Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Detekce síťových útoků na Voice over IP infrastrukturu

Bc. Nikolas Jíša

Vedoucí práce: Ing. Tomáš Čejka

5. května 2015

Poděkování

Rád bych poděkoval svému vedoucímu za ochotu a pomoc a také své rodině za podporu při tvorbě této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 5. května 2015 .....

Odkaz na tuto práci

Jíša, Nikolas. *Detekce síťových útoků na Voice over IP infrastrukturu.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Obrovský nárůst síťového provozu vyžaduje zvyšující se nároky na bezpečnost kvůli rizikům finančích ztrát a narušení soukromí. V oblasti Voice over Internet Protocol (VoIP) mohou finanční ztáty dosáhnout i miliónů českých korun. Tento text se zabývá bezpečností VoIP, zejména protokolu SIP se zaměřením na útoky typu skenování a útoky typu Denial of Service (DoS). Útoky typu skenování použije útočník k získání informací o síti, o zařízeních podporujících SIP a o SIP uživatelích, které dále použije k dalším útokům. Útoky typu DoS může útočník použít ke zpomalení nebo k znemožnění zpracovávání SIP zpráv atakovaného zařízení. Detekce útoků je zde založena na analýze časových řad.

Klíčová slova SIP, VoIP, CPD, DoS, skenovací útoky, NP-CUSUM, EWMA, Nemea, TRAP, UniRec

Abstract

A huge increase of network traffic requires increasing security because of risks of possible financial losses and cases of privacy breach. In Voice over Internet Protocol (VoIP) technology the financial losses can be significant. This text is aimed on security of VoIP particularly security of SIP against scanning attacks and Denial of Service (DoS) attacks. Scanning attacks can be used

to get information about network, about devices supporting SIP and about SIP users. DoS attacks can be used to slow down or prevent processing of SIP messages on an attacked device. The detection procedure is based on time series analysis.

Keywords SIP, VoIP, CPD, DoS, scanning attacks, NP-CUSUM, EWMA, Nemea, TRAP, UniRec

Obsah

Seznam obrázků

Seznam tabulek

Úvod

S obrovským nárůstem síťového provozu a přenosů citlivých informací se bezpečnost stává stále důležitější. Ačkoliv existuje mnoho technik zabezpečení, jako je například šifrování nebo autentizace, ne vždy se tyto techniky používají a ne vždy je vhodné je použít. Navíc existují síťové útoky, proti kterým tyto techniky nejsou účinné. Kromě toho se síťové útoky stávají stále rafinovanější, zatímco znalostí, kterými útočník musí k provedení útoku disponovat, ubývá, což je způsobeno volně dostupnými programy, které umožňují snadné provádění útoků. U Voice over IP (VoIP) můžeme narazit na celou řadu útoků, které mohou přímo či nepřímo způsobit velké škody.

Diplomová práce má dvě hlavní části. V první části se budu zabývat protokolem Session Initiation Protocol (SIP). Popíšu, k čemu se používá, jaké používá zprávy a hlavně se zaměřím na útoky přes protokol SIP. Útoky, které popíšu, jsou: Denial of Service (DoS) útoky, skenovací útoky, útoky založené na modifikaci či podvrhování zpráv, autentizační útoky, Spam over Internet útoky (SPIT) a také zmíním útok založený na neoprávněném volání do veřejné telefonní sítě (PSTN).

Druhá část je zaměřena na zvolení vhodných metod detekce skenovacích a DoS útoků přes SIP. Nejprve zde zmíním, jak probíhá monitorování a analyzování dat v síti CESNET2 pomocí systému Nemea. Dále se budu zabývat metodami detekce útoků, kde se zaměřím na metody Non-parametric Cumulative Sum (NP-CUSUM) a Exponentially Weighted Moving Average (EWMA). Hlavním cílem této práce je pak návrh modulu do systému Nemea, který bude umět detekovat skenovací a DoS útoky přes SIP, jeho implementace a testování.

Kapitola 1

Session initiation protocol

1.1 Přehled funkcionality

Tato sekce čerpá hlavně z RFC 3261 [4].

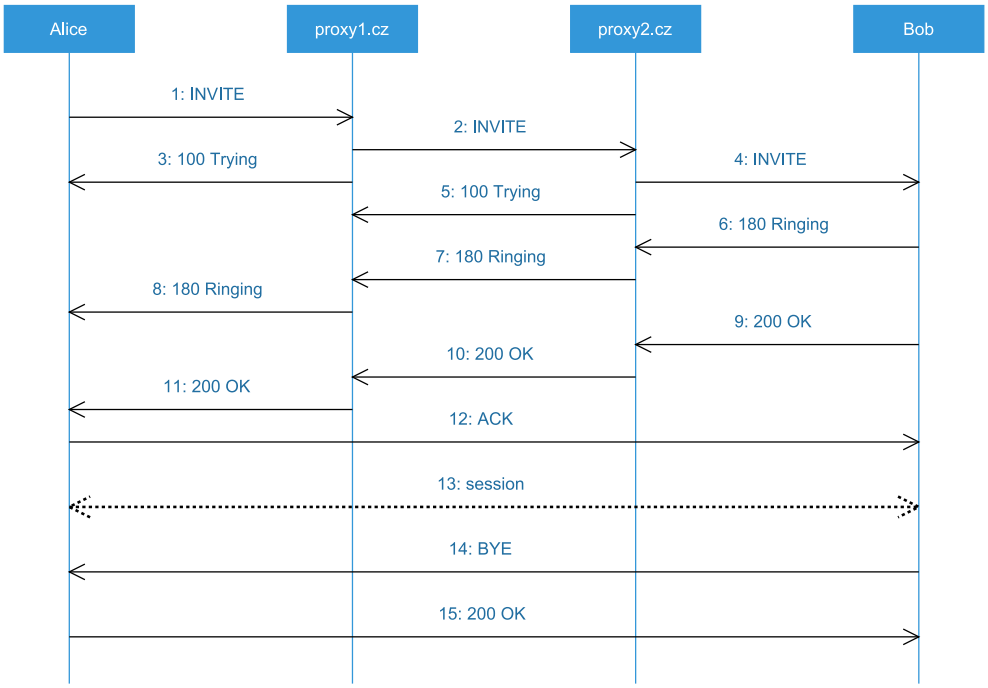
Session initiation protocol (SIP) je protokol aplikační vrstvy, pomocí kterého lze založit, modifikovat a ukončit multimediální relace přes síť, což jsou nejčastěji internetové telefonní hovory, internetové telefonní konference nebo internetové videohovory.

SIP není integrovaný komunikační systém. SIP je spíše komponenta, kterou lze použít s jinými IETF protokoly k vytvoření kompletní multimediální architektury. Většinou tyto architektury budou obsahovat protokoly jako jsou:

- Real-Time Transport protokol (RTP) pro přenosy real-time dat a poskytování zpětné vazby QoS.
- Real-Time streaming protokol (RTSP) pro řízení doručování streamovacích dat.
- Media Gateway Control Protocol (MEGACO) pro řízení bran do Public Switched Telephone Network (PSTN).
- Session Description Protocol (SDP) pro popisy SIP relací.

Pro poskytnutí kompletních služeb by tedy SIP měl být používán s ostatními protokoly najednou, nicméně základní funkcionalita protokolu SIP na těchto protokolech nezávisí.

SIP identifikuje uživatele pomocí SIP URI, což můžeme chápat jako telefonní číslo nebo e-mailovou adresu (např. sip:alice@proxy1.cz), která používá hierarchickou doménu. Pro směrování SIP zpráv mezi zařízeními koncových uživatelů a doménami existují SIP proxy servery, které využívají služby jako DNS. Při směrování z domén do poddomén či z domén koncovým uživatelům jsou také nutné lokalizační služby, které Request URI, což je SIP URI uvedené



Obrázek 1.1: Příklad životního cyklu SIP relace

jako cíl v SIP zprávě (viz 1.3), spojí s nějakou adresou. Tuto asociaci SIP URI s adresou lze nastavit manuálně, nicméně SIP poskytuje k tomuto účelu možnost, která probíhá pomocí REGISTER zprávy (viz SIP požadavky).

SIP také zavádí SIP Secure URI (SIPS), které navíc zaručuje všem SIP relacím, u kterých je dané SIPS URI destinací, bezpečný a šifrovaný provoz pro všechny zprávy, které jsou tomuto SIPS URI doručovány. Použití SIPS URI ovšem neznamená šifrování dat SIP relace, jako jsou například data samotných hovorů.

1.2 Příklad životního cyklu SIP relace

Tato sekce čerpá z RFC 3261 [4].

Tato sekce popisuje příklad průběhu vytvoření a ukončení SIP relace pro telefonní hovor mezi Alicí a Bobem přes Internet viz obrázek 1.1. Alice používá SIP aplikaci na jejím stolním počítači, zatímco Bob používá internetový SIP telefon. Alice používá SIP proxy proxy1.cz a Bob používá proxy proxy2.cz.

Když chce Alice zavolat Bobovi, dojde k těmto operacím:

- Alice odešle zprávu INVITE na proxy1.cz.

- Jakmile proxy1.cz obdrží zprávu INVITE, DNS dotazem zjistí IP adresu proxy2.cz (pokud ji už nemá v nějaké cache). Následně proxy1.cz přepošle INVITE zprávu na proxy2.cz a potom odešle zprávu 100 Trying Alici.
- Jakmile proxy2.cz obdrží zprávu INVITE, pak proxy2.cz přepošle zprávu Bobovi (předpokládáme, že proxy2.cz zná IP adresu, na které se Bob nachází). Následně proxy2.cz odešle zprávu 100 Trying na proxy1.cz.
- Po obdržení zprávy INVITE Bob odpoví zprávou 180 Ringing Alici přes proxy2.cz a proxy1.cz.
- Po přijmu hovoru od Alice, odešle Bob zprávu 200 OK přes proxy2.cz a proxy1.cz Alici.
- Alice poté, co obdrží zprávu 200 OK, odešle Bobovi zprávu ACK. Zpráva ACK je však nyní odeslána podle směrovacích protokolů nižších vrstev přímější cestou. Zpráva ACK tedy nemusí být zaznamenána na SIP proxy serverech.
- Nyní dojde k samotnému hovoru.
- Později Bob zavěsí sluchátko, čímž odešle Alici zprávu BYE (opět nemusí jít přes SIP proxy servery).
- Po příjmu zprávy BYE Alice odešle zprávu 200 OK (opět nemusí jít přes SIP proxy servery).

1.3 SIP zprávy

Tato sekce čerpá z RFC 3261 [4].

Tato sekce popisuje zprávy, které SIP používá, což jsou buďto SIP požadavky, nebo SIP odpovědi.

1.3.1 SIP požadavky

SIP podporuje 6 typů požadavků:

- **REGISTER**

  Požadavek REGISTER slouží k tomu, aby se SIP uživatel přihlásil do ústředny a byl potom schopen přijímat a odesílat SIP zprávy.

- **INVITE**

  Tímto požadavkem odesílající SIP uživatel říká, že chce se SIP uživatelem, uvedeným v položce Request-URI vytvořit SIP relaci.

- **ACK**

  Tuto zprávu odesílá SIP uživatel, který odeslal zprávu INVITE poté, co obdrží odpověď, kterou SIP uživatel, který obdržel zprávu INVITE, potvrzuje, že přijal požadavek INVITE. Následné obdržení této zprávy potvrzuje, že je SIP relace vytvořena.

- **CANCEL**

  Požadavkem CANCEL lze v některých případech zrušit předchozí požadavek. Zpravidla se CANCEL používá ke zrušení INVITE požadavku. Pokud již na předchozí požadavek přišla odpověď jiná než 1xx (viz SIP odpovědi), je požadavek CANCEL ignorován.

- **BYE**

  Zpráva BYE je odeslána ve chvíli, kdy chce jeden z účastníků SIP relace z relace odejít.

- **OPTIONS**

  Požadavek OPTIONS slouží ke zjištění informací o SIP uživateli nebo o SIP zařízení, jako jsou podporované protokoly a kodeky.

Existuje ještě několik dalších typů SIP požadavků, které jsou specifikovány v rozšířeních RFC3261 [4].

1.3.2 SIP odpovědi

Většina SIP požadavků očekává SIP odpověď. Každý typ SIP odpovědi má svůj stavový kód, což je trojciferné číslo. Existuje 6 tříd SIP odpovědí, které jsou odlišeny první cifrou stavového kódu:

- **1xx: Prozatimní odpověď**

  Signalizuje, že je SIP požadavek dále zpracováván. V této kategorii jsou například tyto zprávy:

  - **100 Trying**: Signalizuje, že byl SIP požadavek obdržen a je zpracováván.
  - **180 Ringing**: SIP uživatel, který obdržel INVITE požadavek touto odpovědí informuje SIP uživatele, který INVITE požadavek odeslal, že mu začal vyzvánět SIP telefon.

- **2xx: Úspěch**

  Odpověď 2xx znamená, že SIP požadavek byl úspěšně obdržen a zpracován. Patří sem zpráva **200 OK**.

- **3xx: Přesměrování**

  Tato odpověď říká, že je nutné provést další operaci před tím, než může být požadavek vykonán.

- **4xx: Chyba klienta**

  Odpověď, která signalizuje chybu na straně klienta, čímž rozumíme SIP uživatele, který odeslal SIP požadavek. Patří sem například zprávy:

  - **400 Špatný požadavek**: Požadavek nemohl být zpracován kvůli špatné syntaxy požadavku.
  - **401 Neautorizován**: Požadavek vyžaduje autentizaci.
  - **403 Nepovoleno**: Odmítnutí vykonání požadavku.
  - **404 Nenalezeno**: Uživatel neexistuje v doméně uvedené v Request-URI.

- **5xx: Chyba serveru**

  Tato odpověď signalizuje, že došlo k chybě na straně serveru, čímž rozumíme SIP uživatele nebo SIP zařízení, které reaguje na požadavek.

- **6xx: Globální chyba**

  Odpověď 6xx znamená, že požadavek nemohl být vykonán.

1.4 Útoky přes SIP

Tato sekce popisuje některé útoky, které lze přes protokol SIP provádět. Nejsou zde příliš rozebírány útoky, které lze provést přes nižší vrstvy ISO/OSI.

1.4.1 Denial of Service útoky

Denial of Service útoky (DoS) se pokouší způsobit nestabilitu služeb, kvůli které služba přestane správně fungovat. DoS většinou probíhá odesíláním velkého množství požadavků na atakované zařízení. Každý z těchto požadavků způsobí na atakovaném zařízení alokaci systémových prostředků, jako je operační paměť nebo procesor, na dostatečně dlouhou dobu a eventuelně atakované zařízení přestane být schopno zpracovávat nové požadavky. U SIP lze tohoto útoku docílit odesíláním velkého počtu INVITE či REGISTER požadavků. Obranou proti těmto útokům může být limitování počtu SIP požadavků, které může jedna IP adresa mimo naší síť do naší sítě odeslat v jednom časovém okénku. Při překročení námi stanoveného limitu můžeme třeba nastavit, aby zprávy od dané IP adresy na nějaký čas byly ignorovány. Tento obranný mechanismus nemusí být účinný pro distribuované verze útoků DoS, u kterých dochází k útoku z mnoha IP adres. Navíc je přes SIP možné podvrhovat zdrojové IP adresy, je tedy vhodné zavést nějakou autentizaci.

1.4.2 Skenování

Skenování je způsob, kterým se útočník snaží získat informace o službách běžících na vybraných nebo náhodných zařízeních se záměrem tyto informace použít k nějakému útoku. V Internetu můžeme skenování docílit posíláním paketů na vybrané nebo náhodné IP adresy s cílovými porty, které jsou standardní pro služby, na které chceme dále útočit. Pokud IP adresa na naší zprávu odpoví, můžeme zkoušet odesílat další zprávy specifické pro službu, o které chceme získat informace, a informace získané z odpovědí pak použít k útoku.

U protokolu SIP bude prvním krokem skenování nalézt zařízení, která podporují SIP. K tomu nám může dobře posloužit požadavek OPTIONS, případně požadavek REGISTER. Už jen informace, že nějaké zařízení podporuje SIP, může postačovat k provedení některých útoků.

V [1] byl pozorován síťový SIP provoz na honeynetech. Honeynet je síť, která se skládá z honeypotů. Jako honeypot budu v této práci chápat zařízení, které se více či méně chová, jakoby poskytovalo nějakou službu (nebo služby), na které chceme, aby byly prováděny útoky z reálného provozu s účelem získat o útocích informace [5].

U pozorování honeynetů v [1] byl vytvořet honeynet systém dvou honeynetů, který byl umístěn na Internet do veřejných sítí třídy C - honeynet A obsahoval čtyři SIP komponenty a honeynet B neobsahoval žádné SIP komponenty. Informace o poskytovaní SIP služeb nebyly nikde zveřejněny, jakýkoliv SIP provoz s honeynety bude tedy podezřelý.

Z hlediska skenování byly v [1] zpozorovány zajímavé výsledky. Na obrázku 1.2 jsou zobrazeny počty SIP požadavků za den, které do honeynetů přišly v období od prosince 2010 do konce ledna 2012 (osa y má logaritmické měřítko). V honeynetu A byly nejprve čtyři honeypoty a 17. května 2011 došlo k přidání dalšího honeypotu, což je v grafu 1.2 znázorněno jako masivní nárůst SIP zpráv, které přišly do honeynetu A. Také je v grafu 1.2 vidět, že ačkoliv v honeynetu B není žádná SIP komponenta, skenovaní pomocí SIP zpráv se pořád opakuje, i pokud nedojde k nalezení žádné SIP komponenty. Tabulka 1.1 zobrazuje zastoupení napozorovaných zpráv. Do honeynetu A chodí většinou požadavky na přihlášení (REGISTER), což by mohlo znamenat, že se někdo snaží uhádnout přihlašující údaje. Skenování hledáním SIP zařízení pomocí OPTIONS probíhá ve výrazně menší míře. Naopak na honeypot B přišly hlavně požadavky OPTIONS.

Tabulka 1.1: Zastoupení napozorovaných SIP požadavků

SIP zpráva	Honeynet A	Honeynet B	Obě sítě
OPTIONS	0.02%	97.2%	2.8 %
REGISTER	98.6%	0.7%	95.8%
ACK/BYE/INVITE	1.3%	2.1%	1.4%

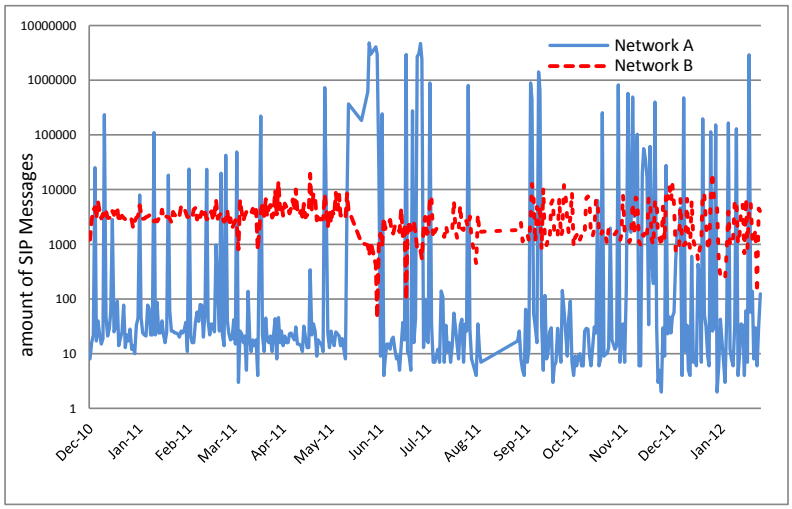


Figure 4: SIP messages per day (Dec 10 - Jan 12)

Obrázek 1.2: Počet SIP požadavků, které přišli na honeynety, mezi prosincem 2010 a lednem 2012. Převzato z [1]

1.4.3 Modifikace zpráv

Jelikož přenos SIP zpráv probíhá sám o sobě v otevřené formě (pokud není používáno SIPS URI nebo jiného zabezpečení pomocí TLS/SSL), je možné, aby jakýkoliv síťový prvek, přes který projde SIP zpráva, danou zprávu přečetl, modifikoval a až potom přeposlal. Útočník takto může změnit některé parametry navazované SIP relace či její popis, čímž může provést například útok Man in the Middle [6]. Také se může pokusit například o útok typu přetečení bufferu [6].

U útoku Man in the Middle (MitM) dochází k tomu, že se mezi dva komunikující subjekty dostane útočník, který ovládne všechny zprávy, které si komunikující subjekty vymění. Útočník tedy bude moci každou zprávu přečíst, modifikovat a teprve potom přeposlat druhému subjektu, bez toho aby si komunikující subjekty něčeho všimly.

Útok přetečení bufferu spočívá v tom, že útočník odešle zprávu, na základě které program, který zprávu obdrží, způsobí spuštění škodlivého kódu. Předpokladem tohoto útoku je chyba v programu nebo operačním systému, která způsobuje zranitelnost vůči tomuto typu útoků. Ačkoliv se nejedná o jednoduchý útok, útočník může tímto útokem dosáhnout velkých úspěchů. Riziko, že má aplikace nebo operační systém, který používáme, zranitelnost na přetečení buferu, můžeme zmírnit pravidelným instalováním aktualizací.

Nejúčinnější obranou proti modifikacím zpráv je šifrování.

1.4.4 Podvrhování SIP zpráv

Tato podsekce čerpá z [6].

SIP povoluje zpracování jakékoliv zprávy bez validace a autentizace. Většině útoků tohoto typu se lze vyhnout používáním autentizace nebo šifrováním SIP zpráv. Podvrhování SIP zpráv lze použít k:

- **Deautentizačnímu útoku**

  Útočník podvrhne zprávu REGISTER a nastaví jí položku *Expire* na 0. Taková zpráva bývá odeslána registrační službě při vypnutí SIP zařízení. Důsledkem je, že uživatel nebude moci přijímat SIP zprávy a ani je odesílat do té doby, než jeho SIP zařízení znovu odešle zprávu REGISTER se správnými údaji.

- **Předčasnému ukončení probíhajícího hovoru**

  Útočník podvrhne zprávu BYE, která ovšem musí mít správně nastavené některé SIP hlavičky. Důsledkem je neočekávané ukončení probíhající SIP relace.

- **Modifikace probíhající SIP relace**

  Útočník podvrhne INVITE zprávu právě probíhající SIP relace a změní některé její parametry, jako je třeba parametr *To* nebo *From* (viz [4]), což může způsobit odmítnutí služby.

1.4.5 SIP autentizační útoky

Autentizačním útokem zde chápeme případ, kdy se útočník snaží neoprávněně přihlásit do systému. V [1] je autentizační útok rozdělen do těchto fází:

1. **Skenování SIP komponent**

   V první fázi dojde k vyhledání zařízení, které poskytují služby registrace přes SIP (dále SIP registrační zařízení), pomocí OPTIONS požadavků (jak jsem již zmínil v 1.4.2).

2. **Nalezení přihlašovacího jména**

   Jakmile útočník najde SIP registrační zařízení, přes které se lze přihlásit, začne zjišťovat přihlašovací jména uživatelů pomocí SIP požadavků REGISTER. Pokud uživatelské jméno, které útočník uvedl v požadavku REGISTER, neexistuje, SIP registrační zařízení odpoví zprávou 404 NOT FOUND. V opačném případě SIP zařízení odpoví zprávou 401 UNAUTHORIZED nebo 403 FORBIDDEN. Výstupem této fáze může být kompletní seznam existujících uživatelů, pod kterými se lze přes nalezené SIP registrační zařízení přihlásit.

3. **Neoprávněné přihlášení**

   Dále se útočník pokouší uhádnout heslo (resp. hesla) nějakého uživatele (resp. uživatelů). Hádání hesel lze provádět kombinací dvou způsobů [1]:

   - Slovníkové hádání hesel: Zde útočník zkouší kombinace různých existujících slov.
   - Enumerativní hádání hesel: V tomto případě útočník systematicky zkouší kombinace znaků, které nejsou příliš dlouhé, z nějaké množiny znaků (např. zkoušení všech čísel od 0 do 9999).

4. **Další útok**

   Jakmile je útočník přihlášen, může provádět další útoky.

1.4.6 SPIT

Zkratka SPIT znamená spamy přes internetové telefony (spam over Internet telephony). Typicky si pod termínem spam představíme hlavně nevyžádanou emailovou poštu. Mezi spamy ovšem můžeme zařadit také nevyžádané telefonní hovory. Provoz protokolu SIP je z hlediska spamů podobnější emailovým službám než tradičnímu telefonnímu provozu přes PSTN (Public switched telephone network). Důvodem je, že odesílání spamů přes PSTN je o několik řádů nákladnější, než je tomu u VoIP, a VoIP navíc poskytuje skrytí identity [2]. Detekce emailových spamů se od detekce spamů přes VoIP liší hlavně tím, že u VoIP dochází k real-time přenosu dat [3]. Spamy přes SIP lze rozdělit podle [2] na:

- **Pasivní marketing**

  Tento typ spamu spočívá v odesílání nahraných hlasových zpráv.

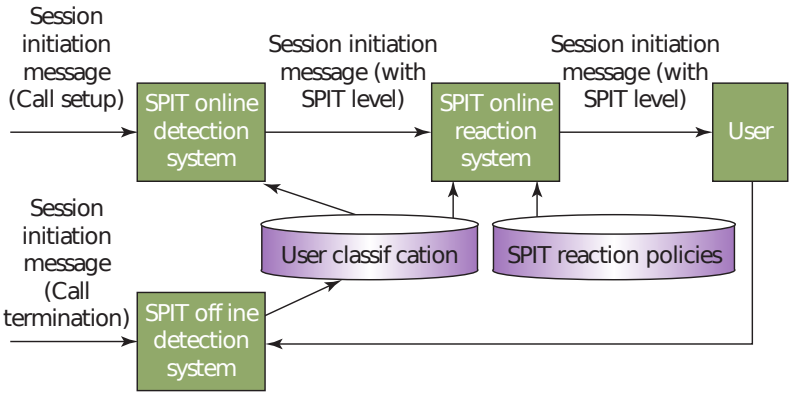
- **Interaktivní marketing**

  U tohoto typu se volající snaží prodat služby nebo produkty volanému.

- **Call back**

  Tento typ spamu spočívá v tom, že volající ukončí volání předtím, než volaný přijme hovor. Volaný poté ze zvědavosti zavolá zpátky, ale bez vědomí, že volá na draze zpoplatněnou linku. Zde lze navíc využít trik, kdy volaný přijme hovor, ale generuje vyzváněcí tón, volající si tedy může myslet, že volanému zvoní telefon, zatímco hovor už probíhá.

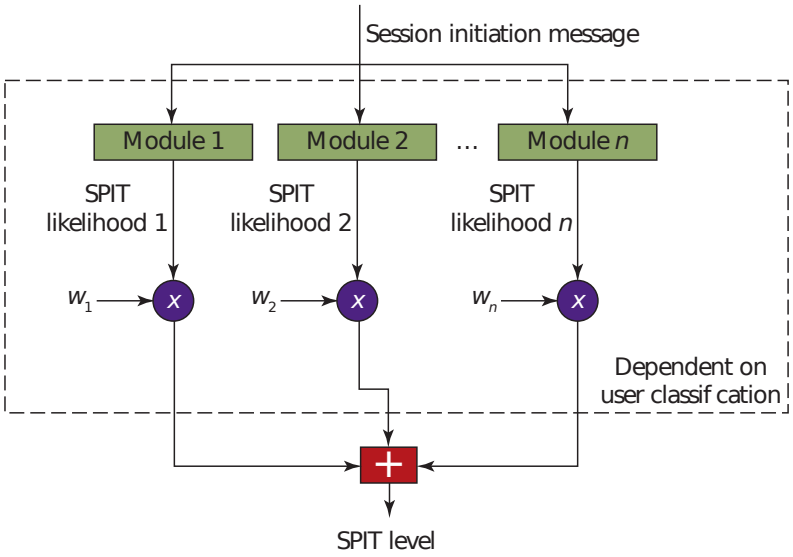
V [2] popisují systém SDRS (SPIT Detection and Reaction System) pro detekci SPIT incidentů a reakci na detekované SPIT incidenty (viz obrázek 1.3). SIP požadavky jsou nejprve doručeny do SPIT online detekčního systému (viz obrázek 1.4). SPIT online detekční systém obsahuje několik modulů. Každý



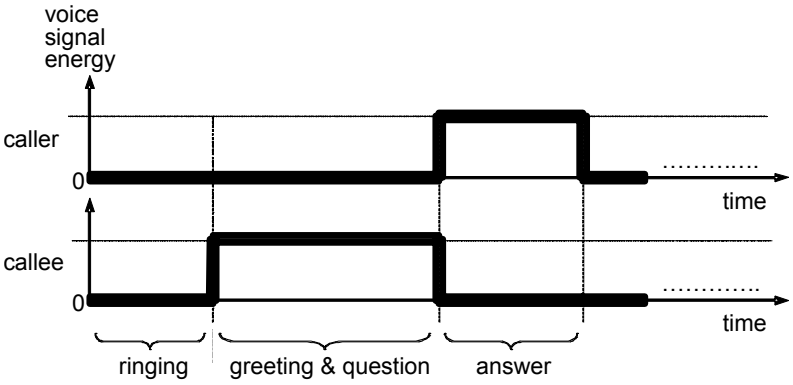
Obrázek 1.3: SPIT Detection and Reaction System (SDRS). Převzato z [2]

modul zpracovává nějakým algoritmem naměřené hodnoty statistik, na základě kterých vyhodnotí pravděpodobnost, že se jedná o pokus o SPIT. Mezi pozorovanými statistikami jsou: počet hovorů, kterých se volající účastní najednou, počet hovorů iniciovaných volajícím a počet chybových zpráv asociovaných s volajícím. Každý modul má na základě uživatelských preferencí váhu. S výstupy všech modulů je pak proveden vážený průměr, který určuje, zda se jedná o SPIT incident. Na základě této pravděpodobnosti rozhodne SPIT online reakční systém, jakou akci podnikout (tj. zda hovor přijmout, zda uživatele nějak upozornit na podezřelost hovoru nebo zda hovor blokovat). Po příjmu hovoru je dále prováděna analýza SPIT offline detekčním systémem, který pozoruje statistiky o SIP hovorech. SPIT offline detekční systém může ovlivnit online SPIT detekci u dalších hovorů od volajícího. U offline detekce je pozorována hlavně délka hovorů, jelikož je očekáváno, že u spamů budou mít hovory podobné délky.

V [3] popisují mechanismy detekce SPIT hovorů založených na hledání znaků přirozených lidských hovorů v datech hovorů. K tomuto účelu používají Turing testy [7]. Turing testy se snaží zjistit, zda je komunikující subjekt člověk nebo jenom stroj. Klasické Turing testy spočívají v tom, že je volajícímu položena nějaká otázka, která očekává nějakou množinu odpovědí. Ačkoliv klasické Turing testy většinou správně rozliší člověka od stroje, u validních volání bude nejspíše otravné odpovídat na testovací otázky. [3] se snaží přijít s takovými Turing testy, které nepotřebují explicitní interakci s komunikujícími lidmi. K tomuto účelu jsou zde definovány čtyři stavy hovoru (mezi dvěma subjekty): Stav M (mutual silence) - ani jeden subjekt nemluví, stavy A a B - když mluví jeden subjekt a stav D (double talk) - mluví oba subjekty najednou. ITU [8] uvádí, že hovory v angličtině, italštině a japonštine jsou průměrně 6.59% svého trvání ve stavu D a průměrně v tomto stavu zůstanou 0.23 sekund. Je očekáváno, že u SPIT hovorů iniciovaných strojem bude hovor ve stavu D déle. V navrženém principu je očekáván průběh hovorů, jako



Obrázek 1.4: SPIT online detekční systém SDRS. Převzato z [2]



Obrázek 1.5: Očekávaný průběh telefonního hovoru mezi 2 subjekty. Převzato z [3]

je zobrazen na obrázku 1.5: Poté, co volaný zvedne telefon, je očekáváno, že volaný pozdraví volajícího a položí volajícímu krátkou otázku, na kterou následně volaný odpoví. Princip v [3] se snaží najít hovory, kterých průběh se od uvedeného liší, přičemž je ovšem nutné vzít v úvahu různé jazykové kultury a různé zvuky z okolí. Uvedený způsob dovede rozlišit člověka od stroje, nicméně prozatím je člověk po příjmu hovoru většinou schopen velmi rychle vyhodnotit sám, zda mu volá člověk nebo jenom stroj.

1.4.7 Neoprávněné volání do sítě PSTN

Tato podsekce čerpá z [9].

Některé VoIP ústředny mají nastavenou možnost volání do zpoplatněné PSTN sítě. Pro volání do PSTN má zpravidla VoIP ústředna nastaveno předčíslí pro položku Request-URI INVITE požadavku. Ústředna bude většinou poskytovat přístup do PSTN pouze zařízením, která jsou v lokálních sítích.

Útok může probíhat v těchto fázích:

1. Útočník si založí telefonní číslo určené pro placené služby. Hovory na toto číslo budou tedy útočníkovi generovat zisky.
2. Ovládnutí nějakého počítače (počítačů) v atakované síti například pomocí malwaru.
3. Z ovládnutých počítačů se dále útočník snaží zjistit předčíslí do PSTN.
4. Po uhodnutí předčíslí jsou dříve nebo později provedeny hovory do PSTN na útočníkovo placené číslo, což způsobí generování zisků útočníkovi (nepřímo) na úkor atakované organizace.

Kapitola 2

Monitorování, analýza a detekce útoků v sítích

2.1 Systémy monitorování a analýzy dat využité v této práci

V této práci je pozorován a analyzován síťový provoz pomocí monitorovacích sond na hraničních linkách sítě CESNET2, z kterých jsou napozorovaná data exportována kolektor, kde je prováděna analýza.

Monitorovací sondy jsou zařízení, která sbírají data síťového provozu, tyto data agregují do záznamů o tocích a exportují. Monitorovací sondy se zhlediska monitorovaného provozu chovají jako pasivní prvky, monitorovaný provoz tedy neovlivňují. Agregace dat do záznamů o tocích výrazně snižuje velikost exportovaných dat. Zpravidla se k agregaci a exportu používají formáty NetFlow [10] a IPFIX [11], které ukládají informace jednoznačně definující toky. V síti CESNET2 jsou záznamy o tocích z monitorovacích sond odesílány do kolektoru, kde pomocí systému Nemea probíhá analýza.

2.2 Systém Nemea

Tato sekce čerpá z [12].

Systém Nemea je framework, jehož cílem je umožnit real-time analýzu dat nasbíraných monitorovacími systémy. Systém Nemea obsahuje moduly, které jsou vzájemně oddělené, nicméně mezi sebou mohou přenášet data. Jeden modul je zpravidla jedním systémovým procesem, který čte data ze vstupních rozhraních, kterými do modulu proudí informace o tocích z monitorovacích sond nebo data z jiných Nemea modulů. Dále data modul zpracovává a zaznamenává výsledky, což může dělat pomocí výstupního rozhraní do jiného Nemea modulu. Mezi existující moduly patří například: moduly pro příjem

dat, moduly pro předzpracovávání dat, moduly pro detekci různých anomálií, moduly pro zpracovávání výsledků a moduly pro logování dat nebo reportování detekovaných událostí.

Systém Nemea klade důraz na vysokou modularitu a flexibilitu, vysokou propustnost, real-time zpracování dat a jednoduchou implementaci modulů.

Komunikační vrstva systému Nemea je postavena na TRaffic Analysis Platform (TRAP) [12]. TRAP je reprezentován sdíleným objektem libtrap, což je dynamická knihovna, která je zlinkována s každým Nemea modulem. TRAP nabízí několik typů rozhraní ke komunikaci - TCP/IP, unixové sokety nebo sdílenou paměť.

Data jsou do modulů a mezi moduly přenášena ve formátu Unified Record (UniRec) [12]. UniRec je formát zpráv, které jsou v Nemea systému přenášeny pomocí TRAP knihovny. Moduly si musí vyměňovat více typů dat a množství těchto dat může být velké. Kvůli tomu je nutné používat struktury, které jsou paměťově a časově efektivní. Formát UniRec tedy klade důraz na flexibilní formát zpráv (lze přenášet i položky proměnných délek), paměťovou efektivitu a časovou efektivitu.

2.3 Metody detekce útoků

Obecným postupem pro detekci síťových útoků je hledat v datech nějaké vlastnosti, které by mohly indikovat útok. Jednou z možností je vhodně zvolit statistiky, o kterých se domníváme, že na základě jejich hodnot bude možné rozpoznat útoky, které chceme detekovat. V naměřených hodnotách statistik budeme dále hledat anomálie. Pokud víme, jaký by měly mít hodnoty pozorovaných statistik průběh, zvolíme dále způsob detekce odchylek od očekávaného průběhu. V případě, že o očekávaném průběhu nemáme příliš jasnou představu, se musíme nejdříve na základě napozorovaných dat rozhodnout jak definovat očekávaný provoz.

Detekci odchylek napozorovaných hodnot statistik od očekávaných hodnot můžeme provádět mnoha způsoby. Lze použít například: střední hodnoty, rozptyly, diskrétní derivace nebo korelace více statistik. Také můžeme použít složitější algoritmy, mezi které patří například Change Point Detection (CPD) algoritmy.

2.4 Change Point Detection pro detekci síťových útoků

Tato sekce čerpá z [13].

Lze očekávat, že některé relevantní statistiky budou mít před útokem a po útoku jiné parametry statistického rozdělení nebo samotný typ statistického

rozdělení jiný. Díky tomu lze použít detekci pomocí metod Change Point Detection (CPD). CPD je způsob detekce, který detekuje změny ve statistickém rozdělení pozorovaných dat. Jedním z hlavních typů detekcí změn rozdělení je sekvenční CPD. Sekvenční CPD se snaží co nejrychleji detekovat změny v rozdělení při zachování specifikovaného poměru falešných pozitiv (false alarm rate - FAR), což jsou případy, ve kterých algoritmus detekuje změnu rozdělení, ačkoliv k ní nedošlo. Při navrhování nejrychlejších sekvenčních CPD metod musíme vhodně zvolit mezi nízkým zpožděním detekce změny od chvíle, kdy ke změně došlo, a nízkým počtem falešných pozitiv.

Mějme posloupnost $\{X_n, n \geq 1\}$ hodnot statistiky, kterou jsme zvolili pro detekci síťových útoků pomocí sekvenčního CPD. Napozorované náhodné proměnné mají funkci hustoty pravděpodobnosti $p_0(X_1, ..., X_n)$ do doby, než dojde ke změně rozdělení na nějakém indexu λ, $\lambda \in 1, 2, ...$, kdy se funkce hustoty pravděpodobnosti změní na $p_1(X_1, ..., X_n)$. Jinými slovy lze předpokládat, že $X_1, X_2, ...$ mají podmíněnou funkci hustoty pravděpodobnosti $p_0(X_n|X_1, ..., X_{n-1})$ pro $n < \lambda$ a $p_1(X_n|X_1, ..., X_{n-1})$ pro $n \geq \lambda$.

Čas, kdy použitý sekvenční CPD algoritmus detekuje změnu, označme indexem τ. Dále definujme, že P_k a E_k označují funkci hustoty pravděpodobnosti a funkci očekávané střední hodnoty pozorované statistiky poté, co dojde ke změně rozdělení na indexu k (tj. $\lambda = k$). Index P_0 a E_0 definujme jako funkci hustoty pravděpodobnosti a funkci očekávané střední hodnoty do první změny rozdělení.

Pro vyhodnocení výkonnosti sekvenční metody nás bude zajímat hlavně průměrné zpoždění detekce změny rozdělení (average detection delay - ADD) od chvíle, kdy ke změně došlo. A poměr falešných pozitiv (FAR).Obecné matematické vzorce pro vyhodnocení ADD a FAR jsou uvedeny v rovnicích 2.1 a 2.2.

$$\mathrm{ADD}_\lambda(\tau) = E_\lambda(\tau - \lambda | \tau \geq \lambda) \tag{2.1}$$

$$\mathrm{FAR}(\tau) = \frac{1}{E_0 \tau} \tag{2.2}$$

2.4.1 CUSUM metody

Jedním z typů sekvenčních CPD metod jsou Cumulative Sum (CUSUM) metody. Základní CUSUM metodou je Pagův CUSUM, který je založen na matematickém vzorci 2.3 a dále počítá maximum podle vzorce 2.4. Hledání, kdy došlo ke změně rozdělení, pak probíhá podle vzorce 2.5, kde h je mezní hodnota.

$$Z_{n,\lambda} = \sum_{k=\lambda}^{n} \log \frac{p_1(X_k|X_1, ..., X_{k-1})}{p_0(X_k|X_1, ..., X_{k-1})}, n \geq \lambda \tag{2.3}$$

$$U_n = \max_{1 \leq \lambda \leq n} (Z_{n,\lambda}), \tag{2.4}$$

$$\tau(h) = \min n \geq 1 : U_n \geq h \tag{2.5}$$

Za předpokladů, že $h > 0$ a že pozorované hodnoty statistiky X jsou nezávislé a mají identické rozdělení, tedy že $p_0(X_n|X_1, ..., X_{n-1}) = p_0(X_n)$ a $p_1(X_n|X_1, ..., X_{n-1}) = p_1(X_n)$, pak lze U_n ve vzorci 2.5 nahradit statistikou $\widetilde{U}_n$ definovanou v 2.6, které je základem pro neparametrické CUSUM metody (NP-CUSUM).

$$\widetilde{U}_n = \max(0, \widetilde{U}_{n-1} + \log \frac{p_1(X_n)}{p_0(X_n)}), \widetilde{U}_0 = 0 \tag{2.6}$$

2.4.2 NP-CUSUM

Tato podsekce čerpá z [14].

Neparametrické CUSUM (NP-CUSUM) metody jsou typem CUSUM metod, které mohou být použity k detekci změn rozdělení statistik, které jsou nezávislé a mají identické, ale neznámé rozdělení. Nejprve došlo ke zobecnění Pagova CUSUM vzorce na vzorec 2.7 s nějakou funkcí f. Změny středních hodnot mohou být detekovány vzorcem 2.8, kde $\widehat{\mu}$ je odhad střední hodnoty X_n před změnou rozdělení, $\widehat{\theta}$ je odhadem střední hodnoty po změně rozdělení a ϵ je optimalizačním parametrem. Pro optimalizace pomocí parametru ϵ se používají dva přístupy. Prvním je minimalizovat ADD tak, aby FAR nepřekročil stanovenou mezní hodnotu. Druhým přístupem je minimalizovat FAR tak, aby ADD nepřekročil stanovenou mezní hodnotu. V [14] byla tato metoda použita k detekci SYN-flood útoků.

$$S_n = \max(0, S_{n-1} + f(X_n)) \tag{2.7}$$

$$S_n = \max(0, S_{n-1} + X_n - \widehat{\mu} - \epsilon\widehat{\theta}) \tag{2.8}$$

2.4.3 EWMA

Tato podsekce čerpá z [15].

Exponentially weighted moving average (EWMA) je metoda výpočtu klouzavého průměru, kterou lze použít k detekci anomálií. Metoda EWMA nepatří do kategorie CPD algoritmů, jelikož spíše než, že by detekovala změny rozdělení, pozoruje průměr za poslední období. Základní vzorec pro EWMA je definován v 2.9, kde α je koeficient pro vyhlazování, který určuje, jak moc má být při výpočtu brána v potaz historie napozorovaných hodnot. Podle [16] se

ukázalo, že EWMA je efektivnější (nižší ADD) pro pomalé změny rozdělení a CUSUM pro rychlé změny rozdělení.

$$z_i = \alpha x_i + (1 - \alpha) z_{i-1}, z_0 = x_0 \tag{2.9}$$

Kapitola 3

Návrh implementace modulu

V této kapitole se budu zabývat návrhem implementace modulu do systému Nemea pro detekci skenovacích útoků a útoků typu DoS přes protokol SIP. Modul bude:

- Přijímat data z kolektoru a z nich filtrovat SIP provoz.
- Počítat statistiky v časových okénkách.
- Počítat statistiky z hodnot naměřených v časových okénkách.
- Detekovat a zaznamenávat anomálie do souboru.

Pro analýzu SIP dat, která na kolektor přijdou, jsem se rozhodl nejprve vytvořit několik dalších programů (podmodulů), pomocí kterých půjde logovat SIP data do souboru a na těchto naměřených datech provádět offline analýzy podle vybraných algoritmů a specifikovaných parametrů. Konkrétně jsem navrhl 4 podmoduly:

1. První podmodul bude umět číst data, která přijdou na kolektor, přes TRAP rozhraní. Z nich vybere záznamy o SIP tocích a o každém SIP toku zaznamená několik vybraných informací do CSV souboru na disk. Informace, které tento podmodul ukládá, jsou: zdrojová IP adresa toku, cílová IP adresa toku, čas vzniku toku a čas ukončení toku. Podmodul jsem pojmenoval sip-detector-logger.
2. Druhý podmodul bude číst data naměřená předchozím podmodulem z CSV souboru, z naměřených dat bude počítat statistiky v časových okénkách a výsledky bude ukládat do dalšího CSV souboru. Podmodul jsem pojmenoval sip-detector-tw-stats (tw je zkratkou pro time window).
3. Třetí podmodul bude číst data z CSV souboru vytvořeným předchozím podmodulem, z kterých bude počítat další statistiky. Výsledky bude

ukládat do dalšího CSV souboru. Podmodul jsem pojmenoval sip-detector-cpd.

4. Čtvrtý podmodul bude číst výstupní data prvního, druhého nebo třetího podmodulu z CSV souboru, dále zanonymizuje IP adresy - tedy každé IP adrese přiřadí nějaký jednoznačný identifikátor - a vytvoří kopii vstupního souboru, kde místo IP adres budou uvedeny identifikátory. Navíc ještě někde zaznamená mapování IP adres na identifikátory.

K analýze bude kvůli očekávanému velkému množství dat nutné vytvořit ještě několik skriptů, které budou: vybírat jenom několik IP adres, u kterých je maximum hodnot statistik nejvyšší, a generovat grafy.

3.1 Statistiky pro časová okénka

U obětí DoS útoků můžeme očekávat, že na atakované IP adresy bude navázán velký počet toků, za krátký čas. U DDoS útoků lze navíc očekávat, že počet IP adres, které danou IP adresu kontaktují, bude v krátkém čase vysoký. DoS útočníci budou odesílat velké množství zpráv za krátký čas. U skenovacích útoků lze očekávat, že útočník za krátký čas vytvoří hodně toků. Pokud se jedná o skenování, kde útočník hledá SIP zařízení, bude počet kontaktovaných IP adres za krátký čas vysoký. Z těchto důvodů jsem se rozhodl pozorovat v časových okénkách tyto statistiky o IP adresách:

- Počet toků, které IP adresa v časovém okénku iniciovala s účelem detekovat útočící IP adresy skenovacích a DoS útoků.
- Počet IP adres, které IP adresa v časovém okénku kontaktovala s účelem detekovat skenovací útočící IP adresy.
- Počet toků, kterých se IP adresa v časovém okénku účastnila jako cíl s účelem detekovat oběti DoS útoků.
- Počet IP adres, které s IP adresou v časovém okénku navázali tok s účelem detekovat oběti DDoS útoků.

3.2 Statistiky pro detekci anomálií

K detekci anomálií jsem zvolil metodu EWMA, která by měla umět dobře detekovat útoky, které probíhají delší dobu a mají vysokou intenzitu. Metodu NP-CUSUM jsem zatím neimplementoval, jelikož se mi nepodařilo nalézt způsob pro uspokojivé odhadnutí parametrů $\widehat{\mu}$ a $\widehat{\theta}$ vzorce uvedeného v podsekci 2.4.2.

3.3 Datové struktury

3.3.1 Záznamy o IP adresách

Skriptem jsem z dat nasbíraných za 10 dnů zjistil tyto poznatky (týká se pouze protokolu SIP):

- Maximální počet zdrojových IP adres, které se objeví v jednom časovém okénku, je 1 098.
- Maximální počet cílových IP adres, které se objeví v jednom časovém okénku, je 794 197.
- Počet zdrojových IP adres napozorovaných za 10 dnů je 104 595.
- Počet cílových IP adres napozorovaných za 10 dnů je 3 019 475.

Vzhledem k nárokům kladeným na rychlost a kapacitu přichází v úvahu několik datových struktur, například:

- **B+ strom**

  B+ strom má rychlé operace vkládání ($\mathcal{O}(\log n)$) a vyhledávání ($\mathcal{O}(\log n)$). Paměťová složitost je $\mathcal{O}(n)$.

- **Hašovací tabulka**

  Hašovací tabulka bude vyžadovat alokaci dostatečně velkého pamětového místa. Nicméně operace vkládání a vyhledávání budou mít při specifikaci maximálního počtu konfliktů na konstantní hodnotu asymptotickou časovou složitost $\mathcal{O}(1)$. Nevýhodou je, že pokud bude tabulka příliš zaplněná, pak bude vkládání nových záznamů často způsobovat překročení maximálního počtu konfliktů, což způsobí odebrání nějakého záznamu z tabulky. Konflikty lze omezit alokováním prostoru pro co nejvíce záznamů hašovací tabulky.

Zvolil jsem hašovací tabulku, konrétně kukaččí hašovací tabulku, jelikož umožňuje efektivně využívat alokovanou paměť pomocí více než jedné hašovací funkce [9].

3.3.2 Záznamy o detekovaných anomáliích

Aby modul uměl detekovat, kdy došlo k začátku anomálního stavu a kdy anomální stav skončil, je potřeba ukládat data o anomáliích. I v tomto případě jsem zvolil kukaččí hašovací tabulku kvůli své rychlosti a efektivnímu využití alokované paměťi.

Kapitola 4

Realizace

K realizaci modulu jsem použil programovací jazyk C z důvodů, že je systémem Nemea podporován a lze jím dosáhnout časově i paměťově efektivnějších řešení než při použití jiných vyšších programovacích jazyků. Systém Nemea s modulem sip-detector jsem sestavoval pomocí balíku Autotools - Autoconf [17], Automake [18], Libtool [19]. Pro verzování kódu jsem použil systém GIT [20].

4.1 Způsob specifikování parametrů programů

U všech čtyř podmodulů i hlavního programu sip-detector jsem zvolil způsob specifikování parametrů přes XML soubor. Parametry, které jsou předávány přes argumenty programů, jsou pouze cesta ke XML konfiguračnímu souboru a případně parametry TRAP rozhraní (z důvodů, že pro inicializaci TRAP rozhraní podle argumentů programu již existují makra v libtrap). Pro specifikaci a validaci XML souborů jsem navíc vytvořil XSD schéma.

Ke zpracování konfiguračních XML souborů jsem použil knihovnu libxml [21] a pro čtení jednotlivých konfiguračních položek jsem použil dotazovací jazyk XPath [22] také z knihovny libxml.

4.2 sip-detector-logger

Program nejprve přečte argumenty programu, což jsou zde: specifikace TRAP rozhraní, z kterého bude program číst data, a cesta ke konfiguračnímu XML souboru. Dále program provede inicializaci TRAP rozhraní a z konfiguračního XML souboru přečte cestu k souboru, do kterého bude ukládán výstup. Program následně zapíše hlavičku do výstupního souboru a začne číst a zpracovávat záznamy o tocích z TRAP rozhraní. Pokud se přečtený záznam o toku netýká SIP, je tento záznam o toku ignorován. V opačném případě je do výstupního souboru zaznamenána zdrojová IP adresa, cílová IP adresa, čas vzniku toku a čas ukončení toku. Při obdržení signálu SIGINT nebo SIG-

TERM dojde k uvolnění rezervovaných prostředků a program končí. Pro lepší představu je struktura programu popsána ještě v pseudokódu 1.

Pseudokód 1 sip-detector-logger

```
1: Inicializuj TRAP rozhraní podle argumentů programu;
2: Přečti cestu ke konfiguračnímu XML souboru z argumentů programu;
3: Načti konfiguraci z XML souboru;
4: Zapiš hlavičku výstupního CSV souboru;
5: while Program neobdržel SIGINT ani SIGTERM do
6:    Přečti data z TRAP rozhraní;
7:    Převeď IP adresy a časovou značku začátku toku na řetězce;
8:    Ulož záznam na konec výstupního souboru;
9: end while
10: Uvolni prostředky;
```

4.3 sip-detector-tw-stats

Program nejprve přečte cestu ke konfiguračnímu XML souboru z argumentů programu. Dále načte z XML souboru vstupní parametry, což jsou:

- Perioda časového okénka.
- Maximální počet zdrojových (resp. cílových) IP adres, které program dokáže k jedné IP adrese v jednom časovém okénku napozorovat.
- Velikost hašovací tabulky pro záznamy o zdrojových (resp. cílových) IP adresách.
- Cesta ke vstupnímu CSV souboru vytvořeném programem sip-detector-logger.
- Cesta k souboru, do kterého bude program ukládat statistiky o zdrojových (resp. cílových) IP adresách.

Program dále provede potřebné inicializace a dále čte řádek po řádku vstupní soubor. Nejprve jsou z řádků zparsovány IP adresy a časová značka začátku toku. Pokud časová značka patří do nějakého dalšího časového okénka, jsou naměřené statistiky o IP adresách z hašovacích tabulek uloženy do výstupních souborů a je nastavena časová značka nového časového okénka. Dále je v hašovací tabulce nalezen záznam o zdrojové (resp. cílové) IP adrese, případně je vytvořen nový záznam o dané IP adrese, pokud zatím v aktuálním časovém okénku neexistuje. Následně dojde k aktualizaci záznamů o IP adresách. Po přečtení posledního řádku vstupního souboru program končí. Pro lepší představu jsem program popsal ještě v pseudokódu 2.

Pseudokód 2 sip-detector-tw-stats

```
1: Přečti cestu ke konfiguračnímu XML souboru z argumentů programu;
2: Načti konfiguraci z XML souboru;
3: Inicializuj hašovací tabulky pro záznamy o zdrojových (resp. cílových) IP
   adresách;
4: Zapiš hlavičky výstupních CSV souborů;
5: Přečti a zkontroluj hlavičku vstupního CSV souboru;
6: while Nejsi na konci vstupního souboru do
7:    Přečti další řádek vstupního CSV souboru;
8:    Z řádku zparsuj IP adresy a časovou značku;
9:    if Časová značka patří do dalšího čas. okénka then
10:      Ulož všechny validní záznamy o IP adresách z hašovacích tabulek do
   výstupních CSV souborů;
11:      Označ všechny záznamy hašovacích tabulek jako nevalidní;
12:      Nastav časovou značku nového čas. okénka;
13:   end if
14:   Najdi v hašovacích tabulkách záznamy o IP adresách;
15:   if Záznam o zdrojové (resp. cílové) IP adrese nenalezen then
16:      Vlož nový záznam o zdrojové (resp. cílové) IP adrese do příslušné
   hašovací tabulky;
17:   end if
18:   Aktualizuj hodnoty statistik daného záznamu o IP adrese;
19: end while
20: Uvolni prostředky;
```

4.4 sip-detector-cpd

Program nejprve přečte cestu ke konfiguračnímu XML souboru z argumentů programu. Dále načte z XML souboru definice množin statistik. Každá množina statistik má tyto parametry:

- Definice několika statistik, které má program počítat. U každé statistiky je specifikován: název, název vstupní statistiky, kterou k výpočtu použít, název metody (zatím je podporována jen metoda EWMA), a specifické parametry (u EWMA parametr α).
- Velikost hašovací tabulky.
- Typ hašovací tabulky - zda jsou v hašovací tabulce pozorovány statistiky o zdrojových IP adresách nebo o cílových IP adresách.
- Perioda pro vyčištění hašovací tabulky.
- Cesta ke vstupnímu CSV souboru vytvořeném programem sip-detector-tw-stats, který by měl korespondovat s typem hašovací tabulky.

- Cesta k výstupnímu CSV souboru.
- Perioda časového okénka.

Dále program zpracuje postupně všechny množiny statistik (zpravidla bude jedna pro statistiky o zdrojových IP adresách a jedna pro statistiky o cílových IP adresách). Při zpracovávání množiny statistik dojde nejprve k potřebným inicializacím a dále jsou ze vstupního souboru čteny naměřené hodnoty v časových okénkách, ze kterých jsou následně počítány další statistiky na základě definice dané množiny statisitk a výstupy jsou ukládány do výstupního souboru dané množiny statistik. Struktura programu je lépe popsána v pseudokódu 3.

4.5 sip-detector-anonymizer

Program opět nejprve přečte cestu ke konfiguračnímu XML souboru z argumentů programu. Dále načte z XML souboru vstupní parametry, což jsou:

- Maximální počet záznamů o mapování IP adres.
- Cesta k vstupnímu CSV souboru s mapováním IP adres.
- Cesta k výstupnímu CSV souboru, do kterého program uloží aktualizované mapování o IP adresách.
- Cesta k vstupnímu CSV souboru, který má být anonymizován.
- Cesta k výstupnímu CSV souboru, do kterého bude uložen zanonymizovaný výstup.

Dále program provede potřebné inicializace, načte vstupní soubor s mapováním a následně začne číst řádek po řádku vstupní soubor, který má být zanonymizován. U každého řádku program nejprve vyhledá záznam o mapování k přečtené IP adrese, případně vytvoří nový záznam, pokud dosud neexistuje, a následně podle identifikátoru v záznamu o mapování zapíše zanonymizovanou kopii přečteného řádku do souboru, kam má být uložen výstup. Přesnější popis programu je uveden v pseudokódu 4.

4.6 sip-detector

Hlavní program sip-detector bude dělat víceméně to, co dělají programy sip-detector-logger, sip-detector-tw-stats a sip-detector-cpd, s odlišností, že do souborů budou ukládány už jenom detekované anomálie při vysokých hodnotách (podle specifikovaných mezních hodnot) vybraných statistik. Navíc je zde

Pseudokód 3 sip-detector-cpd

```
1: Přečti cestu ke konfiguračnímu XML souboru z argumentů programu;
2: Načti konfiguraci z XML souboru;
3: for Množiny statistik do
4:   Inicializuj množinu statistik;
5:   Zapiš hlavičku výstupního CSV souboru;
6:   Přečti a zkontroluj hlavičku vstupního CSV souboru;
7:   while Nejsi na konci vstupního souboru do
8:     Přečti další řádek vstupního souboru;
9:     Z řádku zparsuj IP adresu, časovou značku a hodnoty statistik;
10:    if Jsi na začátku then
11:      Nastav časovou značku posledního vyčištění hašovací tabulky na
       časovou značku prvního přečteného záznamu;
12:    end if
13:    if Součet časové značky posledního vyčištění hašovací tabulky a pe-
       riody pro čištění hašovací tabulky je nižší než časová značka přečteného
       záznamu then
14:      Vyčisti hašovací tabulku;
15:      Nastav časovou značku posledního vyčištění hašovací tabulky na
         časovou značku přečteného záznamu;
16:    end if
17:    Najdi v hašovací tabulce záznam o IP adrese;
18:    if Záznam o IP adrese nenalezen then
19:      Vlož nový záznam o IP adrese do hašovací tabulky;
20:    end if
21:    while Časová značka poslední aktualizace záznamu o IP adrese je
       menší než rozdíl aktuální časové značky a velikosti časového okénka do
22:      Přičti k časové značce poslední aktualizace záznamu o IP adrese
         velikost časového okénka;
23:      Spočítej hodnoty EWMA statistik přidáním nulových hodnot s ča-
         sovou značkou poslední aktualizace záznamu o IP adrese;
24:    end while
25:    Spočítej hodnoty EWMA statistik přidáním aktuální hodnoty statis-
       tiky s aktuální časovou značkou;
26:    Ulož spočítané hodnoty do výstupního CSV souboru;
27:  end while
28:  Uvolni prostředky množiny statistik.
29: end for
```

Pseudokód 4 sip-detector-anonymizer

```
1: Přečti cestu ke konfiguračnímu XML souboru z argumentů programu;
2: Načti konfiguraci z XML souboru;
3: Inicializuj pole pro ukládání záznamů o mapování;
4: Přečti a zparsuj vstupní CSV soubor s mapováním;
5: Přečti a zkontroluj hlavičku vstupního CSV souboru s daty;
6: Zapiš hlavičku výstupního CSV souboru s daty, která bude stejná jako
   u vstupního CSV souboru s daty;
7: while Nejsi na konci vstupního souboru do
8:    Přečti další řádek vstupního souboru;
9:    Z řádku zparsuj IP adresu;
10:   Najdi záznam o mapování;
11:   if Záznam o mapování nenalezen then
12:      Vlož nový záznam o mapování do seřazeného pole se záznamy o ma-
   pování;
13:   end if
14:   Zapiš do výstupního CSV souboru s daty přečtený řádek, který bude
   mít nahrazenou IP adresu identifikátorem;
15: end while
16: Ulož mapování do výstupního CSV souboru s mapováním;
17: Uvolni prostředky;
```

zpracovávání poplachů (tj. detekovaných anomálií), které probíhá tak, že pokud nějaká pozorovaná statistika překročí svou mezní hodnotu, tak je do souboru s poplachy zapsán záznam o začátku anomálie, ve kterém je uvedena IP adresa, název statistiky a časová značka. Poté, co se hodnota statistiky, která byla detekována, dostane opět pod svou mezní hodnotu, je do souboru s poplachy zapsán záznam o skončení anomálie. Struktura programu je popsána v pseudokódu 5 a vstupní parametry konfiguračního XML souboru jsou:

- Perioda časového okénka.
- Maximální počet zdrojových (resp. cílových) IP adres, které program dokáže k jedné IP adrese v jednom časovém okénku napozorovat.
- Velikost hašovací tabulky pro záznamy o zdrojových (resp. cílových) IP adresách. V pseudokódu 5 jsou tyto hašovací tabulky označovány jako *tw-stats* hašovací tabulky.
- Definice množin statistik podobně jako u programu sip-detector-cpd, nejsou zde ovšem cesty k souborům a navíc je u každé statistiky specifikovaná mezní hodnota pro detekce anomálií.

Pseudokód 5 sip-detector

```
1: Inicializuj TRAP rozhraní podle argumentů programu;
2: Přečti cestu ke konfiguračnímu XML souboru z argumentů programu;
3: Načti konfiguraci z XML souboru;
4: Inicializuj tw-stats hašovací tabulky (pro záznamy o zdrojových a cílových
   IP adresách);
5: Inicializuj množiny statistik definované konfiguračním souborem;
6: Inicializuj hašovací tabulku pro poplachy;
7: Zapiš hlavičku CSV souboru, do kterého budou ukládány poplachy;
8: while Program neobdržel SIGINT ani SIGTERM do
9:    Přečti data z TRAP rozhraní;
10:   Přečti IP adresy a časovou značku začátku toku;
11:   if Jsi na začátku then
12:      Nastav časovou značku posledního vyčištění hašovacích tabulek mno-
      žin statistik na časovou značku prvního přečteného záznamu;
13:   end if
14:   for Množiny statistik do
15:      if Součet časové značky posledního vyčištění hašovací tabulky dané
      množiny statistik a periody pro čištění hašovací tabulky dané množiny
      statistik je nižší než časová značka přečteného záznamu then
16:         Vyčisti hašovací tabulku dané množiny statistik;
17:         Nastav časovou značku posledního vyčištění hašovací tabulky dané
      množiny statistik na časovou značku přečteného záznamu;
18:      end if
19:   end for
```

```
20:   while Časová značka patří do některého dalšího časového okénka do
21:     for Všechny validní záznamy o IP adresách v tw-stats hašovacích tabulkách do
22:       for Množiny statistik do
23:         Najdi v hašovací tabulce dané množiny statistik záznam o IP adrese;
24:         if Záznam o IP adrese nenalezen then
25:           Vlož nový záznam o IP adrese;
26:         end if
27:         while Časová značka poslední aktualizace záznamu o IP adrese je menší než rozdíl aktuální časové značky a velikosti časového okénka do
28:           Přičti k časové značce poslední aktualizace záznamu o IP adrese velikost časového okénka;
29:           Spočítej hodnoty EWMA statistik přidáním nulových hodnot s časovou značkou poslední aktualizace záznamu o IP adrese;
30:         end while
31:         Spočítej hodnoty EWMA definovaných statistik přidáním aktuální hodnoty statistiky s aktuální časovou značkou;
32:         Zapiš případné začátky nebo konce poplachů do souboru s poplachy.
33:       end for
34:     end for
35:     Označ všechny záznamy tw-stats hašovacích tabulkek jako neplatné;
36:     Nastav časovou značku nového čas. okénka;
37:   end while
38:   Najdi v tw-stats hašovacích tabulkách záznamy o zdrojové (resp. cílové) IP adrese;
39:   if Záznam o IP adrese nenalezen then
40:     Vlož nový záznam o IP adrese;
41:   end if
42:   Aktualizuj hodnoty statistik záznamu o IP adrese v časových okénkáh;
43: end while
44: Uvolni prostředky;
```

Kapitola 5

Testování a analýza reálných dat

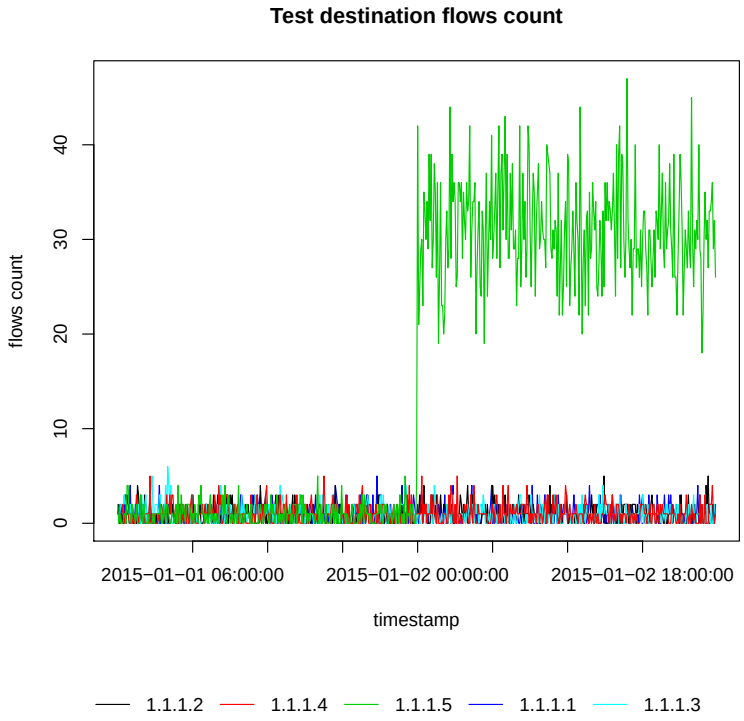
5.1 Testování programu sip-detector-logger

Kvůli očekávání, že data odesílaná na vstupní TRAP rozhraní programu jsou korektní, a kvůli jednoduchosti tohoto programu jsem tento program testoval pouze pozorováním naměřených dat.

5.2 Testování programu sip-detector-tw-stats

Pro testování tohoto podmodulu jsem nejprve vytvořil další program pro vygenerování testovacích dat, která budou zastupovat výstupní testovací data podmodulu sip-detector-logger. Programem jsem vygeneroval data pro 2 dny a to pouze pro pět zdrojových IP adres (zvolil jsem IP adresy 1.1.1.[1-5]) a pět cílových IP adres (zvolil jsem IP adresy 2.2.2.[1-5]). Pro vygenerování provozu jsem použil Poissonův proces. Intenzity jsem nastavil tak, že každá zdrojová IP adresa (1.1.1.[1-5]) vytvoří průměrně jeden záznam o toku za pět minut s libovolnou vygenerovanou cílovou IP adresu a každá cílová IP adresa (2.2.2.[1-5]) vytvoří průměrně jeden záznam o toku za pět minut s libovolnou vygenerovanou zdrojovou IP adresou. Dále o půlnoci dalšího dne IP adresa 1.1.1.5 a IP adresa 2.2.2.5 zvýší intenzitu generování o třicet požadavků za pět minut, čímž je simulován skenovací a DDoS útok. Očekáváme tedy, že ve výstupu programu sip-detector-tw-stats budou mít neútočící IP adresy střední hodnotu počtů záznamů o tocích a střední hodnotu pozorovaných ip adres rovné jedné a totéž bude platit i o adresách 1.1.1.5 a 2.2.2.5 do druhého dne. O půlnoci druhého dne by potom měly střední hodnoty pro IP adresy 1.1.1.5 a 2.2.2.5 konvergovat k 31.

Dále je nad vygenerovanými daty spuštěn program sip-detector-tw-stats. Z jeho výstupů jsem vygeneroval grafy. V grafu 5.1 jsou zobrazeny počty toků zdrojových IP adres a zdá se, že vyhovují očekáváním (tj. u normálního provozu hodnota kolísá kolem hodnoty 1 a u útočného provozu kolem 31). Podobné očekávané průběhy mají i grafy ostatních pozorovaných statistik.



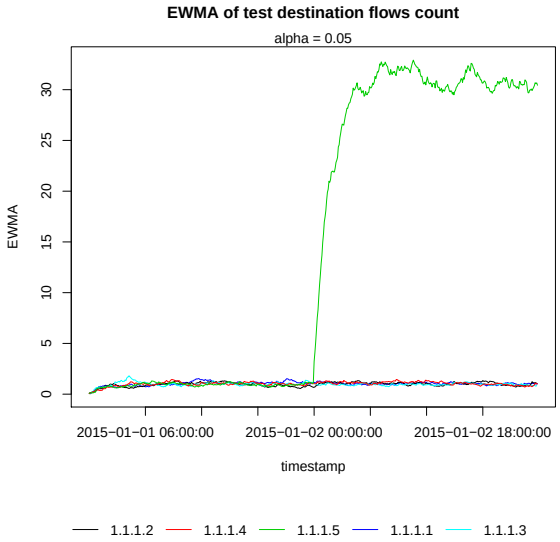
Obrázek 5.1: Počty toků zdrojových IP adres. Je zde vidět začátek útoku.

5.3 Testování programu sip-detector-cpd

Při testování tohoto podmodulu jsem postupoval tak, že jsem spustil podmodul sip-detector-cpd, kterému jsem jako vstupní data specifikoval výstupní data testování sip-detector-tw-stats a nastavil jsem výstupní statistiky pro EWMA. V grafu 5.2 je zobrazen průběh EWMA hodnot pro počty toků zdrojových IP adres. U útočící IP adresy se hodnoty podle očekávání pohybují kolem jedné a o půlnoci dalšího dne se hodnoty zvýší a celkově se blíží ke 31. U neútočného provozu se hodnoty této statistiky pohybují kolem jedné. I zde je tento průběh očekávaný. Podobně vypadají i grafy ostatních pozorovaných statistik.

5.4 Testování programu sip-detector-anonymizer

Tento program jsem testoval tak, že jsem porovnal, zda vstupní soubory s daty odpovídají výstupním souborům s daty, a pro několik IP adres jsem manuálně ověřil mapování.



Obrázek 5.2: EWMA počtů toků zdrojových IP adres.

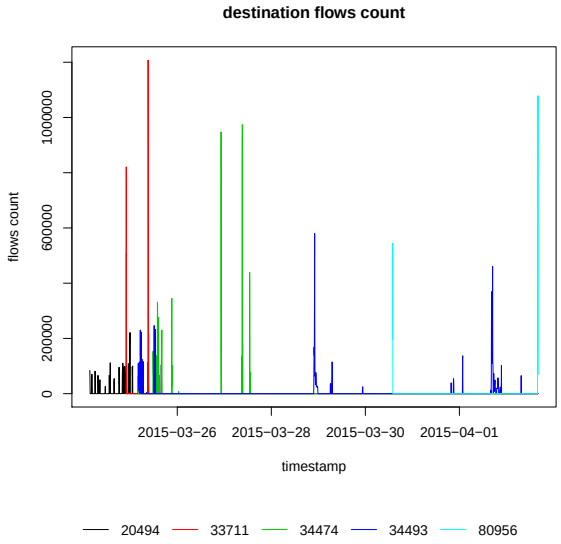
5.5 Testování programu sip-detector

Program sip-detector jsem testoval tak, že jsem nastavil hraniční hodnoty detekčních procedur na nízké hodnoty a program jsem na několik hodin spustil spolu s programem sip-detector-logger. Po několika hodinách jsem programy ukončil. Dále jsem se podíval na několik detekovaných anomálií a pro jejich IP adresy jsem dále provedl offline analýzu dat naměřených programem sip-detector-logger pomocí programů sip-detector-tw-stats, sip-detector-cpd a sip-detector-anonymizer a vygeneroval grafy. V grafech jsem potom ověřil, že detekované události odpovídají.

5.6 Analýza reálných dat

Analýzu reálných dat jsem prováděl na základě dat naměřených podmodulem sip-detector-logger za dva týdny. Ve všech grafech této sekce platí:

- Je vybráno pouze pět IP adres, kterých hodnoty pozorované statistiky dosáhly nejvyšších hodnot, z důvodu přehlednosti grafů.
- IP adresy jsou zanonymizovány a jejich identifikátory navzájem mezi grafy odpovídají.
- Průběhy, které jsou neukončené před koncem pozorovaného období, znamenají, že k dané IP adrese už nebyla do konce pozorovaného období napozorována žádná SIP zpráva.

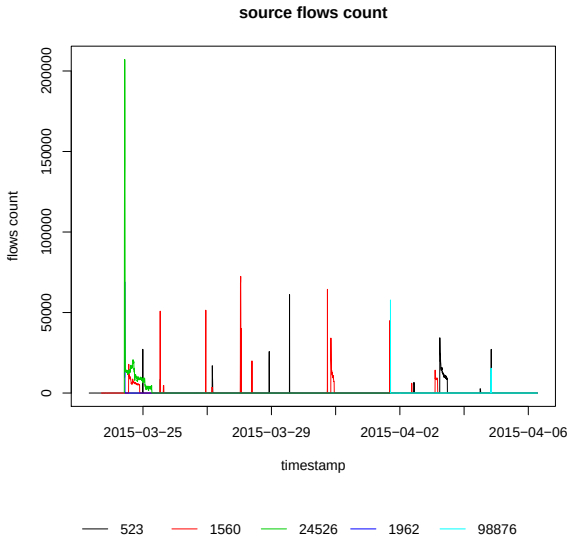


Obrázek 5.3: Počty toků, ve kterých byly dané IP adresy zdrojem.

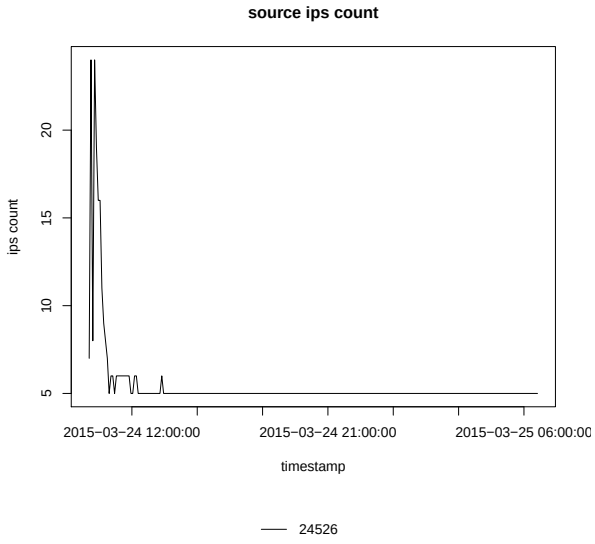
Nejprve jsem pozoroval hodnoty vybraných statistik bez použití EWMA pomocí podmodulu sip-detector-tw-stats. V grafu 5.3 jsou zobrazeny počty záznamů o SIP tocích v časových okénkách, kde byly dané IP adresy zdrojem. Počty toků jsou většinou blízké nule, ale občas nastane prudký nárůst, po kterém nastane brzy rychlý pokles zpět k nule. Tyto průběhy mohou znamenat jakési vlny skenovacích útoků, ale také je možné, že zde výkyvy znamenají útoky typu DoS. Všimněme si také, že obor hodnot, ve kterém se pohybujeme je <0,∼1 200 000>.

Graf 5.4 zobrazuje počty záznamů o SIP tocích, kde byly dané IP adresy cílem. Nejzajímavější je zde nejspíše průběh IP adresy 24526. První den pozorování na tuto IP adresu nepřijdou žádné SIP zprávy. Potom dojde k tomu, že je na danou IP adresu odeslán velký počet SIP zpráv. Krátce poté provoz prudce klesne a v průběhu následujícího dne klesne postupně až na nulu a v pozorovaném období už na tuto IP adresu žádná SIP zpráva nepřijde. V grafu 5.5 je zobrazen průběh počtů IP adres, které IP adresu 24526 kontaktovaly. IP adres není mnoho, nicméně se zde může jednat o útok DDoS, který možná úspěšně způsobil odmítnutí služby. Také se ale může jednat o skenovací útok, kde je účelem zjistit existující jména uživatelů a hesla. IP adresy 523, 1560 a 98876 mají méně prudké výkyvy, ale častější, což může znamenat, že jsou tyto IP adresy ve vlnách skenovány. Oborem hodnot je v grafu 5.4 <0,∼200 000>, maximum je tedy zhruba šestkrát menší, než je tomu u maxima počtů záznamů, kde dané IP adresy byly zdrojem (viz graf 5.3).

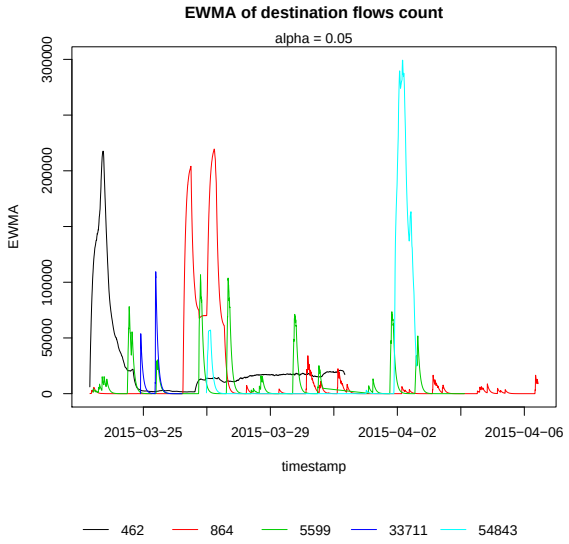
Dále jsem zkoušel na vybrané statistiky aplikovat metodu EWMA pomocí podmodulu sip-detector-cpd. Zkoušel jsem více hodnot parametrů α a nejza-



Obrázek 5.4: Počty toků, ve kterých byly dané IP adresy cílem.



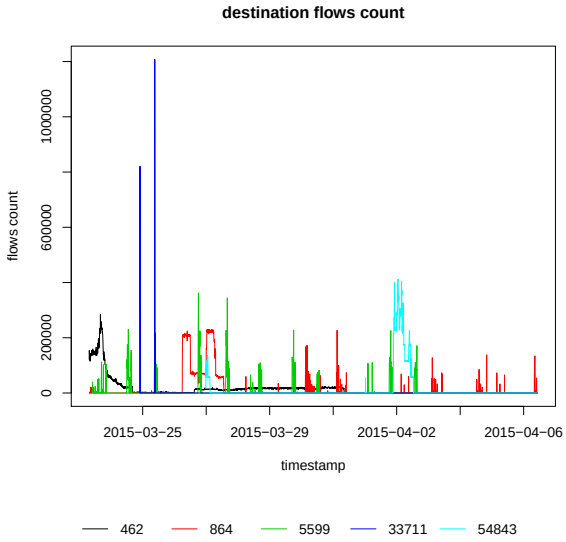
Obrázek 5.5: Počty IP adres, které IP adresu 24526 kontaktovaly.



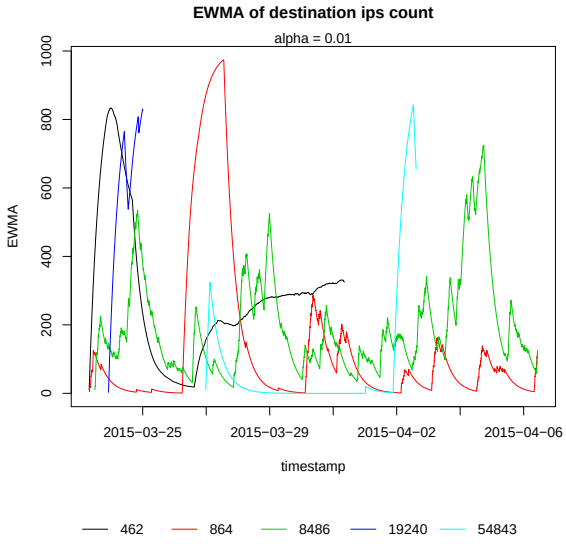
Obrázek 5.6: EWMA počtů toků, ve kterých byly dané IP adresy zdrojem.

jímavější průběhy jsem zobrazil v grafech. V grafu 5.6 jsou zobrazeny průběhy pro EWMA statistiky z počtů toků, kde jsou dané IP adresy zdrojem s parametrem α nastaveným na 0.05. Graf 5.7 potom zobrazuje průběhy počtů toků, kde jsou IP adresy z grafu 5.6 zdrojem. Při srovnání grafů 5.6 a 5.7 lze potvrdit, že EWMA statistika s parametrem α nastaveným na 0.05 bude detekovat hlavně útoky, které trvají delší dobu, hůře bude detekovat krátkodobé extrémy. V grafu 5.8 jsou zobrazeny průběhy statistiky EWMA z počtů IP adres, které daná IP adresa kontaktovala, s parametrem α nastaveným na 0.01. Zde jenom připomínám, že maximum napozorovaných cílových IP adres na jednu zdrojovou IP adresu je v jednom časovém okénku nastaveno na 1000. Při srovnání grafů 5.6 a 5.8 lze říci, že IP adresy, které vytvořily velký počet toků (kde byly zdrojem), také většinou kontaktovaly velký počet IP adres. Takové chování může znamenat skenování sítě s cílem najít SIP zařízení například pomocí OPTIONS SIP požadavků.

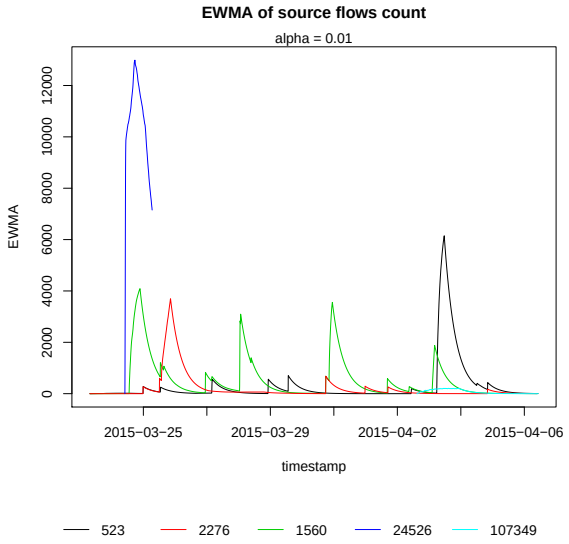
V grafu 5.9 jsou zobrazeny průběhy statistiky EWMA z počtů toků, kde jsou dané IP adresy cílem, s parametrem α nastaveným na 0.01. Graf 5.11 potom zobrazuje průběhy počtů toků, kde jsou IP adresy z grafu 5.9 cílem. Porovnáním těchto dvou grafů lze opět vidět, že EWMA detekuje hlavně dlouhodobější anomálie. V grafu 5.10 jsou zobrazeny průběhy statistiky EWMA z počtů IP adres, které daná IP adresa kontaktovala, s parametrem α nastaveným na 0.01. Zde jenom připomínám, že maximum napozorovaných cílových IP adres na jednu zdrojovou IP adresu je v jednom časovém okénku nastaveno na 100. Je zvláštní, že pouze IP adresa 107349 se nachází v obou grafech. Mohlo by to znamenat, že pokud se jedná o útoky typu DoS, tak největší



Obrázek 5.7: Počty toků, ve kterých byly dané IP adresy zdrojem.



Obrázek 5.8: EWMA počtů IP adres, které daná IP adresa kontaktovala. Maximum je nastaveno v jednom časovém okénku na 1000.



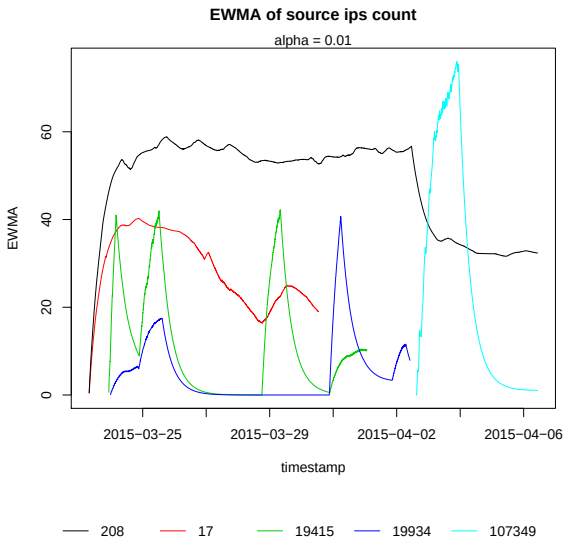
Obrázek 5.9: EWMA počtů toků, ve kterých byly dané IP adresy cílem.

napozorované útoky probíhají distribuovaně pouze z méně IP adres, ale kvůli příliš nízké zmíněné konstantě 100, kterou paměťové nároky nedovolují příliš zvýšit, lze těžko dělat přesnější závěry.

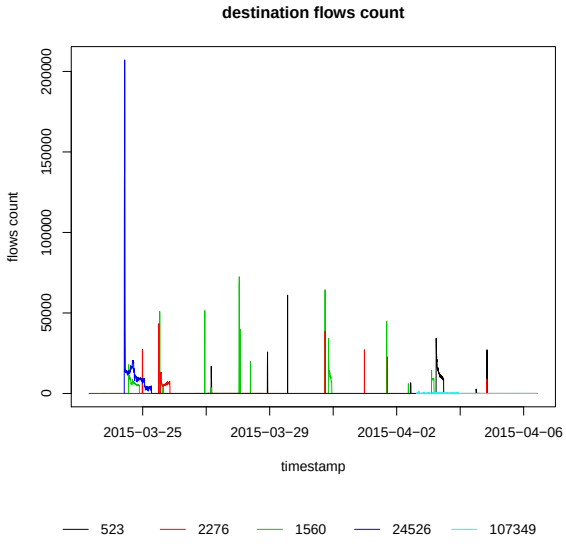
Na základě napozorovaných výsledků jsem nastavil hraniční hodnoty programu sip-detector a parametry statistik EWMA podle tabulky 5.1. Statistiku EWMA z počtů IP adres, které danou IP adresu kontaktovaly, jsem z paměťových důvodů v hlavním programu nepoužil.

Tabulka 5.1: Hraniční hodnoty EWMA statistik

Statistika	α	hraniční hodnota
EWMA počtů toků, ve kterých byly dané IP adresy zdrojem	0.05	150 000
EWMA počtů IP adres, které daná IP adresa kontaktovala	0.01	800
EWMA počtů toků, ve kterých byly dané IP adresy cílem	0.01	4 000



Obrázek 5.10: EWMA počtů IP adres, které danou IP adresou kontaktovaly. Maximum je nastaveno v jednom časovém okénku na 100.



Obrázek 5.11: Počty toků, ve kterých byly dané IP adresy cílem.

Závěr

V první části diplomové práce jsem se zabýval protokolem SIP. Zaměřil jsem se na popis útoků a jejich obranných mechanismů. Útoky, které jsem popsal, jsou: DoS útoky, skenovací útoky, útoky založené na modifikaci či podvrhování zpráv, autentizační útoky, SPIT útoky a také jsem zmínil útok založený na neoprávněném volání do sítě PSTN.

V druhé části jsem provedl analýzu a návrh metod pro detekci skenovacích útoků a útoků typu DoS přes protokol SIP. K detekci jsem použil metodu EWMA, kterou jsem aplikoval na statistiky týkající se základních informací o tocích měřené v časových okénkách. Pro vhodné nastavení parametrů detekčních procedur jsem naimplementoval programy, které umožňují offline analýzu, pomocí které jsem na základě vzorku dat napozorovaném za dva týdny porovnal více způsobů nastavení parametrů a vybral nejvhodnější konfiguraci. Zdrojové kódy programů pro offline analýzu jsem následně použil k implementaci hlavního programu pro real-time detekci skenovacích a DoS útoků přes SIP.

K testování modulu jsem vygeneroval provoz podle Poissonova procesu, ve kterém jsem simulovat skenovací útok a DoS útok. Vygenerované útoky jsem detekoval pomocí programů, podle kterých jsem dělal offline analýzu. Tím jsem otestoval offline analýzu. Dále jsem spustil hlavní program pro real-time detekci útoků a zároveň program pro logování SIP dat. Následně jsem provedl offline analýzu naměřených dat a ověřil, že výsledky odpovídají anomáliím detekovaným hlavním programem.

Vytvořený modul pro detekci skenovacích a DoS útoků funguje tak, že pozoruje SIP data zaznamenaná monitorovacími sondami sítě CESNET2 a pro každou napozorovanou IP adresu pozoruje informace o počtech toků, kde daná IP adresa byla zdroj (resp. cíl) a počtech IP adres, které daná IP adresa kontaktovala v časových okénkách. Z těchto napozorovaných statistik je dále spočítána statistika EWMA. Při překročení stanovených mezních hodnot je uložen záznam o detekci do souboru. Modul je zaměřený na rychlé zpracování velkého objemu dat z monitorovacích sond. Nastavení parametrů modulu

probíhá přes XML konfigurační soubor. K vyšší úspoře paměti jsou zatím podporovány jenom IP adresy verze 4, jelikož IP adres verze 6 bylo při analýze napozorováno jen velmi málo (méně než 0.1%).

Naimplementovaný modul by se dal rozšířit například přidáním dalších pozorovaných statistik nebo zavedením dalších metod (např. CPD metod) detekce útoků, na což jsem při implementaci kladl důraz. Vytvořené programy pro offline analýzu by dále měly usnadnit nastavování parametrů detekčních algoritmů. Také by bylo možné modul jednoduše zobecnit i na detekci útoků typu skenování a DoS útoků v dalších internetových protokolech, jelikož statistiky, které modul používá k detekci, lze pozorovat ve všech internetových protokolech.

Literatura

[1] Hoffstadt, D.; Marold, A.; Rathgeb, E.: Analysis of SIP-Based Threats Using a VoIP Honeynet System. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, June 2012, s. 541–548, doi:10.1109/TrustCom.2012.90.

[2] Mathieu, B.; Niccolini, S.; Sisalem, D.: SDRS: A Voice-over-IP Spam Detection and Reaction System. *Security Privacy, IEEE*, ročník 6, č. 6, Nov 2008: s. 52–59, ISSN 1540-7993, doi:10.1109/MSP.2008.149.

[3] Quittek, J.; Niccolini, S.; Tartarelli, S.; aj.: Detecting SPIT Calls by Checking Human Communication Patterns. In *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, s. 1979–1984, doi:10.1109/ICC.2007.329.

[4] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; aj.: SIP: Session Initiation Protocol. RFC 3261, Červen 2002. Dostupné z: `http://www.ietf.org/rfc/rfc3261.txt`

[5] Mokube, I.; Adams, M.: Honeypots: Concepts, Approaches, and Challenges. In *Proceedings of the 45th Annual Southeast Regional Conference*, ACM-SE 45, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-629-5, s. 321–326, doi:10.1145/1233341.1233399. Dostupné z: `http://doi.acm.org/10.1145/1233341.1233399`

[6] El-moussa, F.; Mudhar, P.; Jones, A.: Overview of SIP Attacks and Countermeasures. In *Information Security and Digital Forensics*, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, ročník 41, editace D. Weerasinghe, Springer Berlin Heidelberg, 2010, ISBN 978-3-642-11529-5, s. 82–91, doi:10.1007/978-3-642-11530-1_10. Dostupné z: `http://dx.doi.org/10.1007/978-3-642-11530-1_10`

[7] Castelfranchi, C.: Alan Turing's "Computing Machinery and Intelligence". *Topoi*, ročník 32, č. 2, 2013: s. 293–299, ISSN 0167-7411, doi: 10.1007/s11245-013-9182-y. Dostupné z: `http://dx.doi.org/10.1007/s11245-013-9182-y`

[8] International Telecommunication Union. [cit. 2015-05-04]. Dostupné z: `http://www.itu.int/`

[9] Truxa, L.: *Detekce zneužití VoIP ústředen*. Diplomová práce, České Vysoké Učení Technické, 2014.

[10] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, Říjen 2004. Dostupné z: `http://www.ietf.org/rfc/rfc3954.txt`

[11] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Informationl. RFC 7011, Září 2013.

[12] Bartoš, V.; Žádník, M.; Čejka, T.: Liberouter / Cesnet TMC group: Nemea Project. [cit. 2015-04-20]. Dostupné z: `https://www.liberouter.org/technologies/nemea/`

[13] Tartakovsky, A.; Rozovskii, B.; Blažek, R.; aj.: A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *Signal Processing, IEEE Transactions on*, ročník 54, č. 9, Sept 2006: s. 3372–3382, ISSN 1053-587X, doi:10.1109/TSP.2006.879308.

[14] Benáček, T.; Blažek, R.; Kubátová, H.; aj.: Change-Point Detection Method on 100 Gb/s Ethernet Interface.

[15] Čejka, T.: Hardware Accelerated Anomaly Detection in Computer Networks. Technická zpráva, České Vysoké Učení Technické, prosinec 2014.

[16] Hawkins, D. M.; Wu, Q.: The CUSUM and the EWMA Head-to-Head. *Quality Engineering*, ročník 26, č. 2, 2014: s. 215–222, doi:10.1080/08982112.2013.817014.

[17] Autoconf. [cit. 2015-05-01]. Dostupné z: `https://www.gnu.org/software/autoconf/`

[18] Automake. [cit. 2015-05-01]. Dostupné z: `https://www.gnu.org/software/automake/`

[19] Libtool. [cit. 2015-05-01]. Dostupné z: `https://www.gnu.org/software/libtool/`

[20] Git - Distributed version control system. [cit. 2015-05-01]. Dostupné z: `http://git-scm.com`

[21] Libxml. [cit. 2015-05-01]. Dostupné z: http://www.xmlsoft.org/

[22] Xpath. [cit. 2015-05-01]. Dostupné z: http://www.w3.org/TR/xpath/

Příloha **A**

Seznam použitých zkratek

CPD Change Point Detection

CSV Comma-separated values

EWMA Exponentially Weighted Moving Average

IP Internet Protocol

IETF Internet Engineering Task Force

IPFIX Internet Protocol Flow Information Export

ITU International Telecommunication Union

NP-CUSUM Non-Parametric Cumulative Sum

MGCP Media Gateway Control Protocol

DoS Denial of Service

DNS Domain Name System

Nemea Network measurement analysis

PSTN Public Switched Telephone Network

QoS Quality of Service

RFC Request for Comments

RTP Real-time Transport Protocol

RTPS Real-time Transport Streaming Protocol

SIP Session Initiation Protocol

TCP Transmission Control Protocol

TLS Transport Layer Security

TRAP Traffic Analysis Platform

UDP User Datagram Protocol

UniRec Unified Record

VoIP Voice over Internet Protocol

XML Extensible markup language

XSD XML Schema Definition

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `doc` ... dokumentace ke zdrojovým kódům detekčního modulu ve formátu HTML
- `src`
  - `impl` ..... zdrojové kódy detekčního modulu
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `DP_Jisa_Nikolas_2015.pdf` ..... text práce ve formátu PDF