem CPU a je docíleno jeho zatížení. [12]

SIP obvykle používá UDP protokol, což znamená, že server používá exponenciální přeposílání požadavku. Po odeslání požadavku v případě, že odesílatel tohoto požadavku nedostane odpověď od serveru, dojde po určitém čase T1 vteřin k opětovnému odeslání požadavku a dojde ke zvýšení hodnoty *timeout.* Požadavky jsou přeposílány po dobu definované v tabulce 1, dokud nedojde k dovršení maximální hodnoty přeposlaných požadavků. Server v případě přetíženosti stále přijímá konstantní počet záškodných požadavků, provoz stále

narůstá i s počtem požadavků platných uživatelů, kterým po určité době vyprší doba platnosti a server je neobslouží. V případě požadavku INVITE odesílatel vysílá požadavky v exponenciálně narůstajících časových intervalech až do Timer B (tabulka 1). Přeposílání požadavků nastane v časech T1, 3T1, 7T1, 15T1 až do času Timer B. To se dá reprezentovat výrazem:

$$(2^1 - 1)T1, (2^2 - 1)T1, (3^3 - 1)T1, \dots, (2_i^N - 1)T1$$

a to až do doby dokud časovač nedosáhne hodnoty Timer B. [24]

Časovač	Základní hodnota	Význam
T1	500 ms	Roud-trip time (RTT) estimate
T2	4 sec	Maximum retransmission interval for non-INVITE requests and INVITE responses
T4	5 sec	Maximum duration that a message can remain in the network
Timer A Timer B	initially T1 64*T1	INVITE request retransmission interval, for UDP only INVITE transaction timeout timer
Timer D Timer E	>32sec for UDP 0 sec for TCP and SCTP initially T1	Wait time for response retransmissions Non-INVITE request retransmission interval, UDP only
Timer F	64*T1	Non-INVITE transaction timeot timer
Timer G	initially T1	INVITE response retransmission interval
Timer H	64*T1	Wait time for ACK receipt
Timer I	T4 for UDP 0 sec for TCP and SCTP	Wait time for ACK retransmissions
Timer J	64*T1 for UDP 0 sec for TCP and SCTP	Wait time for retrasmissions of non-INVITE requests
Timer K	T4 for UDP 0 sec for TCP and SCTP	Wait time for response retransmissions

Tab. 1: Přehled časů SIP zpráv [24]

Útoky zaplavením jsou běžné v paketové síti, proto našly „využití“ i u SIP protokolů, které jsou založeny právě na paketových sítích. U internetové telefonie budou útoky tohoto typu směrovány s největší pravděpodobností na proxy nebo registrar servery. S ohledem na jednoduchost stavby signalizačního protokolu SIP, který nebyl optimalizovaný na zpracování vysokého počtu zpráv, dojde generováním velkého množství požadavků k vyčerpání zdrojů pro zpracování dalších požadavků a tím pádem k neschopnosti vykonávat bezproblémový základní provoz. Úspěšnost odolání útokům zaplavení je spojená s hardwarovým vybavením serveru. S kvalitnějšími a rychlejšími servery je možnost obsloužit

rychleji a větší množství požadavků a tím pádem posunout moment vyčerpání prostředků pro zpracování požadavků a snížit pravděpodobnost úspěchu útoku. [13] [14] [15] [16]

4.2.1.1. RTP DoS

Předpoklad RTP DoS útoku je v serverech připojených do veřejné sítě, které skrz protokoly SIP a RTSP tvoří velká množství RTP spojení nesoucích mediální data ke koncovým zařízením. V případě protokolu SIP sem patří IVR systémy, telefonní brány, konferenční servery a systémy hlasových schránek. Každé z těchto jmenovaných zařízeních je schopno přijímat velké množství SIP hovorů potažmo RTP spojeních. Není žádný požadavek, že by měl hovor pocházet od ověřených (*authenticated*) zdrojů, proto útok může být spuštěn i na zařízeních požadujících ověření. Většina z výše jmenovaných zařízeních si ověření provádí samostatně. IVR a servery hlasové schránky typicky vyzvou uživatele k zadání kódu PIN nebo přístupového hesla. V těchto případech je však datový tok již uskutečněn a předpoklad útoku je právě v uskutečnění datového toku. Proto ověření, které je na úrovni aplikační vrstvy, nemá na úspěšnost útoku žádný vliv. [20]

Při úspěšném útoku je nemožné uskutečnit na koncovém zařízení hovor, nelze přijímat hovory do hlasové schránky, nefunguje datová podpora, volání na zákaznickou linku a může dojít k přerušení již probíhajících spojení. Další možností zahlcení služby a znemožnění jejího používání je přeplnění hlasové schránky nebo server obsluhující služby poskytující krátké textové zprávy. Útočník může posílat nevyžádané zprávy do schránky, dokud nevyčerpá její kapacitu a znemožní tak její další funkčnost. Výsledkem je zaneprázdněný stav datové schránky a její nemožnost přijímat další zprávy a hlasové záznamy. [20]

4.2.1.2. Register flood

Register flood je používán pro směrování mnoha REGISTER zpráv, které jsou využívány pro registrace účastníků, na server registrar. Vzhledem k tomu, že posílání REGISTER žádostí vyžaduje ověření, je útok založen na vyčerpání zdrojů z důvodu náročnosti kryptografických operací spojených s autentizací. [17]

4.2.1.3. ICMP flood

Útok *ICMP flood* je druh DoS, kdy je posíláno velké množství ICMP (*Internet Control Message Protocol*) paketů, které jsou využívány pro zasílání chybových hlášení nebo oznámení o dostupnosti sítě. Používá se ke správě prostředků v síti Internet, jako např. měření obousměrného zpoždění nebo počtu směrovačů na trase mezi dvěma hosty. Běžně se používá ICMP echo pro zjištění zpoždění, kdy host vyšle ICMP paket a čeká na odpověď

s obousměrným zpožděním. V případě poslání velkého množství ICMP paketů útočník docílí zahlcení a oběť se stane nedostupnou.

4.2.1.4. DNS flood

DNS (*Domain Name System*) server hraje velkou roli v sítích založených na SIP protokolu. Jakmile DNS rozezná adresu domény obsaženou v hlavičce SIP zprávy, dojde k následnému poslání dál. Server SIP přijme URI zprávu a dále čeká na přiřazení adresy DNS serverem. Jestliže je rozeznání adresy složité, dojde k navýšení čekací doby DNS serveru až do doby, kdy čas vyprší. Z toho důvodu je útočníkem vyplněna hlavička SIP zprávy složitou adresou, aby bylo pro DNS server složité rozřešení adresy. U toho případu se jedná od *DNS flood*.

4.2.2. Memory Depletion Attack

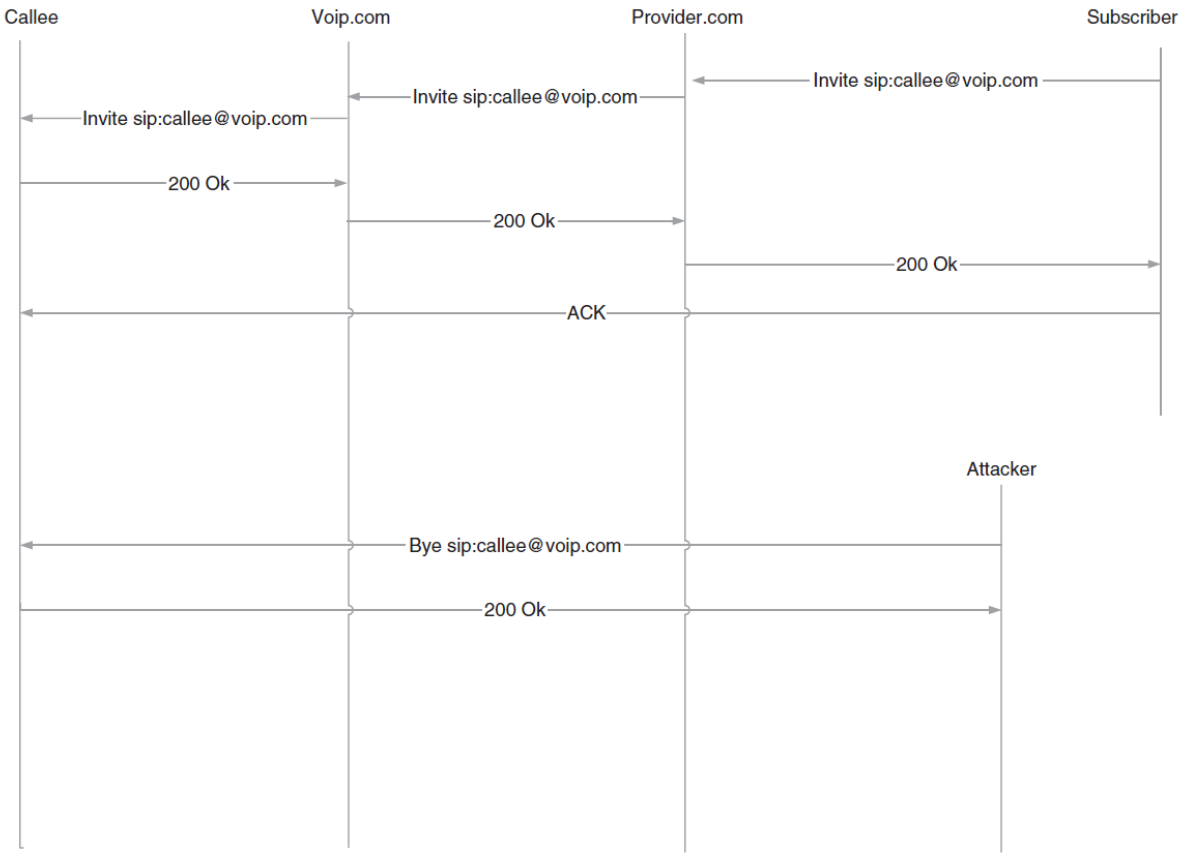
Různé internetové protokoly a služby můžou být uzpůsobeny jako posloupnosti transakcí. To znamená, že klient pošle požadavek serveru. Ten vezme požadavek ke zpracování a provede další akce pro jeho odbavení, jako je zjištění dalších informací nebo směrování požadavku na jiné servery. Operace je považována za ukončenou jakmile server obdrží od klienta potřebné informace nebo jakmile obdrží odpověď od jiného serveru. Po celou dobu trvání operace server udržuje informace popisující požadavek a aktuální stav operace. Paměť věnovaná stavovým informacím je obvykle dimenzován tak, aby server obsloužil určitý počet žádostí s operacemi s očekávanou průměrnou životností. Útočník může vyčerpat vyhrazenou paměť generováním velkého množství žádostí nebo prodloužením délky trvání operace. [12]

U protokolu SIP, jakmile jeho komponenta obdrží požadavek na spojení, udržují se informace o spojení, které jsou vytvořeny příchozím požadavkem a jsou udržovány, dokud není spojení ukončeno. Vyhrazená paměť umožňuje komponentě SIP provést maximální množství operací obsahující stavové informace potřebné průměrné alokace paměti a její přizpůsobení velikosti za určitý časový úsek. Útočník může vyčerpat paměťové prostředky generováním velkého množství požadavků, prodloužením času požadavku potřebného ke zpracování a jeho prodloužením doby strávené v paměti nebo zvýšením paměti potřebné pro udržení stavové informace operace. [12]

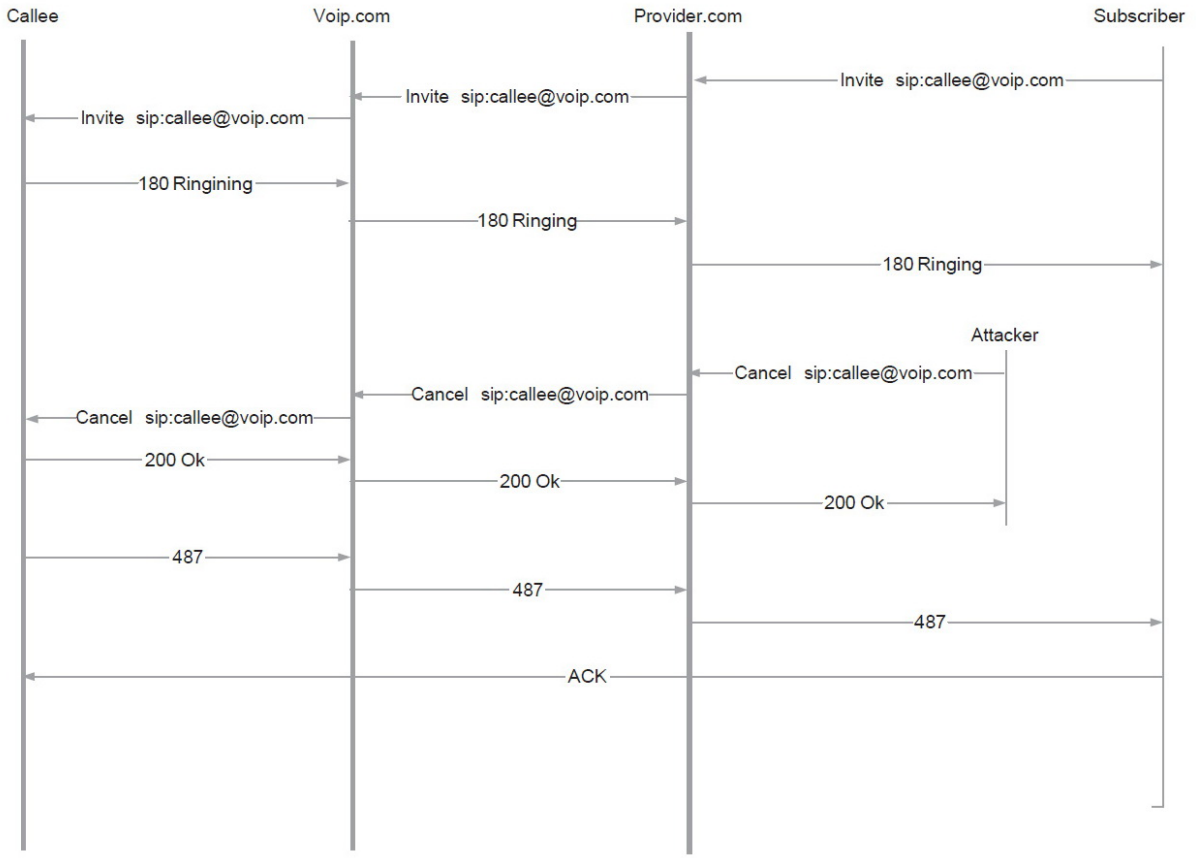
Pro zpracování příchozí zprávy SIP proxy server uloží zprávu do jeho vyrovnávací paměti (*buffer*). Data ve vyrovnávací paměti a délka uložení dat ve vyrovnávací paměti závisí na serveru, v jakém stavu funguje. Rozlišujeme dva typy: stavové a bezstavové. Server musí v některých případech minimálně udržovat data entit jako je AAA, DNS server nebo databáze.

4.2.3. Útoky zneužitím (*misused*)

Základem útoku zneužitím je správně upravená hlavička SIP zprávy, která je následně vložena do provozu SIP komunikace. Důsledkem zaslání upravené zprávy do relace je buď odstavení a přerušení služby nebo zamezení užívání služby účastníkem. Útok zneužitím se nesnaží zahltit nebo odstavit komponenty SIP serveru, pouze se snaží získat přístup ke komunikaci a následně se jí zmocnit a zneužít. Snaha útočníka je zamezit vytvoření dalšího spojení nebo ukončení již probíhajícího spojení. Pro ukončení spojení je typická úprava komunikace, kdy je v rámci spojení zaslána BYE zpráva, která umožní ukončení právě probíhajícího spojení u jednoho nebo více účastníků. BYE je znázorněn v tabulce 2. Podobný BYE útoku je CANCEL útok, který je zaměřen na ukončení spojení během přípravné fáze. Zpráva CANCEL ukončuje spojení během čekání na odpověď ACK. Neukončuje již probíhající komunikaci. Pokud je požadavek CANCEL doručen po obdržení zprávy ACK, je v tomto případě požadavek CANCEL ignorován. Diagram útoku CANCEL je zobrazen v tabulce 3.

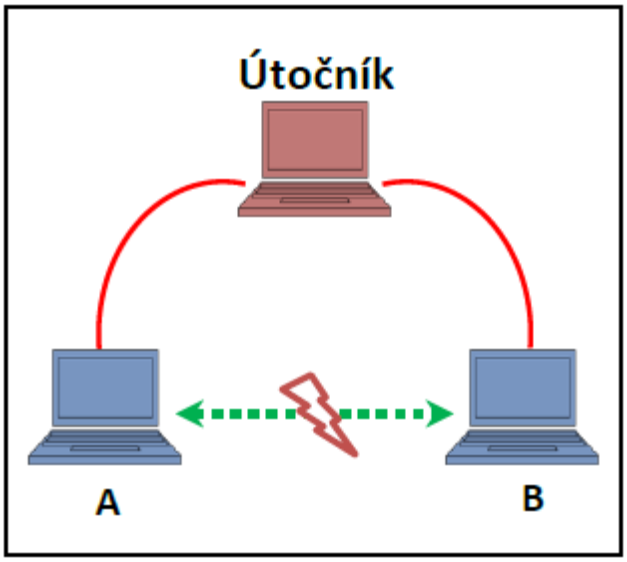


Tab. 2: Diagram útoku BYE [12]



Tab. 3: Diagram útoku CANCEL [12]

Útočník může útokem *Man-in-the-Middle* (MitM) získat kontrolu nad probíhajícím spojením, kdy pomocí správné úpravy pole *From* u hlavičky žádosti SIP získá neoprávněnou registraci v probíhajícím spojení. Jak zobrazuje obrázek 4.2, útočník je situován doprostřed spojení a má kontrolu nad oběma stranami komunikace. Princip útoku je založen na přesměrování probíhající komunikace, kdy komunikace jde přes útočníka a původní je zrušena. Na začátku spojení si uživatelé A a B vzájemně mění zabezpečovací klíče. Ty však zachytí útočník a každému z účastníků pošle svoje upravené klíče. Útočník tak získává kontrolu nad spojením a nad zprávami jdoucím od uživatele A i B a může je měnit, aniž by ostatní účastníci spojení tušili, že obdržené zprávy nepochází od původního odesílatele. Se získáním kontrolou nad spojením může útočník měnit parametry hlaviček zpráv (měnit jejich směrování pomoci From:, To:, …), přesměrovávat, ukončovat a měnit parametry hovoru. [18]



Obr. 4.2: Útok Man-in-the-Middle

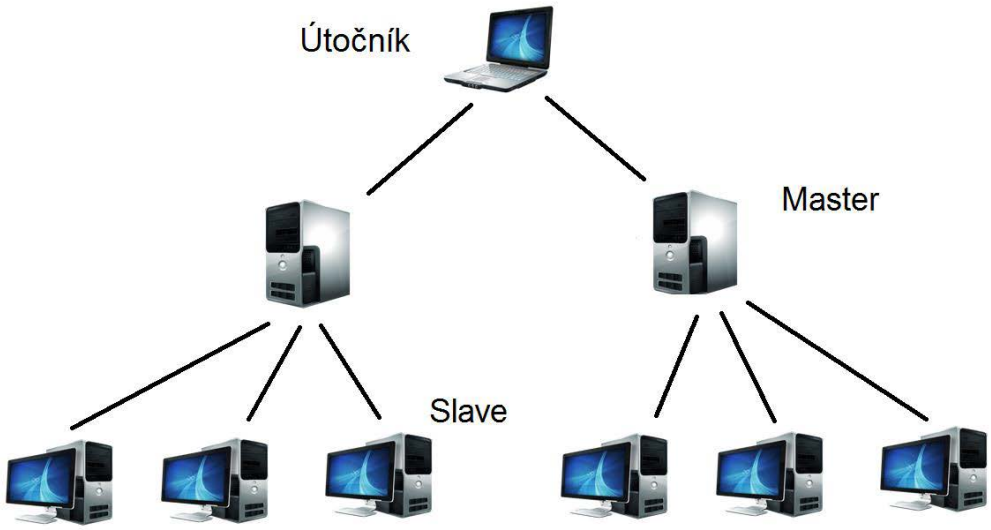
Mezi *misused* útoky lze zařadit i *Ping of death* (ping smrti). Každý IP datagram má velikost 65535 bajtů a navíc hlavička, která upřesňuje, kde se nachází po defragmentaci paketu fragmentovaná část. Hlavička má zpravidla velikost 20 bajtů, ale dá se navolit velikost až 24 bajtů. Zbylá část paketu slouží pro přenos dat. Fragmentace paketu se používá pro posílání většího množství dat v paketu, než je maximální udávaná velikost paketu. Ping smrti na základě chyby v implementaci IP, kdy je vytvořen větší paket než právě udávaná maximální velikost paketu, je poslán do sítě, kde může způsobit *buffer overflow (*přetečení zásobníku). Přetečení zásobníku zapříčiní vyčerpání vyrovnávací paměti. Úspěšnost útoku je závislá na implementaci OS (Operační Systém) a v některých případech může dojít k pádu nebo restartování systému. [15]
Pro *misused* útoky je potřeba, aby útočník měl další informace o SIP spojení (např. *Caller ID*…) a možnost podstrčení upravené SIP zprávy. To znamená přístup do sítě probíhajícího nebo plánovaného spojení.

4.3. DDoS

Distributed Denial of Service je podtypem útoku DoS, který má více jak jeden zdroj útoku. K aplikaci útoku je potřeba vysokého výpočetního výkonu, který zajistil dostatečnou sílu k vyčerpání paměti nebo procesoru oběti. Pro útok DDoS se nejčastěji používají tzv. *botnets*. Je to sada počítačů, které jsou kontrolovány útočníkem a můžou dodat dostatečný výkon, který útočník potřebuje. Jedná se většinou o počítače běžných uživatelů nebo korporací, které byly infikovány určitých virem, který umožňuje útočníkovi kontrolovat a ovládat infikované počítače. Vlastníci a uživatelé infikovaných počítačů mnohdy vůbec netuší, že jejich počítače jsou součástí organizované sítě určené pro DDoS útoky. Princip útoku spočívá v množství počítačů, které se snaží v jeden okamžik spojit se serverem. V případě

většího počtu uživatelů, kteří se chtějí připojit, překročí počet, na který je server dimenzovaný a dojde k neúspěšnému zpracování požadavku. Server se zahltí požadavky a v mnohých případech dojde k omezení jeho výkonu a někdy i k jeho vyřazení. Výhoda útoku DDoS je v množství aplikovaného výkonu a také v zachování diskrétnosti útočníka. Topologie útoku DDoS je zobrazena na obrázku 4.3.

Můžeme rozlišit dva typy DDoS útoku, *botnets* a *amplification*. U *botnets* typu útoku musí mít útočník pod kontrolou počítače, které používá a přímo s nimi manipuluje. Přímo útočník ovládá počítače. Při útoku DDoS *amplification* útočník nepotřebuje plnou kontrolu počítačů. Skrze ně pouze aplikuje útok a rozloží útok do více zařízení.



Obr. 4.3: Topologie při útoku DDoS

4.4. Analýza provozu

U analýzy provozu sleduje útočník pakety ve VoIP, které může analyzovat a může je využít pro odposlech volání v rámci sítě Internet. Zachytávání paketů jsou dvojího typu, aktivní a pasivní odposlech. Pro pasivní odposlech se nejčastěji využívá nástroj Wireshark. Jedná se o GPL *sniffer*, tedy o program schopný analyzovat síťové protokoly nejrůznějších typů. Výhodou programu je práce s mnoha protokoly a jejich export do formátů různých typů, které jsme schopni otevřít na většině platformách. Pro odposlech hovoru navíc podporuje analýzu protokolu RTP. Protokol RTP je důležitým protokolem v rámci komunikace VoIP volání. Jakmile projdou všechny inicializační postupy, je zahájeno samotné přenášení mediální informace, v případě VoIP většinou hlasu. Wireshark podporuje nejen procházení zachycené mediální informace, ale dokážeme ji s jeho pomocí uložit. Informace je ukládána do formátu *.au, který je podporován mnohými běžně dostupnými přehrávači všech

platforem. Ovšem pro jednodušší práci s nahrávkou je lepší převod do více používaných zvukových formátů, např.mp3, wav, flac apod. [19]

Pro aktivní odposlech používáme ARP protokol a jeho upravených ARP rámců. Jedná se o ARP *spoofing*, který umožňuje útočníkovi vydávat se za jiný počítač připojený do stejné sítě. Posláním, za pomoci protokolu ARP, dotazu všem ostatním v síti pro získání jejich MAC adres. Během výměny informací útočník může zaznamenat adresu, za kterou se potom může maskovat.

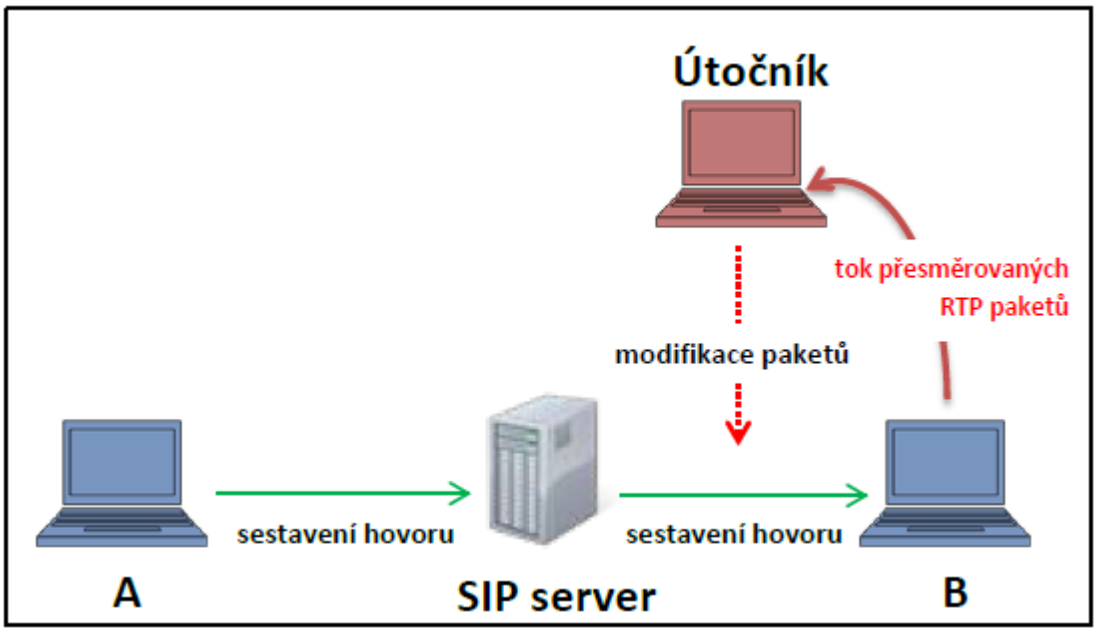
V publikaci [14] je uveden způsob analýzy provozu pomocí dekodéru DTMF (*Dual-tone multi-frequency*). DTMF se jinak říká tónová volba a jde o tón, složený z dvojice sinusových signálů s přesně danýma frekvencemi, jedinečnými pro každý jednotlivý znak. V telekomunikacích se používá k zadávání volby, ovládání, nastavování ústředen atp. Útočníkem může být tato volba zneužitelná, kdy přes telefon zadáváme citlivé informace, jako třeba čísla PIN (Personal Identification Number) nebo jiné autorizační informace. Pomocí dekodéru DTMF je útočník schopen „odposlechnout“ a následně dekódovat tyto citlivé informace a může je dále zneužít.

Důležitým prvkem u telefonie je *caller id* (*Caller Identification*), díky čemuž jsme schopni identifikovat volajícího v případě přijímaného hovoru. Identifikátor je specifikován v hlavičce *From*, která je součástí SIP požadavků, např. INVITE. Útokem *caller id spoofing* je útočník schopen změnou hlavičky *From:* maskovat svoje pravé *caller id*. Na internetu jsme schopni najít i některé stránky, které umožňují změnu identifikaci volaného.[11] Způsoby umožňující *caller id spoofing* zahrnují specializované telefonní karty, online telefonní služby nebo jednoduché vhodné programy. K záměně identifikátoru je možné využít program *inviteflood*, který umožňuje manipulaci s hlavičkou *From* INVITE žádosti. [10]

Další možnost analýzy síťového provozu je pomocí útoku *Man-in-the-Middle*, který je popsaný v kapitole 4.2.2.

4.5. Přesměrování

Jak už bylo mnohokrát napsáno, díky jednoduché skladbě signalizačního protokolu SIP, konkrétně žádostí a jejich hlaviček, je přesměrování hovoru vcelku jednoduché. Přesměrování hovoru jde za předpokladu změny pole INVITE žádosti. U přesměrování hovoru je důležité pole To:, tedy adresa, kam bude spojení směrováno. Probíhající hovor lze přesměrovat úpravou protokolu SDP (*Session Description Protocol*), který se přenáší v rámci INVITE zprávy během započatí SIP komunikace. SDP protokol popisuje nezbytné parametry pro mediální přenos. Měl by obsahovat informace o typu média, použitém transportním protokolu, kodeku, rychlosti přenosu. Navíc může být obsažena informace nesoucí IP adresu a port. Pro přesměrování postačí upravit směrovací adresu, kam chceme síťový provoz směrovat. Přesměrování hovorů je znázorněno na obrázku 4.4.



Obr. 4.4: Přesměrování RTP paketů

Svým způsobem můžeme za přesměrování považovat útok krádeže hovoru (SIP *call hijacking*). Spočívá v ukradení hovoru a jeho následného přesměrování dle zvolené adresy. Obrázek 4.4 znázorňuje princip přesměrování RTP toku. *Call hijacking* spojuje dva jednotlivé útoky. První fází je de-registrace, kdy útočník v prvním případě musí zrušit původní registraci, kterou udělal účastník hovoru. Součástí první fáze je maskování uživatelského ID. V druhé fázi se provede nová registrace. Zapojením útočníka do komunikace je umožněna úprava komunikace a její přesměrování na útočníkovo zařízení, které v rámci komunikace zaregistroval. Na toto zařízení je nově směrován tok RTP. Někdy fáze de-registrace není nezbytně nutná. Lze použít editovanou INVITE zprávu, která obsahuje informace o zařízení, které má být napadeno (ID osoby). Existují zařízení, která mohou pracovat s protokolem SIP umožňující směrování na obě zařízení (na zdrojové i cílové zařízení). Tím, že se nevyskytuje proces de-registrace, je o jeden proces míň a tím pádem se sníží pravděpodobnost odhalení útoku. Daní za snížení pravděpodobnosti odhalení je poskytnutí pouze pasivního náhledu hovoru. Útočník může hovor poslouchat a nahrávat, ovšem nemůže hovor přesměrovat na jiné zařízení.

5. Praktická část

Praktická část je věnována otestování útoků DoS, konkrétně útok SIP *flooding*. Ke generování požadovaných signalizačních zpráv a uskutečnění útoků byl použit program SIPp. Pro odzkoušení útoků byla použita reálná struktura sítě založené na Avaya Aura® topologii. Přístup k zařízením Avaya Aura® byl umožněn za spolupráce firmy Algotech.

5.1. Metodika testování

Metodika testování zabezpečení PBX ústředen je postup pro otestování řešení pobočkové ústředny, které slouží pro stanovení rizik ústředny spojenými s nedostatečným zabezpečením nebo nedostatečným hardwarovým možnostem ústředny.

5.1.1. Návrh obecné metodiky

1) Návrh topologie.
2) Zapojení a příprava topologie. Je důležitá volba prostředí, kde bude test probíhat. Výsledky testů by mohly mít dopad na funkčnost sítě.
3) Stanovení cílů testu. Většinou jde o test zatížení ústředny a její funkčnosti v různých situacích.
4) Zavedení testovacího hovoru mezi dvěma koncovými body a jeho zachycení pomocí programu Wireshark. Díky němu si můžeme důkladně prostudovat hlavičky všech signalizačních zpráv potřebných k uskutečnění testování a k pozdější aplikaci skrze soubor XML.
5) Příprava potřebných podpůrných konfiguračních souborů v závislosti na možnostech a potřebách ústředny PBX zjištěných ze zachyceného testovacího hovoru. Je potřeba nakonfigurovat soubor XML, který slouží k simulaci signalizačních zpráv. Pomocí něj je možná nakonfigurovat jakoukoliv signalizační zprávu a zavést ji do komunikace. Dalším potřebným souborem je CSV, který slouží k implementaci proměnných použitých v XML schématu. CSV soubor doplňuje hodnoty do polí [fieldXY] označující číslo sloupce v CSV souboru, oddělené středníkem. Vzorové konfigurační soubory jsou obsaženy v příloze.
6) Registrace. Před spuštěním samotného hovoru nebo zátěžového testu, je potřeba koncová zařízení registrovat na příslušných IP adresách.
7) Spuštění SIPp s předem připravenými konfiguračními soubory XML a CSV, viz. bod 4). SIPp je potřeba spustit odděleně na dvou počítačích nebo na jednom PC se dvěma virtuálními stanicemi. Jen pro simulaci volajícího (generátor zpráv) UAC a druhý pro simulaci volaného (odpovídač) UAS. Příkazy pro spuštění patřičné konfigurace je pomocí následujících příkazů:

   Pro UAC: `./sipp [remote_ip] -sf [nase_uac].xml -inf [xyz].csv -i [local_ip] –rsa [remote_host]`

Pro UAS: `./sipp -sf [nase_uas].xml -i [local_ip] -rsa [remote_host]`

kde -sf … načte XML schéma
- inf … načte CSV soubor
- i … zadá IP adresu pro [local_host]
- rsa … zadá IP adresu pro [remote_host] (IP adresa ústředny PBX)

8) Spuštění zátěžového testu z třetího PC.
`./sipp [remote_ip] -sf [nas_testovaci_soubor].xml -inf [xyz].csv -i [local_ip] –rsa [remote_host] –r X –rate_increase Y –fd Z –trace_counts`

kde - sf … načte XML schéma
- inf … načte CSV soubor
- i … zadá IP adresu pro [local_host]
- rsa … zadá IP adresu pro [remote_host] (IP adresa ústředny PBX)
- r … počáteční *rate* (počet zpráv za vteřinu)
- rate_increase … navýšení *rate* o hodnotu Y
- fd … navýšení *rate* po čase Z
- trace_counts … log s výstupnými hodnotami počtu zpráv (log se aktualizuje při každé změně hodnoty *rate*)

Zátěžový test je spouštěn v konfiguraci počáteční hodnoty *rate* 10 zpráv/s. Po určité době (delší úsek je vhodnější pro ustálený stav) navýšíme hodnotu *rate* o určitou hodnotu. Doporučují se spíše menší kroky kvůli zamezení skokových výchylek přizpůsobování ústředny. Délka testu závisí na typu generovaných zpráv. Některé zprávy (např. OPTIONS) jsou jednodušší na zpracování, proto hodnoty generovaných zpráv můžou být vyšší a test bude probíhat delší dobu.

9) Zhodnocení výsledků z logu vytvořeným příkazem *-trace_counts*. Porovnáme výstupní hodnoty poslaných zpráv a potvrzených zpráv. Dostáváme tak bod, kdy ústředna přestává zpracovávat požadavky se stoprocentní úspěšností.
10) Návrh na zlepšení.
11) Aplikace případných vylepšení a opětovné spuštění testu.

5.1.2. Metodika testu Avaya Aura

Na základě metodiky sepsané v kapitole 5.1.1 je provedeno otestování architektury Avaya Aura®. Pro začátek testu je tedy důležitý návrh topologie, která je znázorněna na obrázku 5.1 Vzhledem k testování zabezpečení se doporučuje otestovat systém v prostředí, které bude odděleno od ostrého provozu. Průběh testů může mít vliv na funkčnost systému a může způsobit jisté problémy. Jakmile máme topologii navrhnutou, je potřeba připravit testovací

prostředí. Pro simulaci hovorů a útoků poslouží program SIPp. Před samotným testem bylo potřeba přizpůsobit schéma komunikace v programu SIPp v závislosti na funkcích a požadavků architektury Avaya Aura®. Zejména bylo potřeba přizpůsobit autorizační procesy, které jsou standardně vyžadovány registrace, ale také u zpráv INVITE a BYE. Zejména byl problém s ověřovacími zprávami INVITE zprávy, kdy bylo potřeba hlavičku přizpůsobit tak, aby vyhovovala požadavkům Avaya®Session Manager. Test bylo potřeba spustit na dvou různých počítačích z důvodu různých IP adres, na kterých bylo potřeba pustit souběžně běžící programy SIPp. Na jednom počítači ho bylo potřeba pustit v režimu UAC, tedy klient, který generoval hovory, a na druhém počítači běžel v režimu UAS, tedy „odpovídač“ na generované hovory. Na počítači, kde byl puštěn SIPp jako UAS bylo také potřeba pustit SIPp schéma pro registraci. Každý odpovídací uživatel musí být zaregistrovaný na patřičném serveru. V našem případě na Session Manageru. Třetí počítač slouží ke generování útočných zpráv.

Základem testu je zavedení odpovídajícího počtu telefonních hovorů, v závislosti na technických parametrech ústředny. Počet hovorů se bude odvíjet od schopnosti ústředny obsloužit určitý počet hovorů. Výsledkem testu bude zhodnocení ústředny, zda je schopná obsloužit daný počet hovorů a zda je schopna fungovat i při zavedení útoků během testovacích hovorů.

Další částí testu bude generování SIP zpráv, které budou generovány delší časový úsek a po určitých intervalech bude generování zpráv navýšeno o určitý počet zpráv za vteřinu. Všechny testy, krom špičkového zatížení, měly shodný počet generovaných zpráv na začátku testu. Shodně je nastavena počáteční hodnota 10 zpráv za vteřinu. V závislosti na typu generovaných zpráv a jejich časové náročnosti byly voleny příspěvky generovaných zpráv za vteřinu. Ve většině případů se jednalo o navýšení 100 zpráv za vteřinu po 30 nebo 60 vteřinách.

Následujícím testem bude špičkové zatížení Session Manageru. Test spočívá v generování tisíců zpráv za vteřinu v jednom okamžiku. Bude se tak simulovat špičkového zatížení Session Manageru a jeho reakce a přizpůsobení na danou situaci.

Poslední částí testování bude otestování Session Manageru a Session Border Controlleru se zapnutými bezpečnostními prvky. Tato část slouží k otestování systému za reálného provozního nastavení. Bude tak ověřena funkčnost především *firewallu*, který je potřebný pro zabezpečení proti DoS útokům.

Testování probíhá zejména pro ověření nebo stanovaní možností funkčnosti systému PBX a jeho schopností fungovat i v krizových situacích.

5.2. SIPp

SIPp je open source generátor provozu, který je navržený převážně pro testování prostředí SIP. SIPp je schopen simulovat chování jako UAC i UAS a generovat signalizaci a přenos média. Program se s velkou oblibou používá pro generování provozu a simulování zátěžových modelových situací pro ověření zabezpečení ústředny a jejího provozu v různých situacích, které mohou eventuálně nastat. Původní zdrojový kód programu SIPp byl napsán Richardem Gayraudem a upraven Olivierem Jacquesem. Díky stoupající oblibě je program stále více zdokonalován a upravován komunitou developerů. Program si získal popularitu díky jednoduchému použití. Ovládání programu je skrze jednoduchou příkazovou řádku, přes kterou se dají ovládat všechny funkce. Postupem vývoje byla implementována funkce importování XML souborů, pomocí kterých se dají tvořit složitější komunikační schémata a CSV souborů obsahujících rozšiřující data použitelných při psaní skriptů. Nemalou výhodou SIPp je v jazyku, kterým je napsaný. Vzhledem k tomu, že je napsán programovacím jazykem C++, jsou lidé schopni si program poupravit a případně vylepšit. Více informací o programu jsou dostupné na stránkách programu [23].

Důležitou vlastností je úprava a tvorba vlastních schémat. Součástí programu jsou již předem definovaná základní schémata, která však nefungují na topologii Avaya Aura®. Je potřeba tedy upravit schémata v závislosti na vlastnostech ústředny. Podle chování ústředny se postupně upraví celý XML skript. V případě Avaya Aura® bylo potřeba přizpůsobit SIPp hlavně z důvodu ověřování. Fázemi ověření podléhá jednak proces registrace, ale také zahájení relace a ukončení relace zprávou BYE.

5.3. Test Avaya Aura

Zařízení Avaya Aura® byla poskytnuta firmou Algotech, kde byl test aplikován. Topologie SIP byla v poloostrém provozu, kdy volání SIP bylo používáno v minimální míře a nebyl nijak zvlášť ovlivněn provoz. Drtivá komunikace společnosti Algotech probíhá využíváním protokolu H.323.

5.3.1. Zařízení

Konfigurace 1.notebooku (Dell Latitude E6430):

- Procesor Intel® Core i7-3740QM CPU 6MB cache @ 2.70GHz (3.46GHz)
- 8GB (2x4GB) RAM DDR3 @ 1600MHz
- SSD 256GB
- 1Gb Ethernet
- OS Debian Kali Linux (Virtual Box)

Konfigurace 2. notebooku (Sager P150HM):

- Procesor Intel® Core i7-2630QM CPU 6MB cache @ 2.00GHz (2.76GHz)

- 8GB (2x4GB) RAM DDR3 @ 1600MHz
- SSD 128GB
- 1Gb Ethernet
- OS Debian Kali Linux (Virtual Box)

Avaya®Session Manager (HP ProLiand DL360 G7):

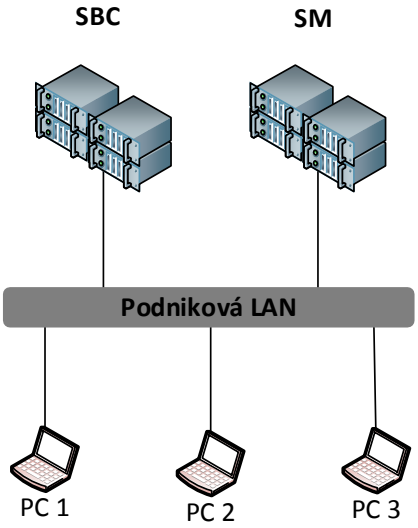
- Procesor Intel® Xeon E5620 CPU 12MB cache @ 2.4GHz
- 12GB (3x4GB) RAM DDR3 @ 1333MHz
- 4x1Gb Ethernet
- HDD 2x300GB

Avaya®Session Border Controler (Dell PowerEdge R210 II):

- Procesor Intel® Xeon E3-1220 CPU 8MB cache @ 3,1GHz
- 8GB (2x4GB) RAM DDR3 @ 1333MHz
- 6x1Gb Ethernet
- HDD 2x300GB

5.3.2. Topologie

Topologie zařízení použitých pro test je znázorněna na obrázku 5.1. Pro test byla použita zařízení zmíněná v kapitole 5.3.1.



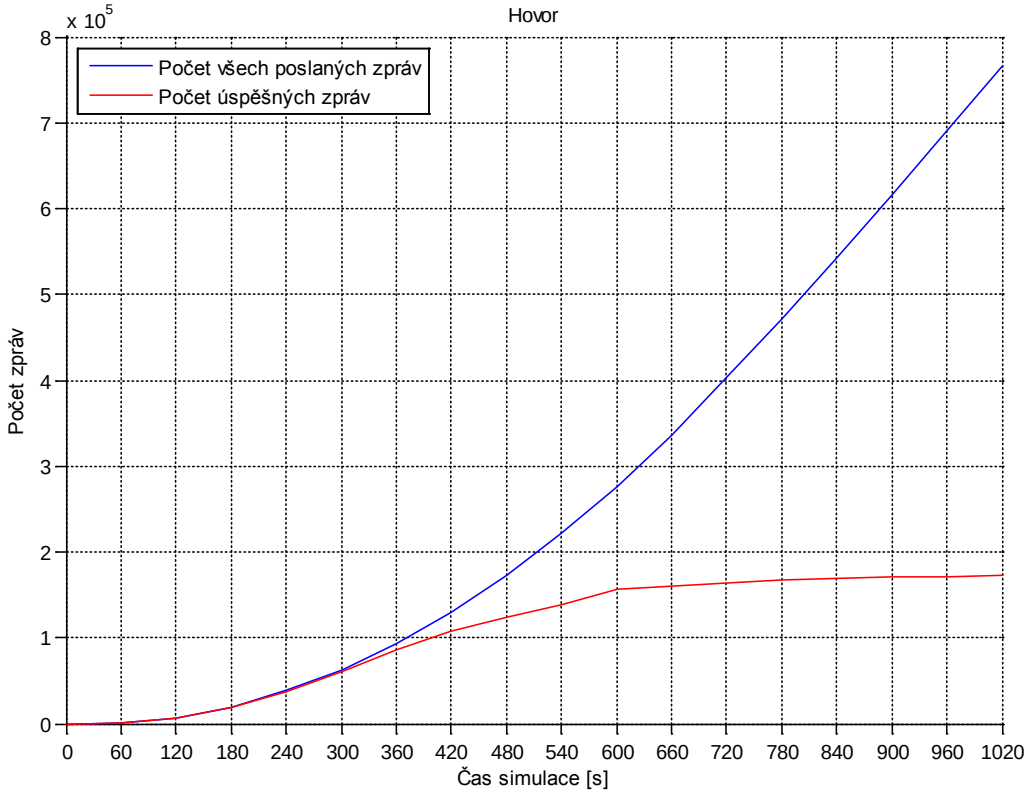
Obr. 5.1: Topologie zapojení testovacích komponent

5.3.3. Optimální provoz SM a jeho hranice

Test optimálního provozu proběhl pomocí generování hovorů až po hranici, kdy požadavky přestanou být zpracovávány. Pro test bylo vytvořeno 100 uživatelů. Jedna polovina uživatelů byla zaregistrována na jednom počítači a druhá polovina padesáti uživatelů byla registrována na počítači druhém. Každý z uživatelů registrovaných na straně A provolával uživatele na straně B. Simulovaný hovor proběhl bez přenosu RTP paketů, protože nebyl podstatný vzhledem k tomu, že se v našem případě RTP pakety přenášejí napřímo mezi koncovými uživateli. Nedochází tak k vytížení ústředny, proto nemusí být zařazen do simulace. V technické dokumentaci [26] se píše, že Session Manager zvládá obsluhovat 90000 simultánních relací. [25]

Délka testu: 1020s

```
INVITE ---------->
   100 <----------
   407 <----------
   ACK ---------->
INVITE ---------->
   100 <----------
   180 <----------
   200 <----------
   ACK ---------->
  Pause [    10.0s]
   BYE ---------->
   407 <----------
   BYE ---------->
   200 <----------
```



Graf 5.1: Generování zpráv v závislosti na čase

Jak jde vidět v grafu 5.1, této hranici jsme se přiblížili a výsledky byly takřka 90000 zpracovaných relací. Po překročení této hranice SM pomalu přestal zpracovávat nově započínající relace. Nezpracované relace se začaly znovu posílat a následně kumulovat v SM. Kumulují se tak dlouho, dokud nedošlo k vypršení časového limitu pro zpracování, který je popsán v kapitole 4.2.1.

5.3.4. Test zátěže (vypnutý firewall)

Zátěžový test útokem probíhal generování SIP zpráv pomocí programu SIPp podle předem připravených schémat. Soubory schémat jsou přiloženy na CD. Délka testu probíhala v závislosti na typu generovaných zpráv. Generované zprávy byly posílány směrem k Session Manageru, na kterém byl vypnutý firewall a tudíž byl naprosto nechráněný. Důležitým faktorem jsou úspěšně poslané zprávy oproti neúspěšným. Hodnoty úspěšných a neúspěšných zpráv jsou zaznamenány v grafech, které slouží k porovnání úspěšnosti posílání zpráv.

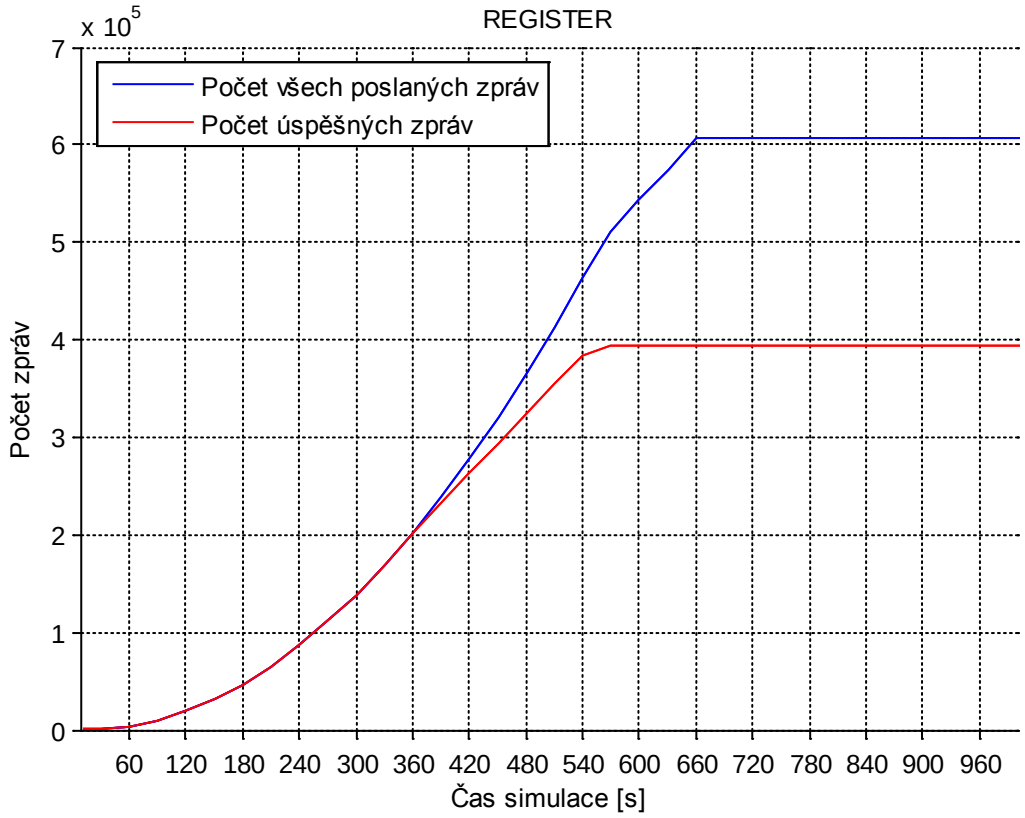
5.3.4.1. REGISTER

Registrace probíhá u protokolu SIP ve dvou krocích. V prvním kroku se při poslání REGISTER žádosti hláškou 401 vyzve k autorizaci a dojde k vyžádání autorizačního klíče. V druhém kroku dojde k poslání REGISTER zprávy již s obsaženými informacemi autorizace.

```
REGISTER ---------->
     100 <----------
     401 <----------
REGISTER ---------->
     200 <----------
```

Test byl spuštěn v konfiguraci: počáteční hodnota generovaných zpráv byla 10 zpráv za vteřinu a v pravidelných intervalech 30s byla tato hodnota navýšena o 100 zpráv za vteřinu. Na konci testu byla hodnota generovaných zpráv 2310 zpráv za vteřinu.

Délka testu: 960s



Graf 5.2: Generované zprávy REGISTER v závislosti na čase

Test proběhl generováním REGISTER žádostí s validní registrací. Cílem testu bylo zjištění zpracování validních REGISTER žádostí. Podle technických parametrů [25] by měl Session

Manager zvládnout obsloužit 800 žádostí o registraci za vteřinu. Z grafu 5.2 je patrné, že hraniční bod, kdy nedocházelo ke stoprocentnímu obsluhování žádostí, nastal po 360s, kdy SM zvládal obsluhovat 1110 zpráv za vteřinu. V čase 540s, kdy generování zpráv čítalo 1810 zpráv za vteřinu, již nedocházelo k žádnému obsluhování požadavků. V čase 660s se přestaly generovat nové zprávy, protože výkon CPU zdroje pro generování zpráv už nebyl dostatečný a nebyly obdržovány potvrzovací zprávy ze SM. Hromadění neobsloužených požadavků mohlo být způsobeno nedostatečně rychlým obsloužením a vypršením limitu k obsloužení některých požadavků. Podle technických předpokladů SM zvládl zprávy obsluhovat lépe, než je udáno v dokumentaci [26].

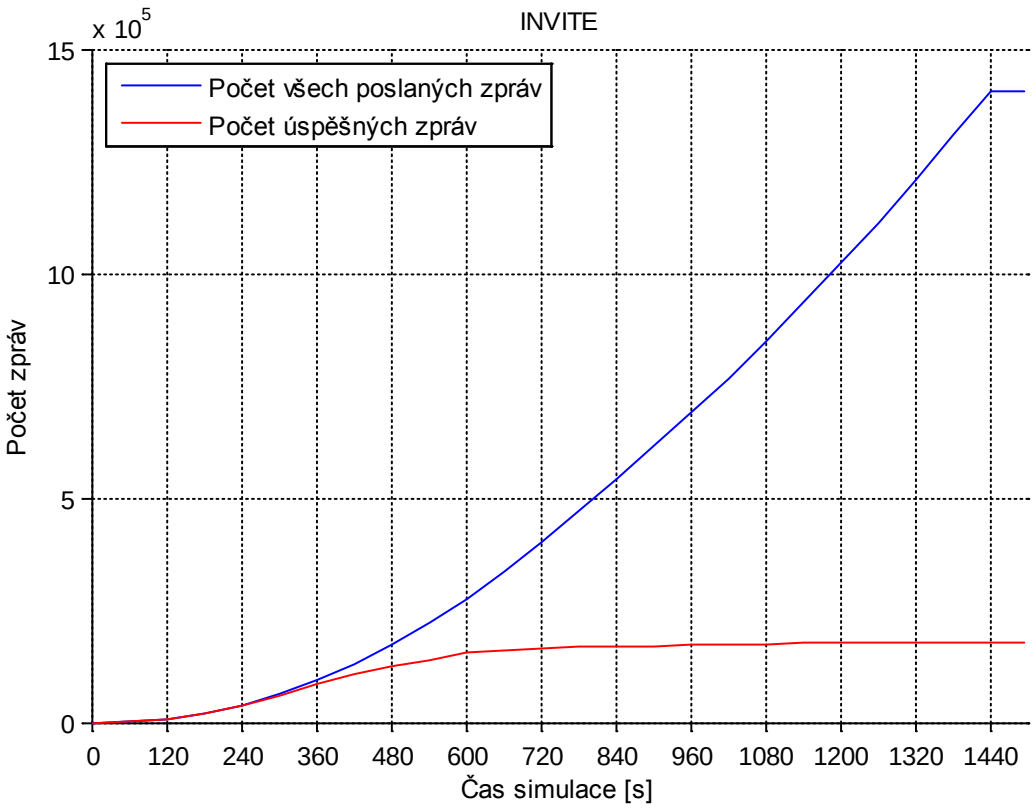
5.3.4.2. INVITE

Začátek hovoru INVITE zprávou probíhá u Avaya Aura® architektury na stejném principu jako proces registrace. V prvním kroku také dochází při poslání INVITE žádosti hláškou 407 k vyzvání k autorizaci a dojde k vyžádání autorizačního klíče. V druhém kroku dojde k poslání INVITE zprávy již s obsaženými informacemi autorizace.

```
INVITE ---------->
   100 <----------
   407 <----------
   ACK ---------->
INVITE ---------->
   100 <----------
```

Test byl spuštěn v konfiguraci: počáteční hodnota generovaných zpráv 10 zpráv za vteřinu a v pravidelných intervalech 60s byla tato hodnota navýšena o 100 zpráv za vteřinu. Na konci testu byla hodnota generovaných zpráv 2510 zpráv za vteřinu.

Délka testu: 1494s



Graf 5.3: Generované zprávy INVITE v závislosti na čase

Testování INVITE žádostmi bylo na stejném principu, jako u REGISTER žádostí, kdy zjišťujeme schopnost SM zpracovat validní INVITE žádosti. U zpráv INVITE byl výsledek podobně uspokojivý, jako u REGISTER žádostí. Technická dokumentace [26] udává vytváření nových relací v objemu 150 relací za vteřinu. Graf 5.3 ukazuje průběh vytváření požadavků, kdy v čase 360s čítalo generování 610 zpráv za vteřinu. Jedná se tedy o čtyřikrát lepší hodnotu, než udává technická dokumentace [26].

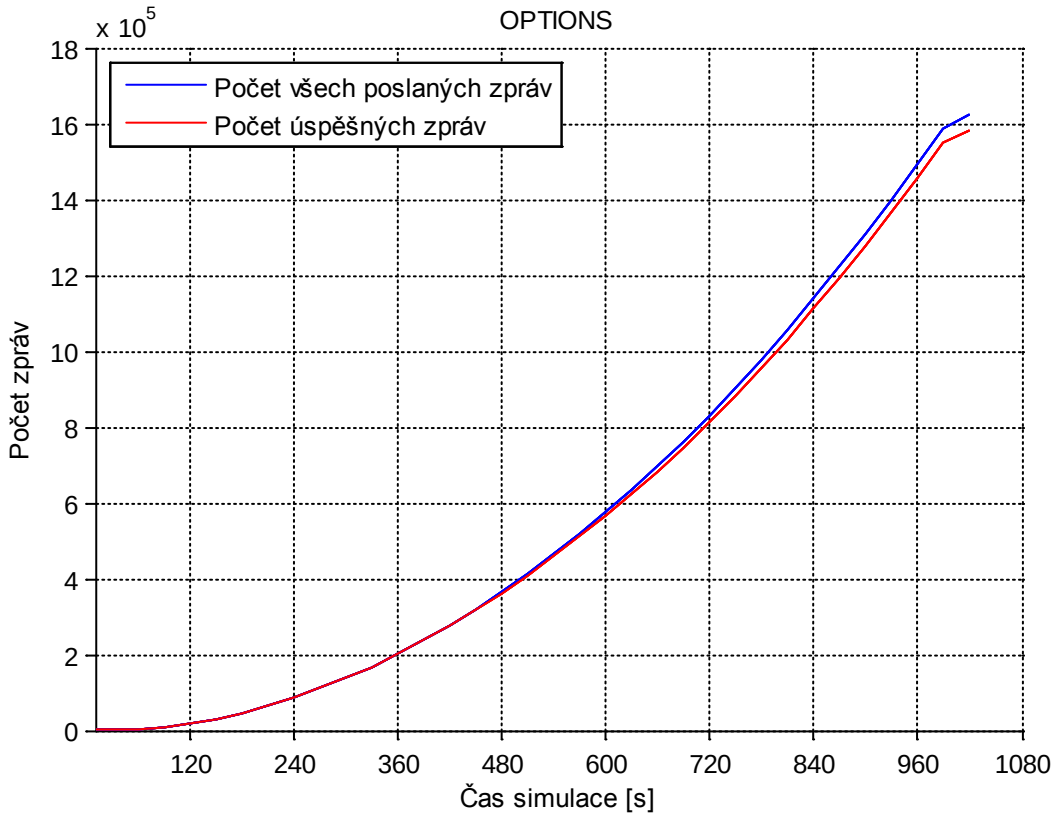
5.3.4.3. OPTIONS

Posílání OPTIONS zprávy byla nejjednodušší zprávou z testu, protože došlo pouze k poslání zprávy OPTIONS a na to byla obdržena potvrzující zprávy 200 OK.

```
OPTIONS ---------->
    200 <----------
```

Test byl spuštěn v konfiguraci: počáteční hodnota generovaných zpráv 10 zpráv za vteřinu a v pravidelných intervalech 30s byla tato hodnota navýšena o 100 zpráv za vteřinu. Na konci testu byla hodnota generovaných zpráv 3410 zpráv za vteřinu.

Délka testu: 1020s



Graf 5.4: Generované zprávy OPTIONS v závislosti na čase

Graf 5.4 nám potvrdil předpoklad, že se jedná o posílání jednoduché zprávy. Je evidentní, že docházelo prakticky k bezproblémovému obsloužení požadavků a následných odpovědí na zprávu OPTIONS. Z pohledu vytížení Session Manageru nedošlo k vyššímu vytížení a požadavky byly zpracovány bez větších problémů.

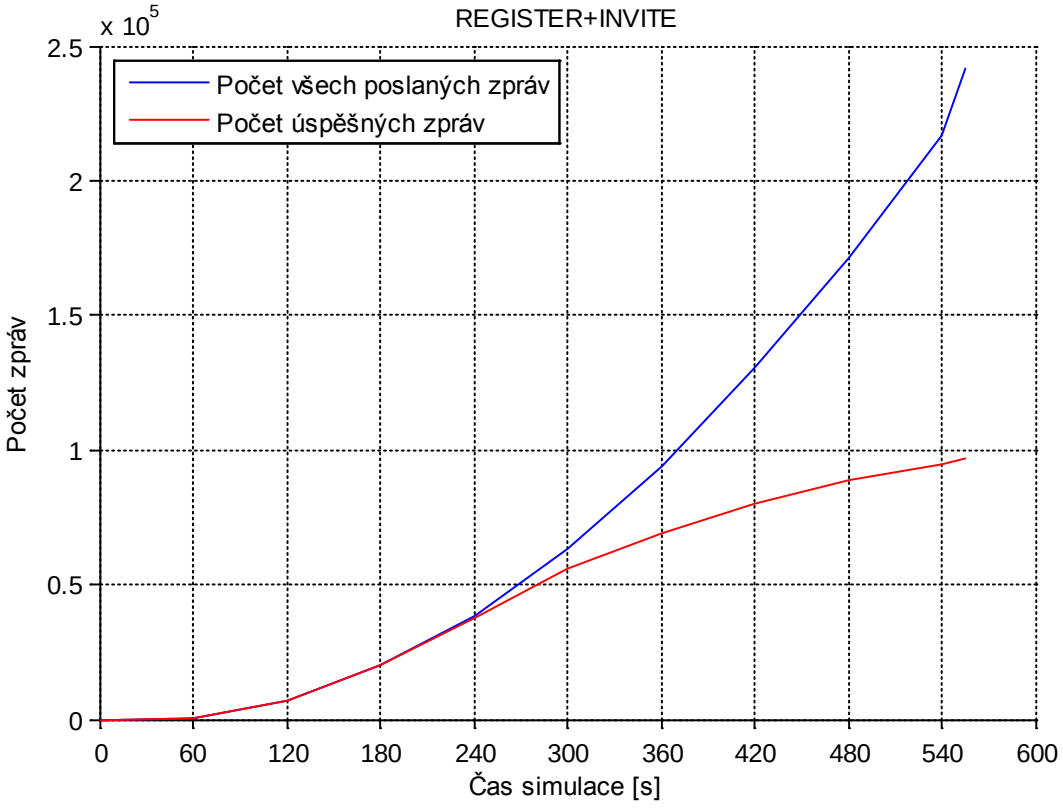
5.3.4.4. REGISTER+INVITE

V případě testu REGISTER+INVITE byly generovány zprávy simulující kompletní započetí hovoru, tedy registrace uživatele a následné zahájení hovoru zprávou INVITE. Jednalo se o nejnáročnější test, protože se muselo generovat více složitějších zpráv včetně ověřovacích procesů.

```
REGISTER ---------->
     100 <----------
     401 <----------
REGISTER ---------->
     100 <----------
     200 <----------
  INVITE ---------->
     100 <----------
     407 <----------
     ACK ---------->
  INVITE ---------->
     100 <----------
  CANCEL ---------->
```

Test byl spuštěn v konfiguraci: počáteční hodnota generovaných zpráv 10 zpráv za vteřinu a v pravidelných intervalech 60s byla tato hodnota navýšena o 100 zpráv za vteřinu. Na konci testu byla hodnota generovaných zpráv 1010 zpráv za vteřinu.

Délka testu: 555s



Graf 5.5: Generované zprávy REGISTER+INVITE v závislosti na čase

Test REGISTER+INVITE zahrnoval generování registračních požadavků a následně požadavky o zahájení relace. Pro test nebyly žádné technické předpoklady, protože jednalo o kombinací dvou metod zpracování požadavků. Vzhledem k vyšší náročnosti zpracování více požadavků, výsledek není až tak uspokojivý. Hranice mezi úspěšně zpracovanými požadavky a neúspěšnými nastává po 240s u hranice 410 zpráv za vteřinu jak je vidět v grafu 5.5.

Při testu REGISTER+INVITE byla navíc ověřena funkčnost Session Manageru, kdy během spuštěného testu bylo vyzkoušeno, zda-li je schopnost sestavení dalšího validního hovoru. Test proběhl za pomoci třetího PC (znázorněno na obrázku 5.1), kde byl nainstalovaný X-Lite *softphone* klient, z kterého bylo zavoláno na další *softphone* nainstalovaný na druhém PC. Proběhlo deset testovacích pokusů o sestavení hovoru, které byly provedeny při maximálním vytížení ústředny. Cílem testu bylo zjistit, zda ústředna může zpracovat další hovory sestavené klasicky, tedy ne za pomocí generátoru. Testem bylo zjištěno, že validní hovor byl každý druhý. Můžeme tedy vidět, že i v maximální zátěž ústředny, je stále schopna odbavovat další hovory, ovšem úž jen s padesáti procentní úspěšností. Jedná se však o nejkritičtější část testu, protože je vypnuto celé zabezpečovací řešení a ústředna je tak vystavena největšímu množství zákeřných žádostí.

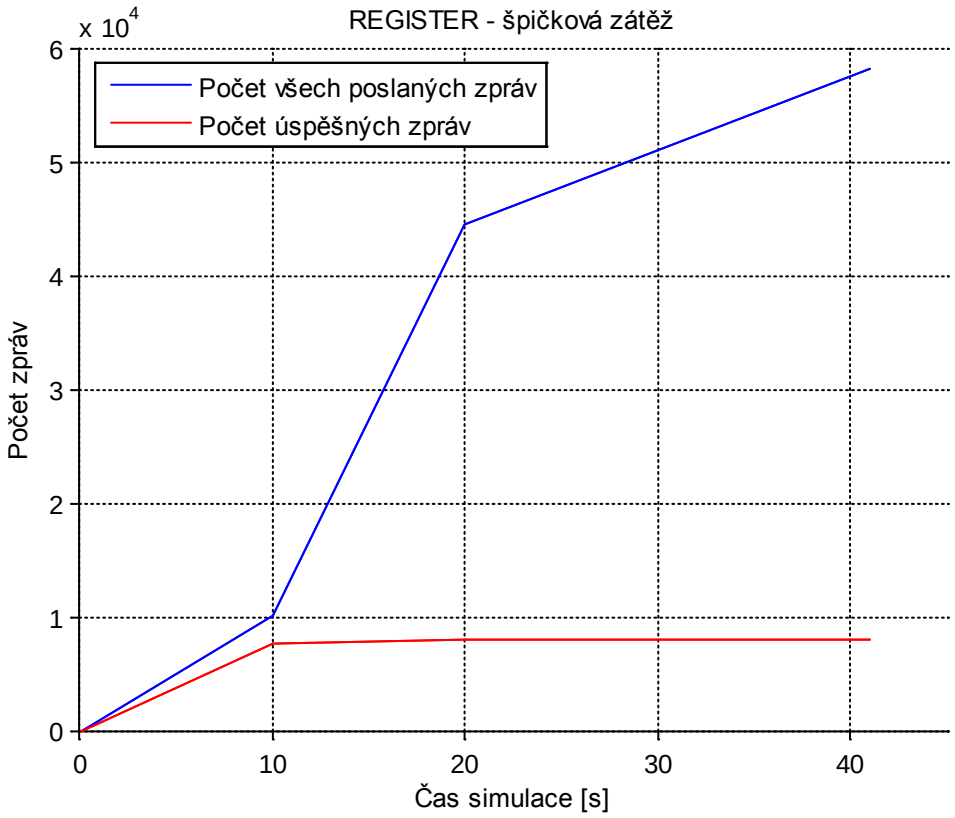
5.3.4.5. REGISTER – špičková zátěž

Test špičkové zátěže byl prováděn pomocí stejných REGISTER zpráv jako v případě klasické zátěže, jen s rozdílem počtu generovaných zpráv. Pomocí simulace špičkového zatížení zjistíme, jak moc se ústředna vyrovná s enormní změnou generovaných zpráv.

```
REGISTER ---------->
     100 <----------
     401 <----------
REGISTER ---------->
     200 <----------
```

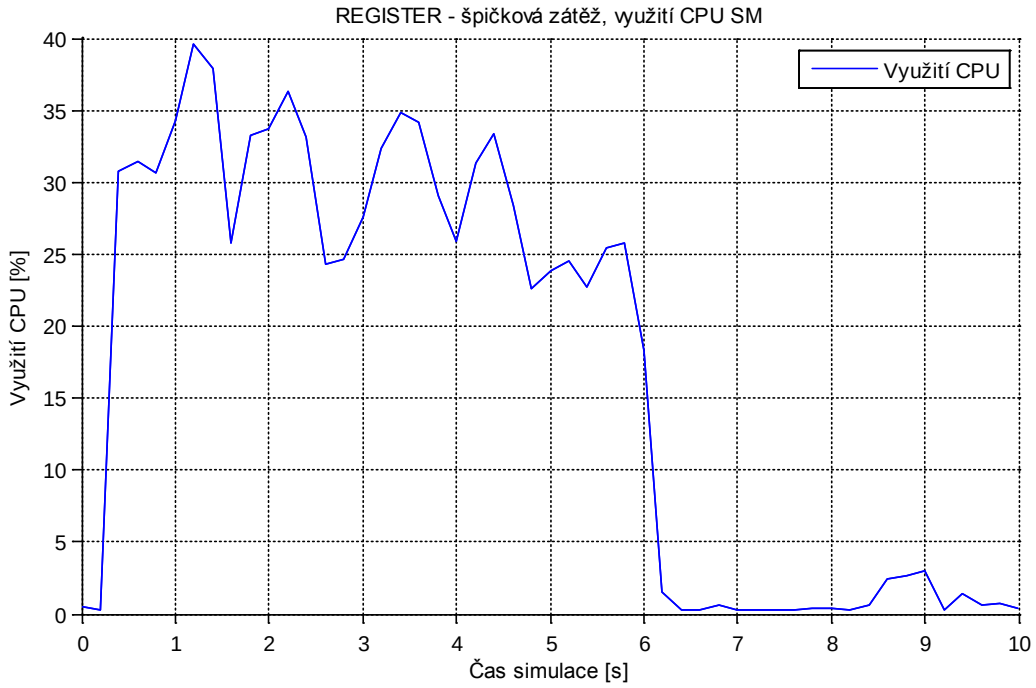
Test byl spuštěn v konfiguraci: počáteční hodnota generovaných zpráv 1000 zpráv za vteřinu a v pravidelných intervalech 10s byla tato hodnota navýšena o 5000 zpráv za vteřinu. Na konci testu byla hodnota generovaných zpráv 15500 zpráv za vteřinu.

Délka testu: 41s



Graf 5.6: Špičkové zatížení REGISTER zprávami v závislosti na čase

Z grafu 5.6 je patrné, že při generování 1000 zpráv za vteřinu Session Manager zvládá s jistými výchylkami zpracovávat požadavky na registraci. Při navýšení generování o 5000 zpráv za vteřinu už však přestane ústředna požadavky zpracovávat. SM, jak už bylo napsáno výše, zvládá podle technické dokumentace [26] 800 požadavků na registraci. Test špičkové zátěže tedy dokazuje, že tato hranice je nastavena níže, než ve skutečnosti SM zvládne obsloužit. Test trval pouhých 41s, protože ihned po překročení 6000 zpráv za vteřinu SM nezpracovával nové požadavky a změna nenastala ani u konce testu.



Graf 5.7: Vytížení CPU SM při špičkové zátěži

K porovnání vlivu špičkového vytížení generováním 5000 REGISTER zpráv za vteřinu na vytížení procesoru CPU Session Manageru je vidět na grafu 5.6.

5.3.5. Test zátěže (zapnutý firewall)

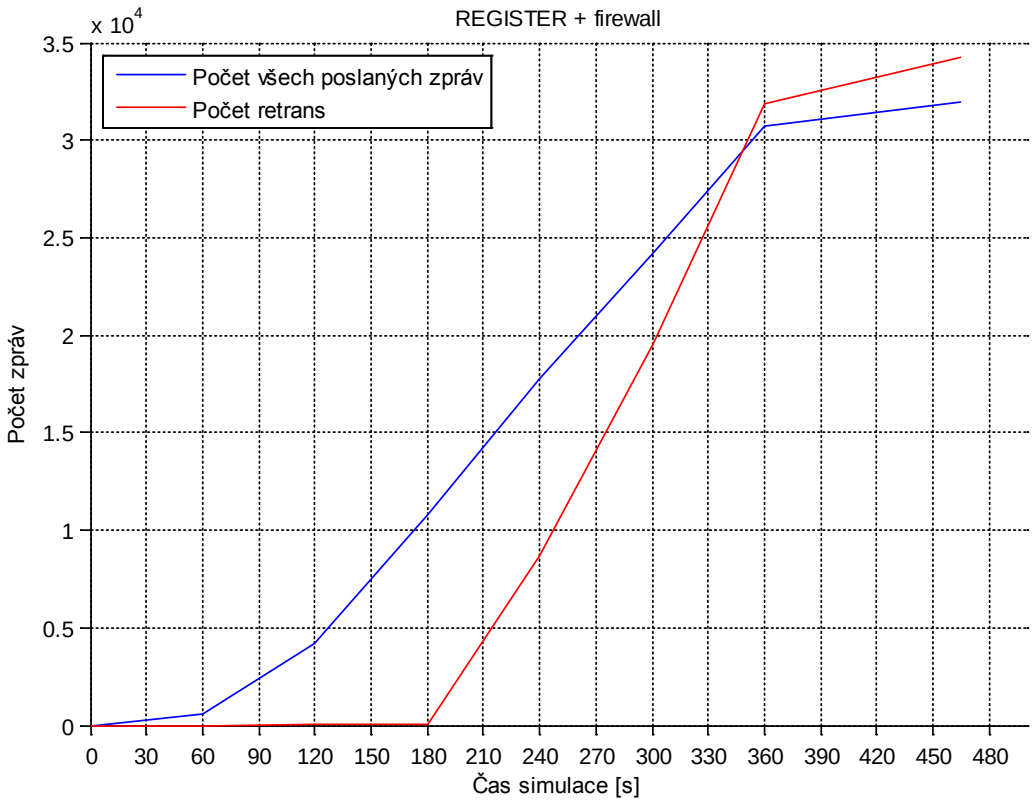
Zátěžový test se zapnutým firewallem na Session Manageru probíhal obdobně jako v případě testu s vypnutým firewallem. Tento test nebylo potřeba dělat pro delší časový úsek, protože Session Manager se zapnutým firewallem začal zahazovat zprávy ihned po dosažení hranice určené v možnostech nastavení firewallu. U všech tří následujících testů (REGISTER, INVITE, OPTIONS) se potvrdila funkčnost firewallu. Projevil se blokováním IP adresy, z které bylo generováno větší množství zpráv. Další možnosti nastavení firewallu je blokování na základě portu nebo SIP čísla. V nastavení bylo navíc možnost nastavit délku stavu blokování (*treshold*).

5.3.5.1. REGISTER

Test posílání INVITE zpráv byl nastaven na počáteční hodnotu 10 zpráv za vteřinu a každých 5s bylo generování navýšeno o 5 zpráv za vteřinu. Test byl záměrně nastaven s nízkými hodnotami generovaných zpráv z důvodu citlivějšího záznamu a přesnějšího zjištění aktivace firewallu u zpráv INVITE. Z grafu 5.9 můžeme vidět, že se firewall aktivoval po 60s, kdy bylo dosaženo potřebné množství v závislosti na nastavení firewallu. Po aktivaci firewall začalo

docházet k zahazování požadavků INVITE, kdy se generátor SIPp snažil tato požadavky poslat znovu. Došlo tedy k zvýšený počtu znovu zaslaných zpráv (*Retrans*).

Délka testu: 91s



Graf 5.8: Generované zprávy REGISTER v závislosti na čase se zapnutým firewallem

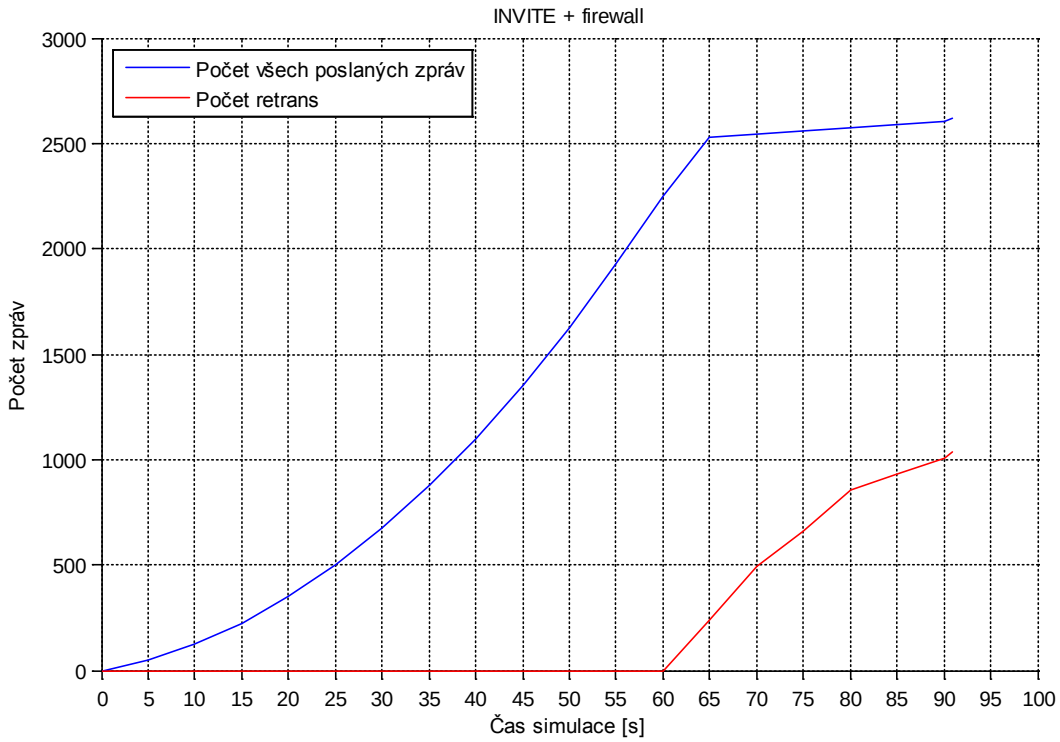
Z grafu 5.8 můžeme soudit, že funkčnost firewallu je správná a při sepnutí firewallu dochází stále ke zpracování požadavků, ovšem množství je omezeno a po určitém množství požadavků dojde k úplnému zastavení zpracování požadavků (v grafu 5.8 bod ve 360s). Firewall je nastaven tak, že dojde k blokování přijímání požadavků z IP adresy, z které bylo zaznamenáno velké množství přijatých zpráv.

5.3.5.2. INVITE

Test posílání INVITE zpráv byl nastaven na počáteční hodnotu 10 zpráv za vteřinu a každých 5s bylo generování navýšeno o 5 zpráv za vteřinu. Test byl záměrně nastaven s nízkými hodnotami generovaných zpráv z důvodu citlivějšího záznamu a přesnějšího zjištění aktivace firewallu u zpráv INVITE. Z grafu 5.9 můžeme vidět, že se firewall aktivoval po 60s, kdy bylo

dosaženo potřebné množství v závislosti na nastavení firewallu. Po aktivaci firewall začalo docházet k zahazování požadavků INVITE, kdy se generátor SIPp snažil tyto požadavky poslat znovu. Došlo tedy k zvýšený počtu znovu zaslaných zpráv (*Retrans*).

Délka testu: 91s



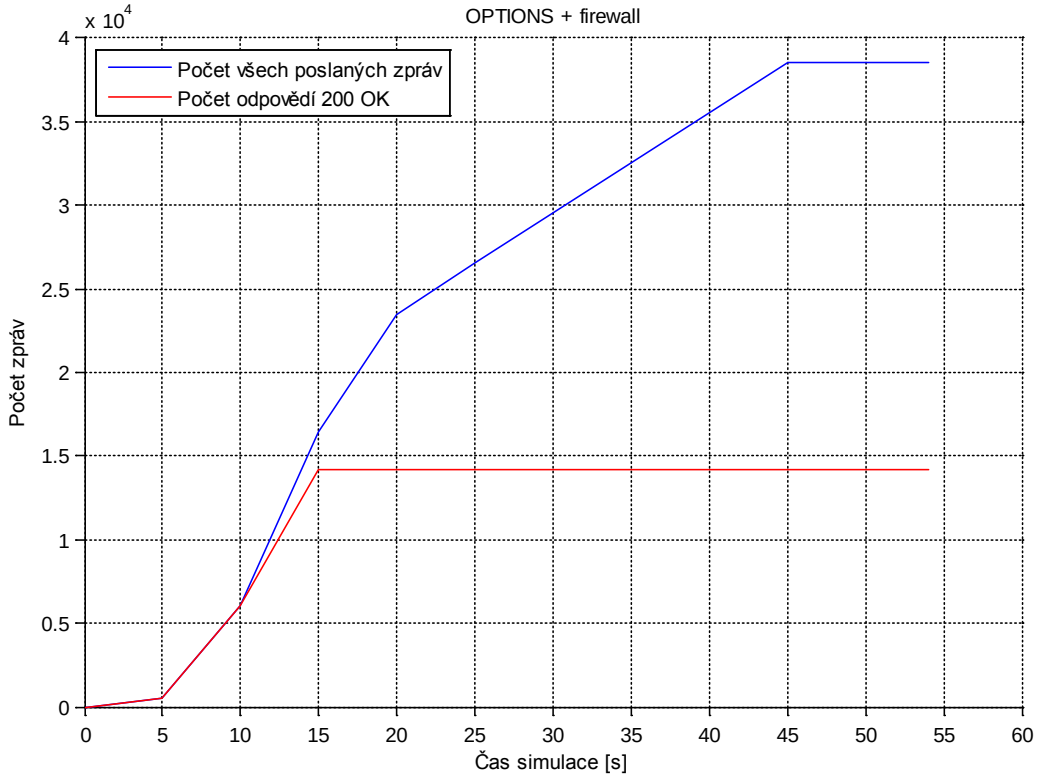
Graf 5.9: Generované zprávy INVITE v závislosti na čase se zapnutým firewallem

Test generování zpráv INVITE se zapnutým firewallem je obdobný jak je tomu u testu s REGISTER zprávami. Z grafu 5.9 jde vidět, že po dosažení počtu 50 zpráv za vteřinu dojde k aktivaci *firewallu* a dojde k novým zasíláním požadavků. Zprávy se v SM kumulují a dochází k jejich postupnému zpracování. Pří překročení 55 zpráv za vteřinu přestane SM zprávy zpracovávat a dojde k zablokování zpracování požadavků v odesílané IP adresy na dobu nastavenou v možnostech firewall.

5.3.5.3. OPTIONS

Test posílání OPTIONS zpráv byl nastaven na počáteční hodnotu 100 zpráv za vteřinu a každých 5s bylo generování navýšeno o 5000 zpráv za vteřinu. Jak je patrné z grafu 5.10, při aktivaci firewallu v čase 15s, přestaly se potvrzovat OPTIONS zprávy a nebyly posílány potvrzovací zprávy 200 OK. Session Manager přestal zpracovávat OPTIONS žádosti.

Délka testu: 54s



Graf 5.10: Generované zprávy OPTIONS v závislosti na čase se zapnutým firewallem

Průběh testu je obdobný, jako u dvou výše zmíněných (REGISTER a INVITE). Po dovršení příchozích zpráv v určité míře dojde k blokování IP adresy, z které nejsou na určitou dobu přijímány žádné požadavky. Požadavky jsou znovu zpracovány až po vypršení nastaveného *treshold*.

6. Zhodnocení

V posledních letech dochází v oblasti telekomunikací k rozmachu používání IP sítí k provozování telefonních a videohovorů. VoIP služba, tedy volání skrze IP sítí, se dostává do popředí možností komunikace. U běžných uživatelů stále hledá cestu k masovému nasazení, avšak v řadách menších i větších podniků našel VoIP svoje místo. Stalo se tak zejména díky flexibilitě použití, rychlému a snadnému přizpůsobení provozu a úspoře prostředků. S jednoduchostí využití stávajících IP sítí je tu však riziko spojené s bezpečnostními nedostatky, které můžou být spojeny právě s využíváním IP sítí. Tím, že je přenos dat v rámci komunikace prakticky celý v digitální podobě a je aplikován v rámci IP sítě, vztahují se na VoIP komunikaci prakticky podobné útoky užívané v IP sítích. Jedním s nejoblíbenějších útoků v posledních letech je bezpochyby DoS útok způsobující vyřazení určité služby.

Výsledkem práce je ověření a zhodnocení možností Avaya Aura® Session Manageru. Přes prvotní problémy s nastavením generovacího programu SIPp a přizpůsobením jeho použití v architektuře Avaya Aura® byla zprovozněna testovací topologie a otestovány nejběžnější *flood* útoky a testy zátěže validními žádostmi (REGISTER, INVITE, OPTIONS, REGISTER+INVITE). Byly zjištěny možnosti ústředny a ověřeny předpoklady uvedené v technické dokumentaci [26]. Výsledky byly velice uspokojivé, protože většina neměřených hodnoty byly o desítky procent lepší, než hodnoty uvedené v technické dokumentaci [26]. Hodnoty zátěžového testu byly získány při vypnuté ochraně pomocí firewall a jednalo se tedy o nejkritičtější možnou situaci. Výsledky jsou získána za použití horšího hardwaru pro generování. Použitá CPU pro generování zpráv byla omezujícím faktorem. Při aplikaci výkonnějšího CPU byly by výsledky určitě směrodatnější. Určitá omezení datového přenosu můžou nastat při použití pomalejší síťové karty. V testu jsme tato omezení nepozorovaly díky použití 1Gbit síťových karet. Nedoporučuje se použití pomalejších karet z důvodů popsaných v kapitole 4.2.1.

Nedílnou součástí práce je taktéž otestování Avaya Aura® Session Border Controller, který vylepšuje a nabízí dokonalejší zabezpečení architektury. V práci je ověřena funkčnost SBC a to zejména funkčnost zabezpečení pomoci firewall. V části se zapnutým zabezpečení došlo k ověření funkčnosti firewall, kdy se ochrana spustila ihned po dovršení limitu, který byl nastaven v možnostech zabezpečení. Firewall se aktivoval ihned po příjmu většího množství zpráv a potvrdila se tím funkčnost zabezpečení.

Avaya Aura® architektura pracovala naprosto spolehlivě a její nasazení do středních i větších firem je určitě zasloužené. Technické parametry uvedené v dokumentaci jsou pouze minimální a zařízení při testech popsaných v kapitole 5 pracovali s mnohem lepšími hodnotami, než je uvedeno v dokumentaci. Pro směrodatnější výsledky se určitě doporučuje použití výkonnějšího hardwaru, protože hardware PC použitých pro generování byl podstatně nižší třídy a to mohlo mít dopad na výsledky.

Celá práce je koncipována pro zhotovení metodiky testování ústředen. Program pro generování SIPp má široké pole využití, jen je potřeba přizpůsobit jeho podpůrné soubory (XML a CSV) pro různé výrobce a typy PBX. SIPp je základním kamenem pro testování provozu a zabezpečení PBX ústředen. Práce se nebere jako návod pro nelegální útočení na ústředny PBX, ale pouze jako návod na otestování vlastních řešení komunikačních prostředků pomocí PBX. Vše co bylo v práci použito je pouze pro testovací účely.

K práci jsou na CD přiloženy konfigurační soubory pro SIPp, výstupy z generátoru SIPp a technická dokumentace Avaya Aura®.

7. Použitá literatura

[1] Cesnet.cz: SIP. In: SIP [online]. 2012. Dostupné z: https://sip.cesnet.cz/cs/protokoly/sip

[2] Peterka, Jiří. eArchiv: Pohled dovnitř (VOIP). [online]. Dostupné z: http://www.earchiv.cz/b06/b0401004.php3

[3] Dočkal J., Malina R., Vaněk T., Markl J.; Management IT služeb: Bezpečnost internetové telefonie. Data Security Management [online]. 2006(6). Dostupné z: http://www.nextsoft.cz/~malina/cs/articles/voip/clanek_Bezpecnost.pdf

[4] Pužmanová, Rita. IT Point: Normalizační aktivity v IP telefonii. [online]. Dostupné z: http://www.itpoint.cz/ip-telefonie/teorie/normy-ip-telefonie.asp

[5] Thom, G.A. IEEE: H.323: the multimedia communications standard for local area networks. Communication Magazine [online]. 1996. Dostupné z: http://80.ieeexplore.ieee.org.dialog.cvut.cz/stamp/stamp.jsp?tp=&arnumber=556487&isnumber=12139

[6] VoIP Think: H.323. [online]. Dostupné z: http://www.en.voipforo.com/H323/H323_components.php

[8] Kuhn, Richard D., Thomas J. Walsh a Steffen FRIES. Security Considerations for Voice Over IP Systems: National Institute of Standards and Technology. [online]. 2005. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf

[9] Rosenberg, Martin a Mácha, Tomáš. Bezpečnost VoIP technologie. *Fakulta elektrotechniky a komunikačních technologií VUT v Brně* [online]. 2011. Dostupné z: http://www.elektrorevue.cz/cz/download/bezpecnost-voip-technologie/

[10] Dwivedi, Himanshu. Hacking VoIP: protocols, attacks, and countermeasures. San Francisco: No Starch Press, 2009. ISBN 978-1-59327-163-3.

[11] Endler, David a Mark D Collier. Hacking exposed VoIP: voice over IP security secrets. New Yrok: McGraw-Hill, 2007, xxvii, 539 s. ISBN 978-0-07-226364-0.

[12] Sisalem, Dorgham. SIP security. Chichester, U.K.: Wiley, 2009, xiv, 336 p. ISBN 04-705-1636-4.

[13] Staněk, Jan. DoS a DDoS útoky na SIP protokol [online]. Praha, 2010. 82462. Dostupné z: https://is.cuni.cz/webapps/zzp/detail/82462/. Diplomová práce. Karlova Universita v Praze. Vedoucí práce RNDr. Ing. Jiří Peterka.

[14] El-moussa F., Mudhar P., Jones A.: Overview of SIP Attacks and Countermeasures, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2010

[15] Ming Luo; Tao Peng; Leckie C.; CPU-based DoS attacks against SIP servers, Network Operations and Management Symposium, 2008. IEEE, April 2008 Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4575115&isnumber=4575087

[16] Lili Chu; Zhiming Huo; Liang Liu; The security research of SIP-based Denial of Service attack, Electrical and Control Engineering (ICECE), 2011 International Conference, Sept. 2011 Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6057845&isnumber=6056741

[17] Geneiatakis D., Vrakas N., Lambrinoudakis C.: Performance Evaluation of a Flooding Detection Mechanism for VoIP Networks, IEEE Xplore, 2009

[18] Ing. Vaněk T.: Bezpečnost provozu VoIP, odborný seminář Hotel Olšanka, ČVUT FEL, 2006

[19] Stehlík M.: Bezpečnost VoIP, Bakalářská práce, Masarykova Univerzita Fakulta Informatiky, 2008

[20] Rosenberg, Jonathan. The real time transport protocol (RTP) denial of service (DoS) attack and its prevention. [online]. 2003 [cit. 2014-11-02]. Dostupné z: http://tools.ietf.org/pdf/draft-rosenberg-mmusic-rtp-denialofservice-00.pdf

[21] Schulzrinne H., Casner S., Frederick R., Jacobson V.: RTP: A Transport Protocol for Real-Time Applications, RFC3550, Packet Design, July 2003

[22] Sulkin, Allan. The Secrets of Avaya Aura Session Manager Application Sequencing Revealed. In: [online]. 2009 [cit. 2014-11-23]. Dostupné z: http://www.nojitter.com/post/224900594/the-secrets-of-avaya-aura-session-manager-application-sequencing-revealed

[23] Gayraud R., Jacques O. et al.: SIPp traffic generator for the SIP protocol, Dostupné z: http://sipp.sourceforge.net

[24] Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley a E. Schooler. SIP: Session Initiation Protocol. In: [online]. 2002. Dostupné z: http://www.ietf.org/rfc/rfc3261.txt

[25] Avaya, Security Design in Avaya Aura® Session Manager. [online] 2013. Dostupné z: http://downloads.avaya.com/css/P8/documents/100168178

[26] Avaya, Avaya Aura® Session Manager Overview and Specification. [online] 2014. Dostupné z: http://downloads.avaya.com/css/P8/documents/100168154

[27] Avaya, Avaya Aura® The communications infrastructure for people-centric collaboration. [online] 2012. Dostupné z:

https://www.devconnectprogram.com/fileMedia/download/9b7542d4-81a9-4dfa-b61a-d8f86c86a607

[28] Avaya, Avaya Aura® Communication Manager. [online] 2011. Dostupné z: https://www.devconnectprogram.com/fileMedia/download/d328ed0d-c90d-43fd-9301-82f385b3c283

[29] Avaya, Avaya Aura® System Manager Overview and Specification. [online] https://www.devconnectprogram.com/fileMedia/download/b7621f6b-9ef3-498b-acb4-ccd9d2d59ba7

[30] Avaya, Avaya Session Border Controller for Enterprise. [online] 2014. Dostupné z: http://www.avaya.com/cs/Satellite?blobcol=urldata&blobkey=id&blobtable=MungoBlobs&blobwhere=1380941517326&ssbinary=true

8. Seznam obrázků

9. Seznam grafů

10. Seznam tabulek

11. Seznam zkratek

API	*Application Programming Interface*
ARP	*Address Resolution Protocol*
Caller ID	*Caller Identification*
CPU	*Central Processing Unit*
DoS	*Denial of Service*
DTMF	*Dual-Tone Multi-Frequency*
HTTP	*HyperText Transport Protocol*
ICMP	*Internet Control Message Protokol*
IDS	*Intrusion Detection System*
IPS	*Intrusion Prention System*
ITU-T	T*elecommunication Standardization Sector of ITU*
MitM	*Man-in-the-Middle*
NAT	*Network Address Translation*
OS	*Operační systém*
PBX	*Private branch exchange*
POD	*Ping of Death*
PSTN	*Public* Switched Telephone Network
RFC	*Request for Comments*
RTP	*Real-time Transport Protocol*
RTCP	*Real-time Transport Control Protocol*
RTSP	*Real-Time Streaming Protocol*
SBC	*Session Border Controller*
SDP	*Session Description Protocol*
SIP	*Session Initiation Protocol*
SM	*Session Manager*
SMTP	*Simple Mail Transfer Protocol*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
UA	*User Agent*
UAC	*User Agent Client*
UAS	*User Agent Sever*
UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
VoIP	*Voice Over IP*

12. Obsah přiloženého CD

- Diplomová práce
    - kovar_ondrej.pdf
- Konfigurační schémata pro SIPp
    - alg_hovor.xml
    - alg_invite.xml
    - alg_options.xml
    - alg_register.xml
    - alg_register_invite.xml
- Konfigurační soubory uživatelů pro SIPp
    - alg_users_50c.csv
    - alg_users_50d.csv
- Výstupní soubory ze SIPp
    - hovor_noFW_counts.xlsx
    - invite_FW_counts.csv
    - invite_noFW_counts.csv
    - options_FW_counts.csv
    - options_noFW_counts.csv
    - register_FW_counts.csv
    - test_zátěže_5K+5K_noF_counts.csv
    - register_invite_noFW_counts.csv
    - register_noFW_counts.csv
    - test_zátěže_5K+5K_noFW_counts.csv
- Vytížení CPU
    - register_noF_10k.xlsx
- Technická dokumentace
    - SM-OverviewAndSpecificationIssue5.pdf