VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ROZŠÍŘENÍ SYSTÉMU PRO ZÁKONNÉ ODPOSLECHY

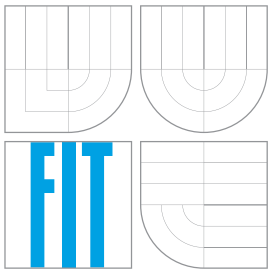
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RADEK HRANICKÝ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ROZŠÍŘENÍ SYSTÉMU PRO ZÁKONNÉ ODPOSLECHY

ADDITIONS TO LAWFUL INTERCEPTION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RADEK HRANICKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LIBOR POLČÁK

BRNO 2014

Abstrakt

V rámci projektu Moderní prostředky pro boj s kybernetickou kriminalitou na internetu nové generace byl vytvořen prototyp systému pro zákonné odposlechy. Tato práce popisuje rozšíření systému, které poskytuje možnost odposlechu aplikačních protokolů (např. e-mailové komunikace) přímo v síti poskytovatele Internetového připojení. Tato nová funkcionalita umožňuje automaticky detekovat a filtrovat související TCP spojení. Odposlech je možné realizovat i v situacích, kdy dosud neznáme identitu (IP adresu) cílového uživatele, nebo v situacích, kdy není jednoduché tuto identitu zjistit (probíhá překlad adres - NAT, uživatel je v Internetové kavárně, za bránou firewall, apod.). Jedním z nejdůležitějších požadavků na vyvíjený prototyp je schopnost rychlého zpracování paketů s maximální propustností a minimálními ztrátami. Z těchto důvodů práce zahrnuje také profilaci výkonnosti, identifikaci kritických míst a jejich následnou optimalizaci.

Abstract

As a part of the Modern Tools for Detection and Mitigation of Cyber Criminality on the New Generation Internet project, a Lawful Interception System was developed. This thesis describes additions to the system, which provide a capability to intercept application protocols (eg. an e-mail communication) directly in a network of an Internet service provider. This new functionality enables automatic detection and filtering of a related TCP transfer. It is also able to handle situations, in which the identity (an IP address) of a target user is not known yet, or when it is difficult to detect it (NAT is in progress, user is at an Internet café, behind the firewall, etc.). One of the most important requirements for the developed prototype is the ability of a fast packet proccessing with maximum throughput and minimal packet loss. Therefore, this thesis also consists of a performance profiling, an identification of critical points and their optimalization.

Klíčová slova

Systém pro zákonné odposlechy, projekt Sec6Net, Dynamická detekce identity uživatele, Analýza událostí na síti, Správa a řízení odposlechů, Záznam a klasifikace paketů, Profilace výkonnosti, Optimalizace kritických míst

Keywords

Lawful Interception System, Sec6Net project, Dynamic user identity detection, Network event analysis, Administration and control of interceptions, Packet recording and classification, Performance profiling, Optimalization of critical points

Citace

Radek Hranický: Rozšíření systému pro zákonné odposlechy, diplomová práce, Brno, FIT VUT v Brně, 2014

Rozšíření systému pro zákonné odposlechy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Libora Polčáka.

.....
Radek Hranický
27. května 2014

Poděkování

Rád bych poděkoval Ing. Liboru Polčákovi za vedení, odbornou pomoc a užitečné rady při řešení mé diplomové práce. Dále bych chtěl poděkovat Ing. Tomáši Martínkovi Ph.D. za podporu, konzultace a zejména vedení LI skupiny v době, kdy byl Ing. Polčák na pracovní stáži v zahraničí. Rovněž děkuji mým kolegům z LI skupiny: Bc. Stanislavu Bártovi, Bc. Barboře Frankové a Bc. Martinu Holkovičovi za vzájemnou motivaci a výbornou spolupráci. V neposlední řadě bych chtěl poděkovat Ing. Petru Matouškovi Ph.D. za úspěšné vedení celého projektu Sec6Net a všem ostatním členům řešitelského týmu za vytvoření skvělého prostředí a spolupráci na společném cíli.

Obsah

Kapitola 1

Úvod

Systém pro zákonné odposlechy (*Lawful Interception System* - LIS) je nástroj, který umožňuje oprávněným orgánům sledovat komunikaci podezřelých subjektů v počítačové, či telefonní síti [7]. Sledování komunikace fyzicky provádí poskytovatel komunikačních služeb na základě soudního příkazu. Informace získané touto činností poté předává oprávněným orgánům. Informacemi můžeme rozumět jak obsah samotné komunikace, tak metadata - identifikace komunikujících stran, časové údaje, apod. Oprávněnými orgány rozumíme orgány činné v trestním řízení. Povinnost poskytnout těmto orgánům v oprávněných případech uvedené informace plyne z legislativy České republiky [22] a také ze směrnic Evropské unie [24].

V rámci projektu *Moderní prostředky pro boj s kybernetickou kriminalitou na internetu nové generace* byl vytvořen prototyp LIS: *Sec6Net Lawful Interception System* - SLIS. Tento systém byl navržen dle standardů ETSI [7] a jeho účelem je zajišťovat komplexní řízení a realizaci odposlechu komunikace v síti Internet. Při reálném nasazení LIS v Internetu často potřebujeme mít možnost odposlechu aplikačních protokolů (např. e-mailové komunikace, IP telefonie, apod.). Současná legislativa [24] odposlech aplikačních protokolů definuje pouze na straně poskytovatelů služeb, např. správců veřejných e-mailů (gmail.com, apod.). Tito poskytovatelé však často působí mimo území České republiky a jejich spolupráce s českými orgány činnými v trestním řízení může být problematická.

Řešením je odposlech aplikačních protokolů přímo v síti národních poskytovatelů Internetového připojení. Právě k tomuto účelu je systém SLIS v rámci této práce rozšířen. Prototyp umožňuje zpracovávat požadavky na odposlech, sledovat aktivitu cíle včetně dynamické změny identity v čase, zachycovat, filtrovat a ukládat obsah komunikace i metadata [18] - informace o událostech na síti (včetně změn identity uživatele), čas zahájení, dobu přenosu, apod.

Dosavadní implementace však umožňovala specifikovat cíl zájmu pouze pomocí IP adresy, MAC adresy síťového rozhraní a několika dalších identifikátorů. Další negativní vlastností systému SLIS byla možnost odposlechu pouze veškeré komunikaci dané IP adresy. Nebylo možné vybrat pouze zájmový provoz na úrovni transportní nebo aplikační vrstvy [14]. Pro praktické použití je často vhodné speficikovat cíl formou identifikátoru aplikační vrstvy a mít možnost zachytit pouze zájmovou komunikaci v rámci konkrétního aplikačního protokolu. Můžeme chtít např. odposlouchávat e-mailovou komunikaci na základě e-mailové adresy, hovor v rámci IP telefonie na základě telefonního čísla, či komunikaci typu Instant Messaging - např. IRC [17] na základě IRC loginu, či čísla IRC kanálu, apod.

V rámci této práce bylo navrženo rozšíření systému SLIS, které umožuje jak detekci identity na základě aplikačního identfikátoru (např. e-mailové adresy), tak odposlech pouze vybrané části provozu daného uživatele (např. pouze e-mailové komunikaci). Rozšíření

umožňuje také identifikovat zájmový provoz v okamžicích, kdy dosud neznáme identitu uživatele (IP adresu) a zachytit zájmový provoz i v případě, kdy nelze identitu uživatele snadno zjistit (např. uživatel je v privátní síti, kde je prováděn překlad adres - NAT, v knihovně, internetové kavárně, apod.). Rozšíření navíc posktuje možnost specifikovat jaký rozsah má odposlech mít. Tedy, zda se chceme omezit pouze na jednu konkrétní IP adresu, na jeden počítač, síťové rozhraní, či chceme zachycovat veškerou komunikaci daného uživatele.

V rámci požadavků (viz přílohu B) na vyvíjený prototyp jsou kladeny vysoké nároky na rychlost zpracování dat. Tyto problémy částečně řeší hardwarové implementace sond pro odposlech (vysokorychlostní a mikro-sonda) vyvíjené v rámci projektu Sec6Net. Obsah komunikace zachycený sondami však stále prochází skrz softwarovou část systému. Kritickým místem je funkční blok nazvaný *Mediační funkce*. Součástí této práce je tedy také profilace výkonnosti tohoto bloku, identifikace kritických míst a jejich následná optimalizace.

Tato práce je rozdělena do několika kapitol. V kapitole 2 je vysvětlena základní myšlenka LIS. Je zde popsána nutnost jejich zavedení z hlediska legislativy ČR a EU. Dále se kapitola zabývá strukturou LIS dle standardů ETSI [7], popisem komunikujích stran a vysvětlením realizace odposlechu. Další část této kapitoly představuje seznámení s projektem Sec6Net a systémem SLIS.

Kapitola 3 popisuje architekturu systému SLIS. Je zde uvedeno zapojení jednotlivých funkčních bloků a popis rozhraní pro komunikaci uvnitř systému i pro komunikaci s jeho okolím. Následuje podrobnější popis jednotlivých funkčních bloků, jejich význam, princip činnosti a implementaci.

Cíle této práce jsou podrobně popsány a vysvětleny v kapitole 4. Jsou zde důkladně rozebrány jednotlivé požadavky na rozšíření systému o podporu aplikačních protokolů i význam optimalizace Mediační funkce spolu se související problematikou.

V kapitole 5 je zdokumentováno rozšíření systému detekce identity systému SLIS a implementace podpory aplikačních protokolů spolu s konfigurovatelnými úrovněmi odposlechu. Jsou zde podrobně popsány nové aspekty, které rozšíření přináší a prostor je věnován také implementaci - je vysvětleno, které funkční bloky musely být upraveny a jakým způsobem.

Kapitola 6 se zabývá profilací výkonnosti Mediační funkce. Jsou zde popsány jak použité metody měření výkonnosti zpracování paketů, tak nalezená kritická místa v tomto funkčním bloku. Hlavním cílem této kapitoly je především optimalizace těchto kritických míst.

Praktickou použitelnost rozšíření systému SLIS a efektivitu provedené optimalizace Mediační funkce pak ukazují experimenty v kapitole 7.

Kapitola 2

Systém pro zákonné odposlechy

Cílem této kapitoly je vysvětlit pojem zákonný odposlech a přiblížit problematiku zákonných odposlechů jak z hlediska legislativy, tak z hlediska technické realizace.

Sekce 2.1 definuje pojmy zákonný odposlech a systém pro zákonné odposlechy. Proč tyto prostředky potřebujeme je vysvětleno v sekci 2.2. Je zde zdůvodněna nutnost existence zákonných odposlechů s ohledem na legislativu ČR a EU.

Od roku 2001 až do současnosti vydal Evropský ústav pro telekomunikační normy (*European Telecommunications Standards Institute* - ETSI)[1] řadu norem, které poskytují standardizovaný popis systému pro zákonné odposlechy. Tento popis specifikuje strukturu systému, jeho konkrétní součásti, princip činnnosti a definuje související pojmy. Celý koncept je vysvětlen v sekci 2.3.

2.1 Definice pojmů zákonný odposlech a systém pro zákonné odposlechy

Zákonné odposlechy představují způsob boje s kriminální, pirátskou, podvodnou, či teroristickou činností. Na základě nařízení orgánů činných v trestním řízení je prováděn záznam a analýza síťové komunikace mezi podezřelými osobami.

- **Zákonný odposlech** (*Lawful Interception* - LI) umožňuje sledovat aktivitu podezřelých osob využívajících veřejných komunikačních prostředků jako jsou telefonní sítě nebo Internet [18]. Dle norem ETSI je pojem zákonný odposlech definován [5, 6, 7] jako činnost prováděná poskytovatelem komunikačních služeb (*Communications service provider* - CSP). Cílem této činnosti je poskytnutí konkrétních informací oprávněným orgánům (orgánům činným v trestním řízení).
- **Systém pro zákonné oposlechy** (*Lawful Interception System* - LIS) je nástroj umožňující oprávněným orgánům realizaci zákonných odposlechů [18].

2.2 Zákonné odposlechy z hlediska legislativy ČR a EU

Z hlediska legislativy ČR nutnost existence LIS plyne ze zákona č. 259/2010 Sb., *o elektronických komunikacích* [22], hlava V, díl 1, odposlech a záznam zpráv: Provozovatel veřejné

[1] http://www.etsi.org

komunikační sítě, či veřejně dostupné služby elektronických komunikací je povinnen na základě vyžádání oprávněného orgánu poskytnout tomuto orgánu provozní a lokalizační údaje. Obdobně je tato povinnost ukotvena také v legislativě Evropské unie [24].

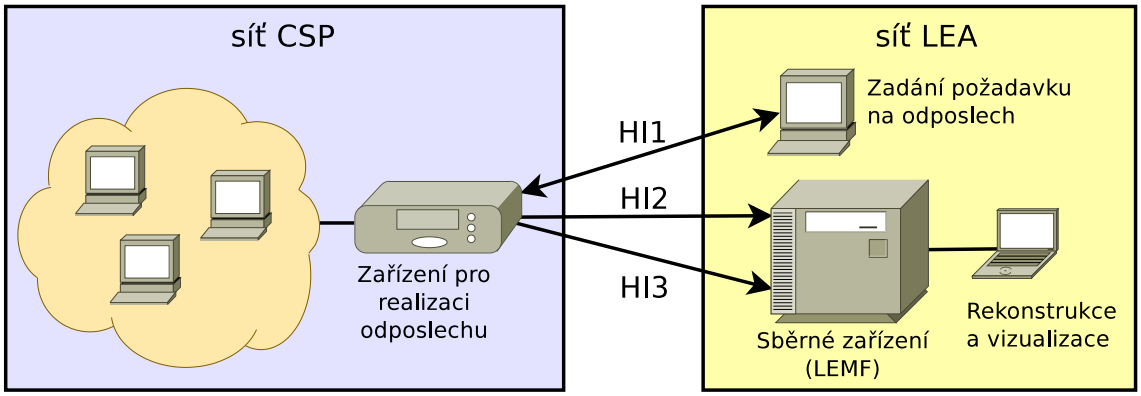
Oba dokumenty uchovávané údaje podrobněji specifikují dle konkrétního typu sítě a povahy přenášených dat. U sítí elektronických komunikací s přepojováním paketů tyto údaje zahrnují identifikaci obou stran, čas zahájení komunikace, dobu přenosu, množství přenesených dat a další údaje.

2.3 Specifikace systému pro zákonné odposlechy dle norem ETSI

2.3.1 Architektura systému

Dle norem ETSI [7] se na zákonném odposlechu podílí dvě strany:

- **Poskytovatel komunikačních služeb** (*Communications service provider* - CSP) - Tím může být například poskytovatel internetu (*Internet service provider* - ISP). Součástí sítě CSP je specializované zařízení pro realizaci odposlechu (sonda, případně další hardware).
- **Oprávněné orgány** (*Law Enforcement Agency* - LEA) - V praxi jde o rganizaci oprávněnou k vyžádání odposlechu a sběru dat. Její součástí je sběrné zařízení (*Law Enforcement Monitoring Facitily* - LEMF), do kterého jsou zaznamenaná data přenášena.



Obrázek 2.1: Architektura systému pro zákonné odposlechy

Obě strany jsou vyobrazeny na obrázku č. 2.1. Mezi CSP a LEA existuje několik rozhraní pro předávání informací (*Handover Interface* - HI) [9]:

- **Rozhraní HI1** - slouží k předávání požadavků na odposlech ze strany LEA. Toto rozhraní může být realizováno softwarově, i manuálně (dopisem, faxem, apod.)
- **Rozhraní HI2** - zajišťuje přenos metadat o aktivitě sledovaných cílů (změny adres, připojení/odpojení ze sítě, apod.)
- **Rozhraní HI3** - zde se přenáší samotný obsah zachycené síťové komunikace.

2.3.2 Scénář realizace odposlechu

V typickém případě LEA skrz rozhraní HI1 požádá CSP o odposlech komunikace mezi podezřelými osobami. CSP na základě tohoto požadavku začne poskytovat LEA data. Zachycenou komunikaci CSP přeposílá do LEMF. Metadata (identifikace, informace o chování cílů) jsou do LEMF zasílána přes rozhraní HI2, zatímco obsah zaznamenané komunikace je zasílán přes rozhraní HI3. LEA získaná data podrobuje analýze. Ta se může skláda např. z filtrace, vizualizace, apod. Konkrétní bod, z jehož pohledu je komunikace zaznamenávána, je označován jako *Internal Intercepting Funciton* (IAP) [7].

Kapitola 3

Systém SLIS

SLIS (*Sec6Net Lawful Interception System*) představuje systém pro zákonné odposlechy vyvíjený v rámci projektu Sec6Net (viz sekci 3.1). Cílem této kapitoly je vysvětlit, co představuje systém SLIS a jaké jsou jeho cíle. Dalším cílem je popsat architekturu systému, přiblížit jak celý systém funguje a vysvětlit účel jednotlivých funkčních bloků, kterými je tvořen.

Účel a cíle systému vysvětluje sekce 3.2. Sekce 3.3 popisuje strukturu systému SLIS. Jsou zde popsány jednotlivé funkční bloky, jejich a rozhraní mezi nimi. V dalších sekcích je každý z funkčních bloků rozebrán podrobněji.

3.1 Projekt Sec6Net

Sec6Net je zkratkou pro projekt *Moderní prostředky pro boj s kybernetickou kriminalitou na Internetu nové generace*. Projekt Sec6Net je zaměřen na výzkum a vývoj prostředků monitorování provozu sítí, analýzu záznamů provozu sítí a metod prostředků zabezpečení lokálních sítí. Důraz je přitom kladen zejména na sítě nové generace s protokolem IPv6. Cíle projektu spadají do dvou tématických oblastí. První oblastí je monitorování síťového provozu, druhou oblastí je výzkum metodik pro správu sítě a prevence nežádoucích aktivit.

Jedním z cílů z oblasti monitorování síťového provozu je vytoření prototypu systému pro sběr a uchování dat na lokální síti dle vyhlášky č.450/2005 Sb. pro komunikaci protokolem IPv6. Tento prototyp: *Sec6Net Lawful Interception System* - SLIS bude podrobně popsán v kapitole 3.

3.2 Účel systému

SLIS představuje konkrétní implementaci systému pro zákonné odposlechy (*Lawful Interception Sytem* - LIS).

Jeho cílem není konkurovat existujícím komerčním systémům jako např. VERINT[1], TROVICOR[2], DeepProbe[3] nebo QOSMOS[4], ale poskytnout jednoduchý, dostatečně výkonný, funkční a snadno použitelný prototyp, který je v souladu s normami ETSI [7]. Prototyp by měl být jednak schopen praktického nasazení, jednak by měl sloužit jako platforma

[1] http://www.verint.com/solutions/communications-cyber-intelligence/products/reliant/index

[2] http://trovicor.com

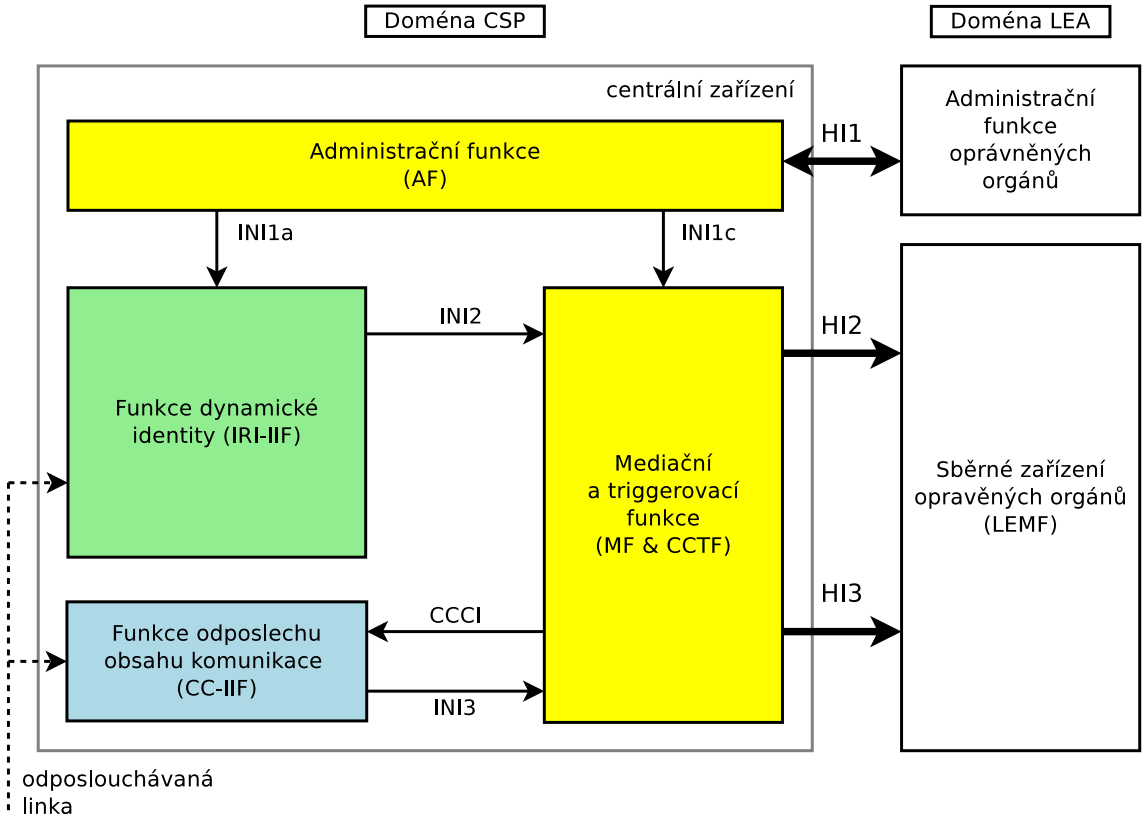
[3] http://new-www.ipfabrics.com/products/DeepProbe.php

[4] http://www.qosmos.com/

pro další vývoj a výzkum. Ten by se měl týkat zejména nových technik dynamické identifikace uživatele v prostředí sítí IPv6, sběru dat, rekonstrukce a vizualizace zachyceného provozu. Zároveň by měl vyvíjený prototyp sloužit jako testovací prostředí pro vývoj mikrosody a vysokorychlostní sondy (viz sekci 3.1). Přesné požadavky na vyvíjený prototyp jsou uvedeny v příloze B.

3.3 Architektura systému

Struktura funkčních bloků systému SLIS je vyobrazena na obrázku 3.1.



Obrázek 3.1: Struktura systému SLIS (pozn. Pro zjednodušení jsou šipky orientovány pouze ve směru přenosu užitečných dat. Potvrzovací zprávy, které se přenášejí v opačném směru, jsou zanedbány.)

Schéma je rozděleno do domény CSP a domény LEA. V doméně LEA se nachází administrační funkce, která představuje orgán LEA zadávající CSP požadavek na odposlech. Tato interakce dle ETSI [7] představuje rozhraní HI1. Dále je zde sběrné zařízení (LEMF), které uchovává data získaná z rozhraní HI2 a HI3. Tato data poté může LEA podrobit další analýze, provádět jejich rekonstrukci, vizualizaci, apod. Rozhraní HI1 až HI3 jsou popsána v kapitole 2 v sekci 2.3.

V doméně CSP je nachází systém centrální zařízení systému SLIS (označeno šedým rámečkem), které je připojeno k samotné odposlouchávané lince. Systém se skládá ze čtyř celků, přičemž blok CC-IIF (viz dále) může existovat buď v softwarové podobě jako součást centrálního zařízení, nebo jako samostatné zařízení. Samostatným zařízením rozumíme

hardwarovou CC-IIF sondu, která se nachází mimo centrální zařízení systému SLIS. Těchto sond může být k systému připojeno i více. Mezi funkčními bloky systému existují následující rozhraní:

- **INI1** až **INI3** - vnitřní rozhraní pro komunikaci jednotlivých bloků (Internal Network Interface - INI) - Tato rozhraní až na několik úprav odpovídají normám ETSI [7].
- **CCCI** - rozhraní pro řízení obsahu komunikace (*Content of Communication Control Interface*) - Toto rozhraní slouží pro konfiguraci bloku CC-IIF. V případě, že blok CC-IIF existuje jako samostatné zařízení bude rozhraní vést mimo centrální zařízení systému SLIS.

Vstupní požadavky na aktivaci odposlechu jsou zadávány skrz rozhraní HI1. Tyto požadavky přijímá Administrační funkce (*Administration Function* - AF), která je zpracovává a pomocí rozhraní INI1a a INI1c konfiguruje zbytek systému.

Protože identita uživatele se může měnit (např. přidělení nové IP adresy, navázání spojení, ukončení spojení, apod.) [16], je součástí systému Funkce dynamické identity (*Intercepted Related Information - Internal Interception Function* - IRI-IIF). IRI-IIF sleduje události na síti, vytváří a udržuje model sítě (identita uživatelů, adresování, existující spojení, apod.). O událostech, které souvisí s některým z odposlechů neprodleně informuje formou metadat (IRI zpráv [7]) skrz rozhraní INI2 Mediační funkci.

Funkce odposlechu obsahu komunikace (*Content of Communication - Internal Interception Function* - CC-IIF) představuje samotnou sondu, či více sond realizujících odposlech. CC-IIF zachycuje obsah komunikace z odposlouchávané linky. Dle požadavků CCTF filtruje zájmová data a přes rozhraní INI3 tato data posílá MF. Blok CC-IIF může existovat buď v softwarové podobě jako součást systému SLIS (jak je vyobrazeno na obrázku 3.1) nebo jako samotné zařízení v podobě mikro-sondy, či vysokorychlostní sondy.

Mediační funkce (*Mediation Function* - MF) a Triggerovací funkce (*Content of Communication Trigger Function* - CCTF) z norem ETSI [7] byly z důvodu zjednodušení správy odposlechů v systému SLIS spojeny do jednoho funkčního bloku [18]. Toto rozhodnutí bylo inspirováno architekturou LIS firmy Cisco [1]. MF přijímá jak metadata o identitě odposlouchávaného z rozhraní INI2, tak samotný obsah komunikace z rozhraní INI3. S těmito daty dále pracuje a zajišťuje jejich centrální správu. Metadata o zachycené komunikaci jsou posílána LEMF skrz rozhraní HI2, samotný obsah komunikace pak skrz rozhraní HI3. Funkce CCTF je pak zodpovědná za konfiguraci CC-IIF sondy - skrz rozhraní CCCI říká sondě, co „má“ odposlouchávat.

3.4 Použité identifikátory

V dalším textu budou zmiňovány různé identifikátory, které systém SLIS používá. Tato sekce poskytuje jejich souhrn a popis.

Použité identifikátory v systému SLIS:

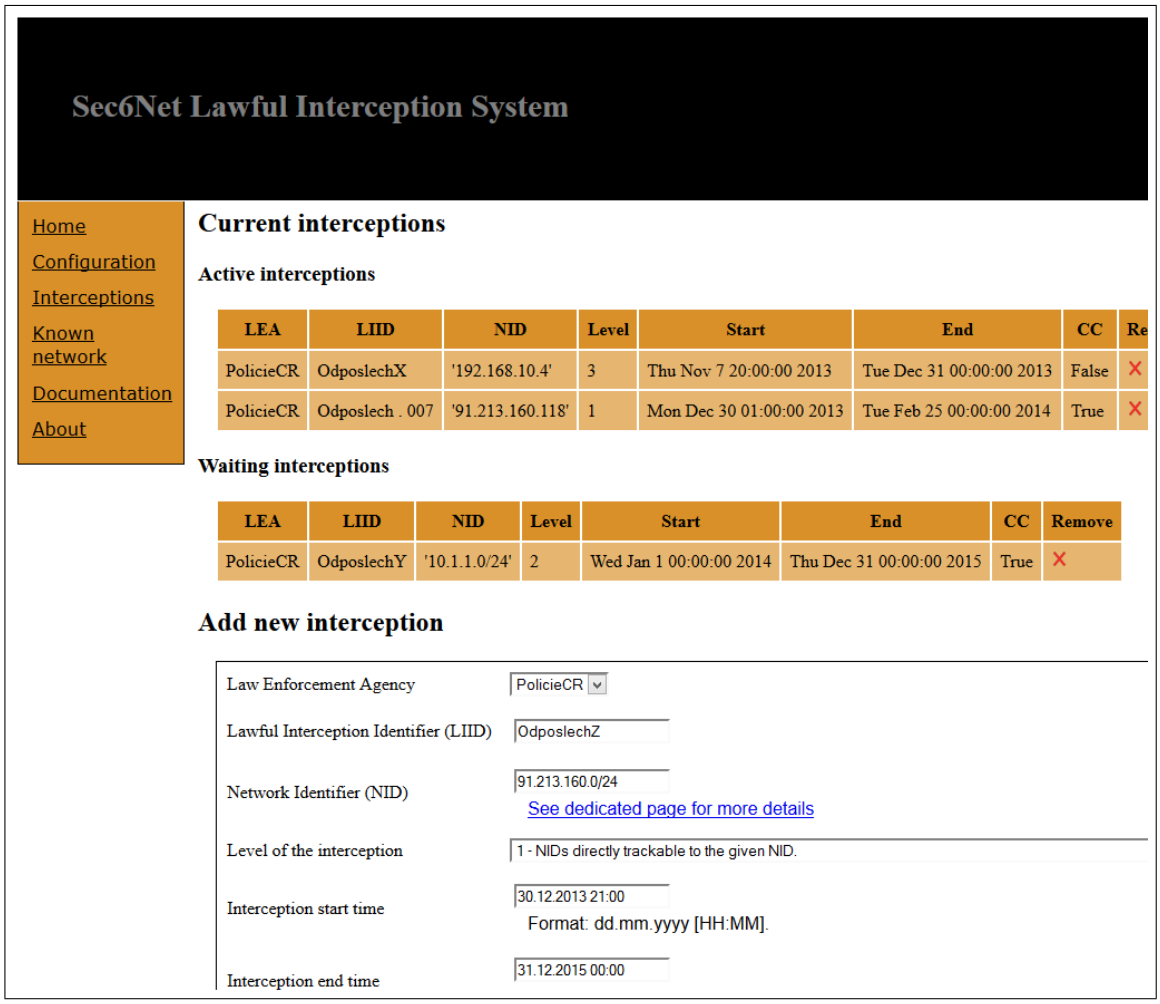
- *Lawful Interception IDentifier* (LIID) - Řetězec alfanumerických znaků, který jednoznačně identifikuje odposlech [11]. LIID je dodáván LEA a je unikátní nejen v rámci LEA, ale také na mezinárodní úrovni [12] (řetězec obsahuje dvoupísmennou zkratku státu definovanou ISO 3166-1 [15]). Všechna data přenášena rozhraními HI2 a HI3 musí být tímto identifikátorem označena.

- *Network IDentifier* (NID) - Identifikátor používaný v síťových protokolech k označení účastníků komunikace. V současnosti LIS podporuje MAC adresu, statickou IPv4 a IPv6 adresu nebo obecně rozsah IP adres definovaný adresou sítě a maskou.
- *Network IDentifier, Content of Communication* (NID_{CC}) - Speciální případ NID. Lze jej přímo zadat sondě CC-IIF k odposlouchávání. Jako NID_{CC} uvažujeme: IPv4 adresu, IPv6 adresu a po nových úpravých také další identifikátory (viz kapitolu 5).
- *HI1 IDentifier* (HI1ID) - Jednoznačná identifikace odposlouchávané osoby, která je součástí požadavku na odposlech ze strany LEA. Tento identifikátor může být tvořen např. MAC adresou, IPv4/IPv6 adresou nebo zcela obecně např. rodným číslem, adresou trvalého bydliště apod. V případě obecné identifikace je úkolem pověřeného pracovníka na straně operátora tuto identitu převést na takovou, jakou je možné použít ve zbytku systému. Pověřený pracovník obvykle využije interní databáze obsahující seznam zákazníků a převede HI1ID na NID.
- *Communication IDentifier* (CID): Identifikátor, který jednoznačně identifikuje komunikaci [11, 12]. CID se skládá z následujících částí:
  - unikátního ID operátora přiděleného LEA,
  - NID, kterého se odposlech týká,
  - Communications Identity Number (CIN) identifikuje sezení, nebo komunikaci v rámci jednoho odposlechu, který je identifikován pomocí LIID,
  - Delivery Country Code (DCC) je dvoupísmenné označení země, kde se nachází MF.
- *System IDentifier* (SID): 32-bitový celočíselný identifikátor označující množinu odposlechů (LIID). SID byl do LIS zaveden kvůli minimalizaci dat přenášených ze sond IRI-IIF a CC-IIF do MF & CCTF. Mapování SID na množinu LIID se uchovává v MF & CCTF, které také spravují přidělování nových SID. Výhodou použití SID je jeho pevná velikost, což umožňuje vytvářet jednodušší hlavičky pro data zasílaná z IRI-IIF a CC-IIF do MF & CCTF.
- *Reason IDentifier* (RID): efektivně 16-bitový identifikátor využívaný blokem MF & CCTF a sondami CC-IIF. Pokud MF & CCTF žádá sondu CC-IIF skrz rozhraní CCCI o odposlech určitého NID_{CC}, je v této žádosti RID shodné se SID. U odpovědi na tuto žádost představuje RID číslo pravidla přiřazeného sondou k zaslanému požadavku/pravidlu. Existuje surjektivní zobrazení (ID sondy, RID) $\rightarrow$ SID. Je povinností sondy zajistit, aby dané RID bylo unikátní v rámci pravidel nakonfigurovaných v daný okamžik a v průběhu odposlechu se nezměnilo.

3.5 Administrační funkce (AF)

Administrační funkce (AF) zpracovává požadavky od jedné, či více LEA. Požadavky jsou přijímány skrz rozhraní HI1 [9]. Norma ETSI TS 101 671 [11] říká, že rozhraní HI1 může být řešeno jak manuálně, tak automaticky (elektronicky). Z této normy ovšem také vyplývá, že přímá aktivace/deaktivace/modifikace odposlechu by měla být v kompetenci CSP a tedy neměla být přímo přístupná LEA.

V současnosti se uvažuje manuální rozhraní HI1 - např. dopis poštou, který přijme pracovník CSP a požadavek na aktivaci odposlechu do systému SLIS zadá pomocí webového uživatelského rozhraní, které vyobrazeno na obrázku 3.2.



Obrázek 3.2: Správa odposlechů systému SLIS přes webové uživatelské rozhraní

Pracovník CSP [11] zadá do webového rozhraní požadavek na nový odposlech. Je-li požadavek korektní, je předán AF k dalšímu zpracování. Požadavek na nový odposlech zahrnuje následující údaje:

- **LEA** - název LEA, která odposlech zadává
- **LIID** - identifikátor odposlechu (Lawful Interception IDentifier)
- **NID** - síťový identifikátor (Network IDentifier) - např. IP adresa speficikující odposlouchávaný cíl
- **Úroveň** - rozsah odposlechu - nově přidaný parametr (viz kapitolu 4)
- **Datum a čas zahájení odposlechu** - doba, od které je platné soudní povolení na získávání dat

- **Datum a čas ukončení odposlechu** - doba, do které je platné soudní povolení na získávání dat
- **CC ano/ne** - informace, zda bude zaznamenáván i obsah komunikace (*Content of Communication* - CC) nebo pouze metadata o komunikaci.

AF v sobě uchovává dvě fronty. Fronta čekajících odposlechů je prioritní fronta, kde nejvyšší prioritu má odposlech s nejbližším časem zahájení. Druhou frontou je fronta aktivních odposlechů, kde nejvyšší prioritu má odposlech s nejbližším časem ukončení.

Při přijetí nového požadavku porovná AF jeho časové údaje s aktuálním časem. Mohou nastat dvě situace:

(a) Ještě nenastala doba, od které je platné soudní povolení na záskávání dat. Systém tedy vloží požadavek do speciání fronty čekajících odposlechů.

(b) Doba platnosti soudního povolení již nastala. Odposlech je ihned aktivován a vložen do fronty aktivních odposlechů.

Naopak, jakmile má nastat čas ukončení některého aktivního odposlechu, AF vyčká 10 sekund a až poté jej odebere ze zbytku systému. Toto zpoždění bylo zavedeno, aby bylo zaručeno bezproblémové zpracování dat zachycených těsně před povoleným koncem odposlechu [18].

Zprávy posílané skrz rozhraní INI1a a INI1c mají specifický formát. Rozhraním INI1a se posílají výše uvedené údaje z požadavku na odposlech a navíc je k nim přidán CID s dosud neinicializovanou hodnotou CIN (viz sekci 3.4). Skrz rozhraní INI1c se posílají pouze nejnutnější informace: LIID, čas začátku a konce odposlechu a informace, zda bude zaznamenáván také obsah komunikace.

Současná verze AF je implementována v jazyce Python 3 [19]. Webové rozhraní HI1 využívá jazyků HTML a PHP[5].

3.6 Funkce dynamické identity (IRI-IIF)

Funkce dynamické identity (IRI-IIF) představuje důležitou součást systému SLIS. Jejím cílem je sledovat události na odposlouchávané lince a poskytovat zbytku systému informace o identitě odposlouchávaného cíle. V podsekci 3.6.1 bude popsána vnitřní struktura celého bloku. V podsekcích 3.6.3 až 3.6.5 budou popsány tři tabulky, které IRI-IIF používá k uložení informací o aktuálním stavu. V podsekci 3.6.6 pak bude vysvětlena činnost jádra IRI-IIF.

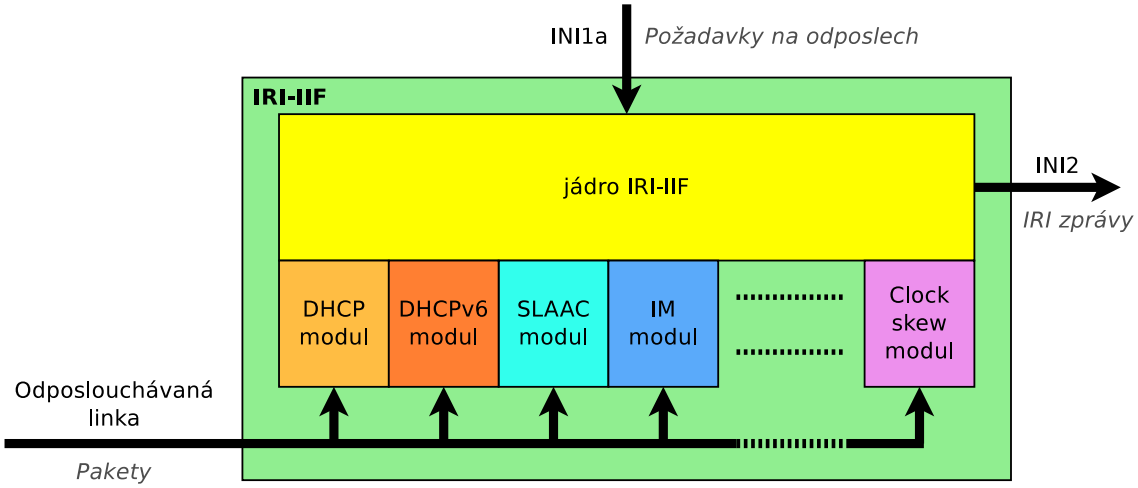
3.6.1 Vnitřní struktura bloku IRI-IIF

Funkce dynamické identity (IRI-IIF) se skládá z jádra IRI-IIF a množství modulů (např. modul pro DHCP, modul pro SLAAC, apod.). Vnitřní struktura bloku IRI-IIF je znázorněna na obrázku 3.3.

K modulům je přivedena odposlouchávaná linka. Každý z modulů zkoumá data přenášená po lince ze specifického úhlu pohledu. Modul může zkoumat události konkrétního

[5] http://php.net/

protokolu (např. přidělování adres pomocí DHCP), rodiny protokolů (navazování spojení a komunikace YMSG, IRC, aj.), či jiná specifika přenosu (např. clock skew - odchylku hodin dvou počítačů v síti). O zaznamenaných událostech pak moduly pomocí specializovaných IRI zpráv informují jádro IRI-IIF [7].



Obrázek 3.3: Struktura bloku IRI-IIF

3.6.2 IRI zprávy

Komunikace modulů s jádrem IRI-IIF probíhá pomocí specializovaných IRI zpráv [10, 18]. Pokud některý z modulů pošle IRI zprávu, znamená to, že nastala určitá specifická událost na síti. Každá IRI zpráva obsahuje typ zprávy, textový popis události a množinu NID, které s událostí souvisí. Rozlišujeme základní čtyři typy IRI zpráv:

- **IRI BEGIN** - Vznikla nová asociace mezi NID v síti (např. počítači s MAC adresou MAC_X byla přiřazena IPv4 adresa IPv4_X).
- **IRI END** - Skončila platnost existující asociace mezi NID v síti (např. počítači s MAC_X již vypršelo období platnosti přidělen adresy IPv4_X).
- **IRI CONTINUE** - Obnova platnosti existující asociace mezi NID v síti (např. přiřazení adresy IPv4_X počítači s MAC_X bude i nadále platné).
- **IRI REPORT** - Jde o čistě informativní sdělení (např. počítač s MAC_X požádal o přidělení IPv4 adresy, přidělení adresy selhalo, apod.)

3.6.3 Tabulka odposlechů

Tabulka odposlechů (*Interception table*) obsahuje odposlechy zadané administrační funkcí. Příklad ukazuje tabulka 3.1. Jednotlivé sloupce odpovídají popisu v sekci 3.5. Schéma tabulky je zjednodušené - ve skutečnosti je v tabulce ukládán celý CID. Pro vysvětlení principu IRI-IIF je však podstatný pouze CIN, který je jednou ze součástí CID. Jak bylo řešeno,

při konfiguraci IRI-IIF se rozhraním INI1a posílá požadavek s neinicializovaným CINem. Jeho inicializace a další úpravy jsou právě úlohou IRI-IIF. CIN v záznamu tabulky odposlechů bude odpovídat vždy nejvyššímu CIN pro dané LIID z tabulky CIN [10]. Sloupec *úroveň*, který je zde také uveden, je již součástí nově provedené modifikace. Tato modifikace je popsána v kapitole 5.

LIID	NID	CIN	CC	úroveň	začátek odposlechu
X	MAC: 00:25:90:0f:81:37	1	ne	1	1. 1. 2013 13:37
Y	IPv4: 192.168.1.63	4	ano	1	10. 4. 2014 20:00

Tabulka 3.1: Příklad tabulky CIN v bloku IRI-IIF

3.6.4 Tabulka CIN

Tabulka CIN (*CIN table*) ukládá informace o hodnotách CIN přiřazených dvojicím NID_{CC} + LIID. Příklad znázorňuje tabulka 3.2. Každému nově nalezenému NID_{CC}, který souvisí s některým z odposlechů, je přiřazen nový CIN (o 1 vyšší než poslední přiřazený). Tyto změny se promítají také do tabulky odposlechů, kde je u každého odposlechu uchováván vždy nejvyšší přiřazený CIN pro odpovídající LIID [10].

NID_{CC}	LIID	CIN
MAC: 00:25:90:0f:81:37	X	1
IPv4: 192.168.1.63	Y	4
IPv6: 2001:0db8:3c4d::abcd:ef12	Y	3

Tabulka 3.2: Příklad tabulky CIN v bloku IRI-IIF

3.6.5 Tabulka NID

Tabulka NID (*NID table*) představuje model současného stavu sítě. Její obsah je vytvářen a modifikován na základě IRI zpráv [10, 18] přijatých od modulů. Při přijetí zprávy typu *IRI BEGIN* vzniká nový záznam v tabulce s odpovídajícími NID obsaženými ve zprávě. V rámci nových úprav (viz kapitolu 5) jsem se po konzultaci s vedoucím práce rozhodl postupovat stejným způsobem i v případě přijetí zprávy typu *IRI CONTINUE*. Důvodem tohoto doplnění je předejít ztrátám informace v případě, kdy by spuštění určitého IRI-IIF modulu bylo provedeno až po proběhnutí specifické události, při které modul generuje *IRI BEGIN*. V případě přijetí zprávy typu *IRI END* je záznam z tabulky smazán (pokud existuje). Zprávy typu *IRI REPORT* jsou pouze informativní a na obsah tabulky NID nemají vliv.

Ukázkovou situaci představuje tabulka 3.3. První sloupec obsahuje identifikátor IRI-IIF modulu, který událost ohlásil. Každý další sloupec představuje jeden existující NID, který systém podporuje. (Pozn. DUID a IPv6 adresy byly zkráceny z důvodu limitu šířky tabulky.)

Modelová situace a posloupnost událostí, při které vznikl obsah tabulky 3.3, je popsána v příloze C.

ID modulu	MAC	DUID	IPv4	IPv6	PPP_L	$RADIUS_L$	...
PPPoE	00:25:90:0f:81:37				radek		
RADIUS	08:00:69:02:01:fc					standa	
DHCP	00:13:a9:a7:ef:4b		10.1.0.6				
RADIUS	00:80:c7:0f:4b:22		10.2.0.4			tomas	
SLAAC	00:80:c7:e4:81:1f			fd02..			
DHCPv6		0003..		fd02..			

Tabulka 3.3: Příklad tabulky NID v bloku IRI-IIF

Popisovaná implementace tabulky NID ukazuje situaci před provedením úprav. Vzhledem k nově vzniklým požadavkům (více stejných NID v jednom záznamu, konfigurovatelné úrovně odposlechů, apod.) se ukázala jako nevhodná a byla nahrazena jinou reprezentací. Důvod této změny a provedné úpravy jsou popsány v kapitole 5.

3.6.6 Činnost jádra IRI-IIF

Jádro IRI-IIF je realizováno jako událostmi řízený program. Událostmi se rozumí jednak příchozí požadavky od AF z rozhraní INIa, jednak IRI zprávy ze strany modulů.

- Obdrží-li jádro IRI-IIF z rozhraní INI1a požadavek na nový odposlech:
  1. Přidá odposlech jako novou položku do tabulky odposlechů.
  2. Analyzuje obsah tabulky NID a idenfitikuje všechy NID_{CC}, které s daným odposlechem souvisí.
  3. Pro každý NID_{CC} vygeneruje zprávu *IRI BEGIN* (odposlech na již aktivní komunikaci) a pošle ji na rozhraní INI2.
- Obdrží-li jádro IRI-IIF z rozhraní INI1a požadavek na zrušení odposlechu:
  1. Odstraní odpovídající položku z tabulky odposlechů.
- Obdrží-li jádro IRI-IIF IRI zprávu ze strany modulů:
  1. Zkontroluje, zda se některý z NID ve zprávě netýká některého z aktivních odposlechů. Pokud ano, potom:
     (a) Identifikuje všechny odposlechy (LIID identifikátory), ke kterým se IRI zpráva vztahuje.
     (b) Rozkopíruje IRI zprávu pro všechny příslušné LIID a všechny kopie pošle na rozhraní INI2.
  2. Vloží/odstraní/aktualizuje záznam v tabulce NID. Konkrétní činnost se liší podle typu zprávy:
     – *IRI-BEGIN*
       (a) Vloží do tabulky NID nový záznam obsahující ID modulu a příslušné NID.
       (b) Přidá nový záznam do tabulky CIN pro všechny LIID, kterách se NID týká. U dotčených odposlechů aktualizuje hodnoty CIN také v tabulce odposlechů na nejvyšší CIN pro dané LIID.

– *IRI END*
  (a) Odstraní z tabulky NID záznam vytvořený odpovídající zprávou *IRI BEGIN*.
– *IRI CONTINUE*
  (a) Pokud tento ještě neexistuje, vloží do tabulky NID nový záznam obsahující ID modulu a příslušné NID.
– *IRI REPORT*
  (a) (Žádné další akce)

3.7 Mediační a triggerovací funkce (MF & CCTF)

Mediační a triggerovací funkce (MF & CCTF) provádí centrální správu všech aktivních odposlechů. Jejím účelem je jak konfigurace sondy CC-IIF, tak filtrování získaných dat a jejich předávání na výstup. Protože různé odposlechy se mohou z hlediska cíle vzájemně překrývat (případně může být jeden podmnožinou druhého), byl zaveden identifikátor SID (viz sekci 3.4), který označuje množinu odposlechů. V podsekci 3.7.1 bude popsáno, jak probíhá v MF&CCTF správa odposlechů na základě informací získaných od AF. Podsekce 3.7.2 vysvětluje zpracování informací získaných od bloku IRI-IIF a mapování LIID na SID. V podsekci 3.7.3 bude popsáno rozhraní CCCI a konfigurace sondy CC-IIF.

Většina bloku MF&CCTF je implementována v jazyce Python 3 [19]. Některé části jsou však z důvodu výkonnosti implementovány v jazyce Cython [2]. Cython představuje nadmnožinu jazyka Python. Zdrojový kód v jazyce Cython je určen k automatizovanému převodu do zdrojového kódu v jazyce C a případnému propojení s již existujícím nativním kódem v jazyce C.

3.7.1 Správa odposlechů a tabulka LIID

Blok MF&CCTF je konfigurován AF přes rozhraní INI1c [18]. Jak již bylo řečeno (viz sekci 3.5), předávanými údaji jsou pouze LIID, začátek a konec odposlechu a informace, zda budeme zaznamenávat také obsah komunikace (CC) nebo pouze metadata (IRI zprávy). Tyto informace si blok MF&CCTF ukládá do tzv. *tabulky LIID*. Podle informací z této tabulky MF&CCTF zpracovává a ukládá data ze sondy CC-IIF. Příklad představuje tabulka 3.4. První sloupec je LIID daného odposlechu. Sloupec t_s představuje čas začátku a t_e čas konce odposlechu. Dále je zde typ odposlechu: odchytávání CC nebo pouze IRI. V současné implementaci systému SLIS jsou rozhraní HI2 a HI3 implementována jako výstup souboru. Z těchto důvodu tabulka LIID obsahuje také názvy těchto souborů. Ty jsou odvozeny z LIID a konfigurace systému. Metadata jsou ukládána v textové podobě do souboru s příponou `.hi2`, obsah komunikace je ukládán ve formátu PCAP[6]. Víme také (ze sekce 3.5), že AF konfiguruje blok MF&CCTF 10 sekund před stanoveným začátkem odposlechu a konfiguraci odposlechu ruší 10 sekund po jeho skončení. Odstranění dat nasbíraných sondami mimo toto rozmezí je také úlohou bloku MF&CCTF [18].

[6] http://www.tcpdump.org/pcap.html

LIID	t_s	t_e	typ	soubor pro HI2	soubor pro HI3
X	1. 1. 2011	1. 1. 2014	IRI	X.hi2	X.pcap
Y	2. 1. 2011	10. 4. 2016	IRI + CC	Y.hi2	Y.pcap

Tabulka 3.4: Příklad tabulky LIID v bloku MF&CCTF

3.7.2 Mapování LIID na SID

Blok CC-IIF (sonda CC-IIF) pracuje na síťové vrstvě ISO/OSI modelu [14] a zachytává pakety odposlouchávaných cílů na základě identifikátorů typu NID_{CC} (tedy IPv4, IPv6 adresy. Pokud je cílem odposlechu jiný NID než NID_{CC}, potom je potřeba k němu dohledat konkrétní NID_{CC}, který lze odposlouchávat. Toto je úkolem bloku IRI-IIF, který byl popsán v sekci 3.6. IRI-IIF sleduje události na síti a je schopen k požadovanému vstupnímu NID nalézt jeden nebo více NID_{CC}. Protože nalezených NID_{CC} může být i více (např. počítač má více IP adres a zároveň jeden NID_{CC} může spadat i do více různých odposlechů, byl zaveden identifikátor SID zastupující množinu odposlechů (viz sekci 3.4). Díky tomu můžeme zachycená data označit jedním SID a jejich kopírování a rozesílání různým LEA realizovat až v rámci bloku MF&CCTF. Tímto přístupem se minimalizuje nezbytná komunikace mezi bloky systému a je také redukována potřebná šířku pásma na rozhraní INI3, skrz které jsou přenášena zachycená data [18].

Případná duplikace zachycených dat a jejich zasílání jednotlivým LEA se provádí až MF&CCTF. Z tohoto důvodu je v bloku MF&CCTF také realizován samotný algoritmus vytvářející relaci přiřazující LIID k SID. K tomuto účelu si blok MF&CCTF uchovává tzv. *tabulku SID*, která tuto relaci reprezentuje. Vždy, když je do systému přidán nový požadavek na odposlech statické IP adresy nebo dojde k dynamické změně IP adresy odposlouchávaného cíle (např. skrze protokol DHCP), může obecně dojít ke změně relace přiřazující LIID a SID. Blok IRI-IIF, který tyto události sleduje, informuje blok MF&CCTF. MF&CCTF upraví odpovídajícím způsobem relaci přiřazující LIID a SID a o provedené změně informuje oba bloky IRI-IIF a CC-IIF, které novou nebo upravenou hodnotu SID potřebují pro označení zachytávaných dat nebo vytvářených metainformací [18].

Příklad *tabulky SID* vyobrazuje tabulka 3.5. Její podrobný popis, algoritmus mapování a modelová situace, při které tabulka vznikla, jsou uvedeny v příloze D

SID	NID_{CC}	LIID	CID
1	IPv4: 10.0.0.1/32	X Y	(LEA_1, MAC: 00:11:22:aa:bb:cc, 1, CZ) (LEA_2, IPv4: 10.0.0.1/32, 5, CZ)
2	IPv6: 2001:db8::5/128	X	(LEA_1, MAC: 00:11:22:aa:bb:cc, 2, CZ)

Tabulka 3.5: Příklad tabulky SID v bloku MF&CCTF

3.7.3 Konfigurace sondy CC-IIF

O konfiguraci sondy CC-IIF se stará funkce CCTF, která byla kvůli jednodušší správě odposlechů spolu s MF sjednocena [18] do společného bloku MF&CCTF. Řízení probíhá pomocí rozhraní CCCI formou posílání zpráv typu žádost/odpověď. MF&CCTF posílá žádost, CC-IIF vrací odpověď. Žádost ze strany MF&CCTF obsahuje:

- hlavičku (identifikace sondy - sond může být více), verze, apod.,
- typ akce - přidání/odebrání pravidla pro odposlech,
- samotný obsah pravidla - IPv4/IPv6 adresa, nově také typ transportního protokolu a zdrojový + cílový port (viz kapitolu 5),
- RID (viz sekci 3.4) - v požadavku je tato hodnota shodná se SID,
- SID.

Odpověď ze strany CC-IIF je totožná s žádostí až na hodnotu RID, která obsahuje nově zvolené číslo, které sonda přiřadila danému odposlechu. Pomocí hodnoty RID jsou označovány pakety s obsahem komunikace, kterou sonda přeposílá skrz rozhraní INI3. Je povinností sondy zajistit, aby dané RID bylo unikátní v rámci pravidel nakonfigurovaných v daný okamžik a v průběhu odposlechu se nezměnilo. Hodnotu RID v odpovědi sonda bloku MF&CCTF sděluje, aby bylo pakety přijaté z INI3 přiřazovat konkrétnímu SID a poté předávat přes HI3 na výstup konkrétní LEA. Z tohoto důvodů si blok MF&CCTF udržuje (kromě již zmíněných tabulek) také speciální převodní tabulku pro převod RID na SID pro každou sondu CC-IIF.

3.8 Funkce odposlechu komunikace (CC-IIF)

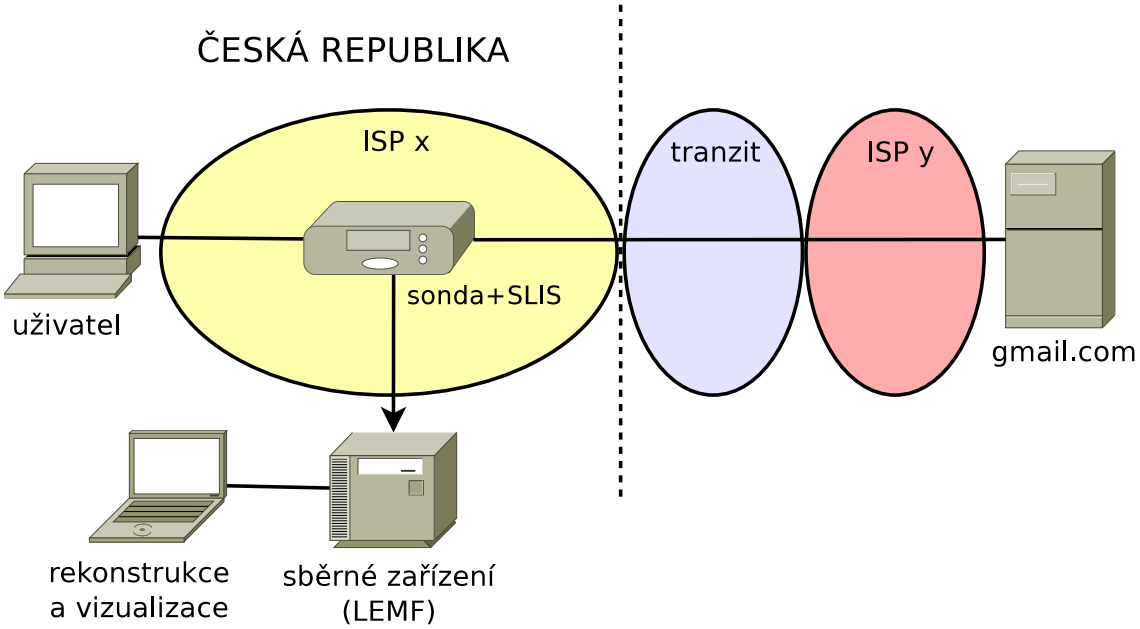
Blok CC-IIF představuje sondu realizující samotný odposlech komunikace. Blok CC-IIF může být řešen jednak softwarově jako součást systému SLIS, jednak jako samostatné zařízení v podobě mikro-sondy nebo vysokorychlostní sondy (viz sekci 3.1). Dle norem ETSI [7] nesmí být žádná z komunikujících stran schopna detekovat, že jejich komunikace je odposlouchávána. Sonda CC-IIF tedy působí jako neviditelné zařízení na síti. Konfigurace je prováděna blokem MF&CCTF přes rozhraní CCCI. Zachycená komunikace je poté bloku MF&CCTF předávána přes rozhraní INI3.

Kapitola 4

Cíle práce

Tato kapitola se skládá ze dvou částí. V sekci 4.1 jsou popsány cíle vylepšení detekce identity a odposlechu aplikačních protokolů. Sekce 4.2 vysvětluje nutnost profilace výkonnosti a optimalizace bloku MF&CCTF.

4.1 Vylepšení detekce identity a odposlechu aplikačních protokolů



Obrázek 4.1: Nasazení systému SLIS v síti ISP

4.1.1 Odposlech aplikačních protokolů

Praktické nasazení odposlechů v Internetu velmi často vyžaduje odposlech konkrétního aplikačního protokolu (např. e-mailové, či VoIP komunikace). Současná legislativa [24] však odposlech aplikačních protokolů definuje pouze na straně poskytovatelů služeb (např.

gmail.com). Tito poskytovatelé však velmi často působí mimo území České republiky a jejich spolupráce s orgány činnými v trestním řízení může být mnohdy problematická. Řešení tohoto problému je nastíněno na obrázku 4.1 - odposlouchávat aplikační protokoly přímo v síti národních poskytovatelů Internetového připojení (*Internet Service Provider* - ISP).

4.1.2 Detekce identity na základě aplikačního identifikátoru

Stávající implementace systému SLIS umožňovala realizovat odpolech, jehož cílem byla pouze IP adresa, identifikátor síťového rozhraní a několik dalších NID souvisejících s autentizací. V případě odposlechu aplikačních protokolů potřebujeme často detekovat identitu cíle právě na základě některého z aplikačních identifikátorů (např. e-mailové adresy, XMPP loginu, apod.). Tuto možnost systém až do současné doby neumožňoval. Až dosud bylo pro odposlech aplikačních protokolů nutné znát např. konkrétní IP adresu alespoň jedné z komunikujících stran. Odposlech pouze na základě e-mailové adresy, či IM identity (např. XMPP loginu) nebylo možné realizovat.

Až dosud podporoval systém SLIS možnost cílit odposlech pouze na některý z následujících NID (viz sekci 3.4):

- IPv4 adresa - Systém umožňuje specifikovat jak konkrétní IPv4 adresu, tak rozsah, např. 192.168.0.0/16.
- IPv6 adresa - Systém umožňuje specifikovat jak konkrétní IPv6 adresu, tak rozsah, např. fd02:1234:abcd::/48
- MAC adresa - Adresa síťového rozhraní
- DHCP client ID - Identifikátor, pomocí kterého DHCP server identifikuje klienta [3].
- DHCP DUID - Identifikátor klienta u DHCPv6 [4].
- RADIUS login - Přihlašovací jméno u bezpečnostního protokolu RADIUS [20].
- PPP login - Přihlašovací jméno u Point-to-Point protokolu [23].
- PPP session - Číslo sezení u Point-to-Point protokolu [23].

Jedním z cílů práce je přidat možnost detekce identity cíle na základě aplikačního identifikátoru (e-mailové adresy, XMPP loginu, IRC loginu, názvu IRC kanálu, apod.)

4.1.3 Selekce konkrétního TCP spojení

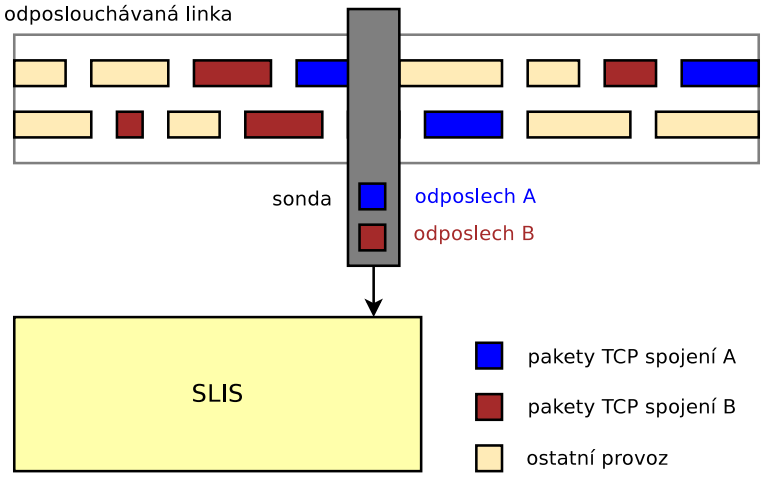
Další velkou nevýhodou byla také skutečnost, že až do současné doby byla odposlouchávána vždy veškerá komunikace dané IP adresy. Tato IP adresa mohla být zadána staticky jako cíl odposlechu, nebo byla nalezena jako NID_{CC} (identita uživatele) blokem IRI-IIF (viz kapitolu 3).

Nebylo možné odposlouchávat pouze vybranou část provozu. Pokud cílem zájmu byla např. pouze e-mailová komunikace, jediným řešením bylo pomocí systému SLIS zachytit

veškerou komunikaci cílove IP adresy a odfiltrování zájmových dat (např. e-mailové komunikace) realizovat až na straně LEA. Takové řešení představovalo jednak zbytečný nárůst objemu přenášených dat (jak na rozhraních HI2 a HI3, tak na vnitřním rozhraní INI2), jednak bylo nepohodlné pro LEA, protože bylo na straně LEA nutné provádět explicitní filtrování a selekci zájmových dat.

Dalším cílem práce je tento nedostatek odstranit. Chceme tedy, aby:

- Funkce dynamické identify (IRI-IIF) dokázala nalézt nejen konkrétní IP adresu, ale také identifikátor konkrétního TCP spojení v rámci nějž komunikace probíhá.
- Funkce odposlechu obsahu komunikace (sonda CC-IIF) byla schopna z odposlouchávané linky tato zájmová TCP spojení přímo vyfiltrovat - jak je znázorněno na obrázku 4.2.



Obrázek 4.2: Selekce konkrétních TCP spojení

4.1.4 Konfigurovatelné úrovně odposlechů

Dalším problémem byla skutečnost, že systém v rámci zadání požadavku na odposlech neumožňoval specifikovat, co vše má být do odposlechu zahrnuto a co již nikoli. V některých situacích (především s ohledem na různé typy soudních povolení) můžeme být žádoucí odposlouchávat pouze jednu konkrétní IP adresu. Někdy však chceme kompletní komunikaci probíhající skrz dané síťové rozhraní, i z více IP adres. To bylo dosud možné realizovat pouze v případě, že byl odposlech cílen na konkrétní MAC adresu. Zadání typu: „Uživatel má IP adresu xxx. Odposlouchávej tuto IP a všechny ostatní IP, které danému uživateli patří!“ nebylo možné řešit. Dalším požadavkem na rozšíření systému SLIS je tedy mít možnost definovat tzv. *úroveň odposlechu* (co odposlouchávat a co nikoli).

4.1.5 Celkový souhrn požadavků na rozšíření systému SLIS

Součástí této práce je navrhnout a implementovat rozšíření systému SLIS, které splňuje následující požadavky:

- Rozšíření bude umožňovat detekci identity uživatele na základě aplikačního identifikátoru (např. e-mailové adresy, či XMPP loginu).
- V případě odposlechu aplikačních protokolů bude Funkce dynamické identity (IRI-IIF) schopna detekovat nejen identitu v podobě IP adresy cíle, ale také v podobě identifikátoru konkrétního TCP spojení.
- Systém bude schopen na základě nalezeného identifikátoru TCP spojení provádět selekci paketů souvisejících právě s tímto spojením (vyžaduje implementaci podpory odposlechu aplikačních protokolů také do bloku MF&CCTF, sondy CC-IIF a rozhraní CCCI). Tímto se rozumí odposlech pouze vybrané části provozu daného uživatele (např. pouze konkrétní emailovou/IM komunikaci, telefonní hovor, apod.).
- V rámci každého požadavku na odposlech bude rozšíření umožňovat definovat tzv. *úroveň* odposlechu, která specifikuje, co vše bude do odposlechu zahrnuto a co nikoliv. Tato možnost souvisí s různými možnostmi soudních povolení umožňujících odposlouchávat pouze určité specifické cíle.
- Rozšíření bude umožňovat identifikovat zájmový provoz i v okamžicích, kdy ještě neznáme identitu uživatele (IP adresu).
- Rozšíření bude umožňovat zachytit zájmový provoz i v případě, kdy nelze identitu uživatele snadno zjistit (např. uživatel je v privátní síti, kde je provádět překlad adres NAT, v knihovně, internetové kavárně, apod.)

Realizace rozšíření systému SLIS o podporu odposlechu aplikačních protokolů včetně detekce identity na základě aplikačních identifikátorů je podrobně popsána v kapitole 5.

4.2 Optimalizace mediační funkce

Dle požadavků na prototyp systému pro sběr dat (viz přílohu B) je nutné, aby systém byl schopen zpracovávat data i z plně vytížené 10 Gbps linky bez ztráty paketů a navíc musí být schopen zpracovat souhrnně alespoň 500 Mbps filtrovaného provozu. V kapitole 3 bylo vysvětleno, že *funkce odposlechu obsahu komunikace* (blok CC-IIF) může být řešena i hardwarovou (např. vysoko-rychlostní) sondou. Rozhraní pro komunikaci se zbytkem systému je jednotné nezávisle na použité variantě sondy CC-IIF.

Mezi sondou CC-IIF a výstupním rozhraním systému (HI2 a HI3) se ovšem (dle popisu z kapitoly 3) nachází softwarový blok *Mediační a triggerovací funkce* (MF&CCTF). Tato skutečnost je znatelná také z obrázku 2.1. I pokud pomineme výkonnost sondy CC-IIF, coby omezující faktor, musí vždy blok MF&CCTF zpracovat každý dílčí paket zaslaný sondou (z rozhraní INI3) před jeho posláním na výstup (např. zapsáním do .pcap souboru). Je tedy zřejmé, že právě blok MF&CCTF je kritickým místem z hlediska výkonnosti systému.

V minulosti byla v rámci vývoje prototypu SLIS tato skutečnost několikrát teoreticky i experimentálně ověřena. V průběhu vývoje systému SLIS byla provedena řada optimalizací:

1. Místo původní metody `select.select` v Pythonu byla použita efektivnější varianta `select.epool`[1]

[1] http://blog.mongohq.com/blog/2013/02/19/behind-the-curtain-of-mongohq-a-gotcha-with-select-in-python/

2. Bylo zavedeno „bufferování“ dat v rámci socketu INI3.

3. Čtení z INI3 začalo být prováděno v samostatném vlákně `ini3Thread()` pro každou sondu CC-IIF. Tento krok přinesl další zrychlení při zpracování paketů. Kromě urychlení tato změna také eliminovala riziko možného vyhladovění volané funkce, které způsobovala tehdejší implementace `LISocketManager.mainLoop()` - tedy smyčky, jejíž cílem byla obsluha všeck socketů v rámci systému SLIS.

4. Některé části systému (Tabulka SIDů - viz kapitolu 3 a vlákno `ini3Thread()` spolu s obslužnými podfunkcemi) byly reimplementovány v jazyce Cython[2]. Při instalaci systému je pak kód v jazyce Cython automaticky převeden do jazyka C a následně zkompilován do statických binárních knihoven. Tato realizace přinesla další urychlení zprácovávávání paketů bloku MF&CCTF.

Ačkoli tyto kroky přinesly několikanásobné zrychlení zpracování paketů blokem MF&CCTF, současný stav stále není optimální. Součástí této práce je tedy další profilace a optimalizace bloku MF&CCTF. Tomuto tématu se věnuje kapitola 6.

[2] http://cython.org/

Kapitola 5

Rozšíření detekce identity a odposlechu aplikačních protokolů

V této kapitole je popsán návrh rozšíření systému SLIS o podporu aplikačních protokolů a konfigurovatelné úrovně odposlechu. V kapitole 4 byly podrobně popsány cíle tohoto rozšíření. Tato kapitola se věnuje jeho samotné realizaci.

5.1 Popis rozšíření systému

5.1.1 Nově přidané funkce

Nové rozšíření nabízí možnost do systému SLIS kdykoli integrovat podporu libovolného protokolu aplikační vrstvy. S tímto protokolem mohou být přidány nové identifikátory NID, na které je možné cílit odposlech. Tato možnost vyplývá z modulární architektury celého systému. Nový protokol znamená přidání nového modulu do bloku IRI-IIF (viz sekci 3.6).

Pro tyto účely bylo nutné přepracovat celý blok IRI-IIF. Novou podobu tohoto bloku představuje *jádro IRI-IIF založené na grafové reprezentaci*. Její návrh bude podrobněji popsán v sekci 5.2.

Abychom mohli specifikovat konkrétní aplikační protokol a nebylo nutné odposlouchávat veškerou komunikaci cíle s danou IP, je potřeba mít možnost určit sondě CC-IIF konkrétní port, či typ transportního protokolu [14]. Je tedy nutné, aby sonda CC-IIF byla schopna odposlouchávat i jiné NID_{CC} než pouze IPv4/6 adresu. Za tímto účelem byly přidány dva nové NID_{CC}:

- Pro označení komunikace dvou stran uvažujeme pětici:
  - IPv4/6 adresa klienta,
  - IPv4/6 adresa serveru,
  - port klienta,
  - port serveru,
  - typ transportního protokolu (např. TCP).
- Pro označení jedné komunikující strany uvažujeme trojici:
  - IPv4/6 adresa,
  - port,

– typ transportního protokolu (např. TCP).

První aplikační protokoly, jejichž podpora byla do systému integrována, jsou protokoly pro komunikaci typu Instant Messaging (IM): IRC [17], XMPP [21], YMSG[1] a OSCAR[2]. Podpora těchto protokolů byla doplněna do jádra IRI-IIF a její řešení bude dále popsáno v sekci 5.2. Samotný IRI-IIF modul pro zpracování uvedených IM protokolů byl implementován v rámci projektu Sec6Net nezávisle na této práci. Pro podporu IM protokolů byly zavedeny následující nové NID:

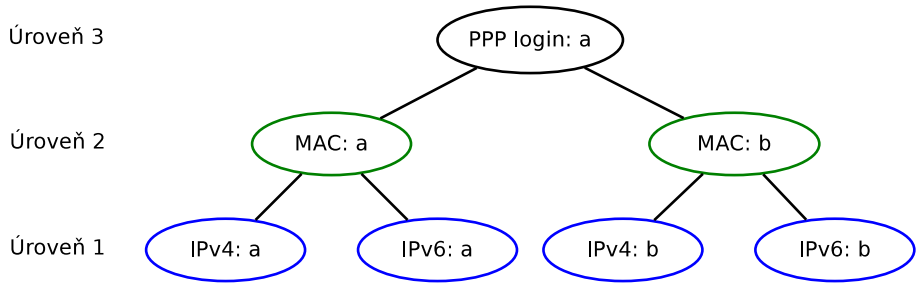
- IRC login - Identifikátor uživatele v rámci protokolu IRC,
- IRC channel - Název komunikačního kanálu na serveru protokolu IRC,
- XMPP login - Identifikátor uživatele v rámci protokolu XMPP,
- YMSG login - Identifikátor uživatele v rámci protokolu YMSG,
- OSCAR login - Identifikátor uživatele v rámci protokolu OSCAR.

Provedené úpravy systému SLIS umožňují bez výraznějších změn stávajícího systému přidat podporu pro libovolný aplikační protokol. Do budoucna se uvažuje podpora odposlechu e-mailové komunikace a IP telefonie.

Dalším rozšířením je podpora konfigurovatelných úrovní odposlechu. Pro ilustraci uvažujme obrázek 5.1.1. Uživatel s PPP loginem a [23] vlastní dva počítače, případně jeden počítač s dvěma síťovými rozhraními. Adresy síťových rozhraní jsou MAC a a MAC b. Rozhraní s MAC: a má dvě IP adresy: IPv4 a, IPv6 a. Rozhraní identifikované MAC b má dvě IP adresy: IPv4 b, IPv6 b. Při zadávání požadavku na odposlech můžeme zvolit jednu z následujících tří úrovní:

1. Odposlech v rozsahu síťové adresy - Předmětem zájmu je komunkace spojená pouze s konkrétní IP adresou (např. IPv4 a. Přestože lze z grafu dohledat informaci, že uživatel používá více IP adres, budou se zachytávat pouze pakety s IP adresou IPv4 a.
2. Odposlech v rozsahu rozhraní nebo počítače - Předmětem zájmu je veškerá komunikace v rámci jednoho síťového rozhraní nebo počítače. Cílem odposlechu může být NID představující síťovou adresu (např. IPv4 a) nebo adresu rozhraní (např. MAC a). Blok jádro IRI-IIF v grafu dohledá všechny IP adresy související se stejným rozhraním (tj. adresy IPv4 a a IPv6 a). Přes úroveň rozhraní však nezasahuje (tj. adresy IPv4 b a IPv6 b nejsou předmětem odposlechu).
3. Odposlech v rozsahu uživatele - Předmětem zájmu je veškerá komunikace daného uživatele. Cílem odposlechu může být NID představující síťovou adresu (např. IPv4 a) nebo adresu rozhraní (např. MAC a) nebo jiný identifikátor (např. PPP login p) [23]. Blok IRI jádro je schopen v grafu dohledat všechny IP adresy související se stejným uživatelem (tj. adresy IPv4 a a IPv6 a, ale i IPv4 b a IPv6 b).

[1] http://messenger.yahoo.com/
[2] http://iserverd1.khstu.ru/oscar



Obrázek 5.1: Znázornění jednotlivých úrovní odposlechu

5.1.2 Souhrn všech uvažovaných úprav

- Kompletně nová implementace jádra IRI-IIF - *Jádro IRI-IIF založené na grafové reprezentaci.*
- V bloku MF&CCTF přepracováno mapování LIID na SID, aby uvažovalo aplikačních NID_{CC}.
- Přidání podpory aplikačních NID_{CC} do softwarové implementace sondy CC-IIF a rozhraní CCCI.
- Optimalizace bloku MF&CCTF s ohledem na rychlost.

5.2 Jádro IRI-IIF založené na grafové reprezentaci

5.2.1 Vysvětlení konceptu nového jádra IRI-IIF

Pro podporu aplikačních protokolů a především konfigurovatelných úrovní odposlechu byla provedena rozsáhlá úprava bloku IRI-IIF. Tato úprava zahrnuje také novou implementaci jádra IRI-IIF - tzv. *grafovou IRI-IIF*. Základní funkcionalita odpovídá popisu ze sekce 3.6. Pro reprezantaci stavu sítě však byl zvolen zcela jiný model a také dohledávání konkrétních NID_{CC} ke vstupnímu NID je řešeno jiným způsobem.

Základním modelem odposlouchávané sítě je graf. Uzly grafu jsou identifikátory NID. Hrany představují asociaci dvou NID v rámci nějakého protokolu.

5.2.2 Typy NID

Veškeré podporované NID jsou nově rozděleny do čtyř kategorií představujících typ NIDu z hlediska jeho významu v síti. Uvažujeme čtyři typy:

- **typ A** - Aplikační identifikátor (viz sekci 5.1)
  - Identifikátor v rámci IM protokolu: IRC login, číslo IRC kanálu, XMPP login, YMSG login, OSCAR login
  - 5-tice (IP klienta, IP serveru, port klienta, port serveru, typ transportního protokolu)
  - 3-jice (IP, port, typ transportního protokolu)
  - V budoucnu libovolný další aplikační identifikátor (např. e-mail, apod.)

- **typ B** - Adresa síťové vrstvy
  - IPv4 adresa
  - IPv6 adresa
- **typ C** - Adresa síťového rozhraní, nebo identifikátor konkrétního počítače
  - MAC adresa
  - DHCP client ID [3]
  - DHCPv6 DUID [4]
- **typ D** - Ostatní identifikátory (především pro autentizaci)
  - RADIUS login [20]
  - PPP login [23]
  - Číslo PPP sezení [23]

5.2.3 Princip vyhledávání v grafu

Cílem bloku IRI-IIF je ke vstupnímu NID, který je součástí zadání odposlechu, nalézt všechny související NID. Ze souvisejících NID jsou podstatné především NID_{CC}, které se používají ke konfiguraci sondy CC-IIF (viz kapitolu 3). Ostatní související NID mají pouze informativní hodnotu a jsou posílány na výstup jako metadata skrz rozhraní HI2 [7].

Způsob nalezení souvisejících NID představuje prohledávání grafu. Množina obsahující nalezené NID nejprve obsahuje pouze vstupní NID - výchozí uzel grafu. Postupně procházíme hrany grafu, dokud nalézáme nové uzly.

Je ovšem třeba také zohlednit jednotlivé úrovně odposlechu. Které uzly budou použity je určeno pravidly uvedenými v tabulce 5.1. První sloupec představuje typ vstupního NID (dle kategorií uvedených v podsekci 5.2.2). Pro každý typ jsou v druhém sloupci uvedeny odpovídající uzly, které budou do výsledné množiny zahrnuty. Tabulka je rozdělena do tří částí podle použité úrovně odposlechu.

U některých pravidel z tabulky 5.1 existuje výjimka, kdy neprocházíme uzly „typu B připojené k uzlům typu A jako server“. Toto souvisí s nově přidaným NID_{CC}, který představuje 5-tici (IP klienta, IP serveru, port klienta, port serveru, typ transportního protokolu) představenou v sekci 5.1. Tato pětice identifikuje komunikaci klienta a serveru. Na serveru s danou IP běží na daném (např. TCP) portu určitá služba. K této službě se připojuje klient s jinou IP a jiným klientským portem. Hledání identity (souvisejících NID) se vždy (včetně odposlechů 3. úrovně) týká pouze jednoho uživatele. 5-tice představuje identifikátor typu A (aplikační). V praxi k tomuto uzlu budou připojeny dva uzly typu B představující IP adresy klienta a serveru. Uvažme situaci, kdy prohledávání grafu dojde skrz IP klienta až k uzlu představujícímu zmíněnou 5-tici. Pokud bychom dále zahrnuli i IP serveru, získali bychom identifikátor, který sledovanému uživateli nepatří. Z tohoto důvodu jsou při procházení grafu serverové IP ignorovány.

5.2.4 Ukázka vyhledávání v grafu

Ukázkový příklad prohledávání grafu uvažuje situaci uvedenou na obrázku 5.2. Máme zde uživatele, identifikaného pomocí PPP loginu: *ferda*. Pod tímto loginem uživateli patří dvě

Úroveň 1	
typ	uzly patřící k danému odposlechu
A	uzel odpovídající identifikátoru typu A a přímo propojené uzly typu A
B	uzel odpovídající identifikátoru typu B a přímo propojené uzly typu A
C	uzel odpovídající identifikátoru typu C, přímo propojené uzly typu B a k nim přímo propojené uzly typu A
D	uzel odpovídající identifikátoru typu D, přímo propojený uzel typu C, k nim přímo propojené uzly typu B a k nim přímo propojené uzly typu A
Úroveň 2	
typ	uzly patřící k danému odposlechu
A	všechny uzly přístupné cestou přes uzly typu A, B, C, s výjimkou uzlů typu B připojených na uzly typu A jako server
B	všechny uzly přístupné cestou přes uzly typu B, C, přímo propojené uzly typu A
C	všechny uzly přístupné cestou přes uzly typu B, C, přímo propojené uzly typu A
D	uzel odpovídající identifikátoru typu D, všechny k němu přímo propojené uzly typů C, všechny uzly přístupné cestou přes uzly typu B, C, přímo propojené uzly typu A
Úroveň 3	
typ	uzly patřící k danému odposlechu
A	všechny uzly přístupné cestou přes uzly typu A, B, C, D, s výjimkou uzlů typu B připojených na uzly typu A jako server
B	všechny uzly přístupné cestou přes uzly typu B, C, D, přímo propojené uzly typu A
C	všechny uzly přístupné cestou přes uzly typu B, C, D, přímo propojené uzly typu A
D	všechny uzly přístupné cestou přes uzly typu B, C, D, přímo propojené uzly typu A

Tabulka 5.1: Pravidla pro prohledávání grafu

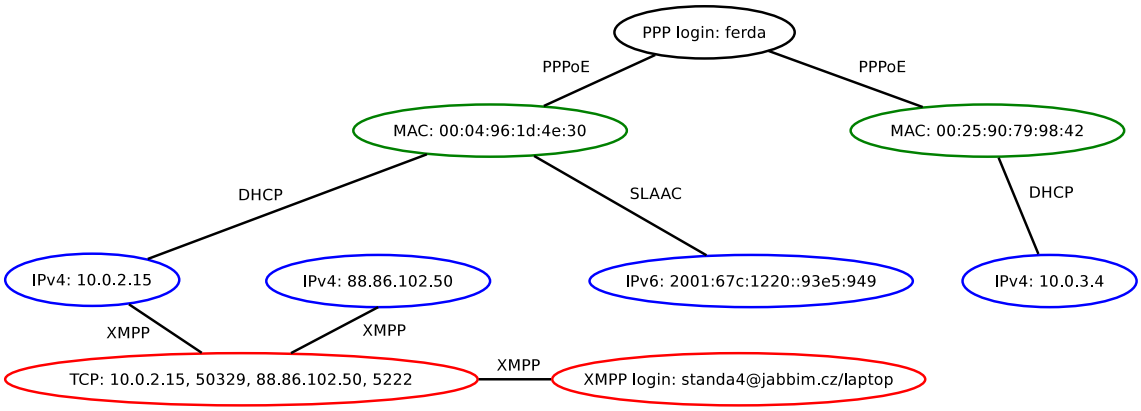
síťová rozhraní s MAC adresami *00:04:96:1d:4e:30* a *00:25:90:79:98:12*. Prvnímu rozhraní je přiřazena IPv4 adresa *10.0.2.15* a IPv6 adresa *2001:67c:1220::93e5:949*. Druhému rozhraní je přiřazena IPv4 adresa *10.0.3.4*.

Uživatel komunikuje pomocí protokolu XMPP [21] přes vzdálený server s IP *88.86.102.50*. Tato komunikace probíhá v rámci TCP spojení uvedeného na obrázku. Uživatel v rámci dané komunikace požuívá XMPP login: *standa4@jabbim.cz/laptop*.

Uvažujme tři různé odposlechy:

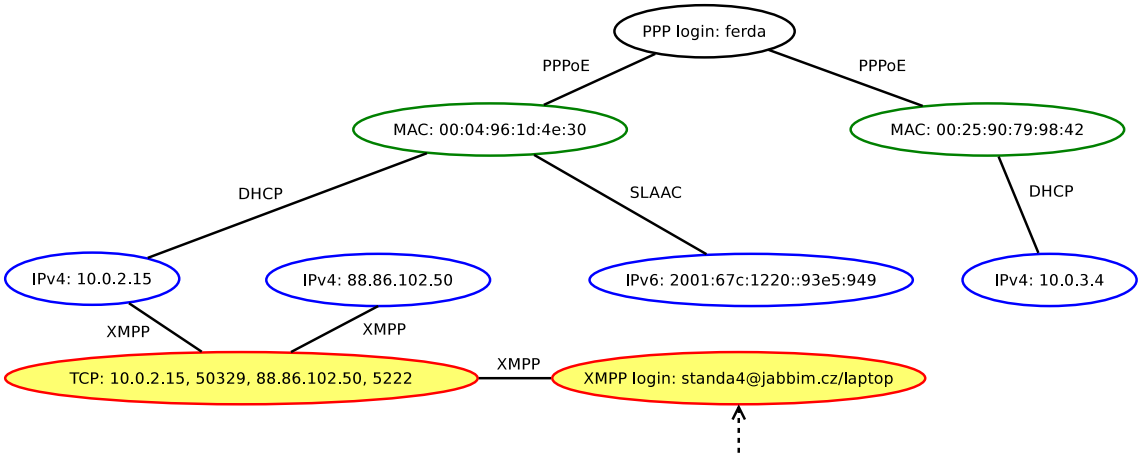
- **Odposlech X** - Cílem je NID představující XMPP login: *standa4@jabbim.cz/laptop*. Úroveň odposlechu je 1.
- **Odposlech Y** - Cílem je také NID představující XMPP login: *standa4@jabbim.cz/laptop*. Úroveň odposlechu je 2.
- **Odposlech Z** - Cílem je také NID představující XMPP login: *standa4@jabbim.cz/laptop*. Úroveň odposlechu je 3.

Odposlech X má nastavenou **úroveň 1**. Dle pravidel z tabulky 5.1 bude do výsledné množiny uzlů zahrnut pouze samotný vstupní uzel (NID) typu A a přímo propojené uzly typu A. Výsledek hledání je vyobrazen na obrázku 5.3. Uzly patřící do nalezené množiny



Obrázek 5.2: Ukázkový příklad grafu v IRI-IIF

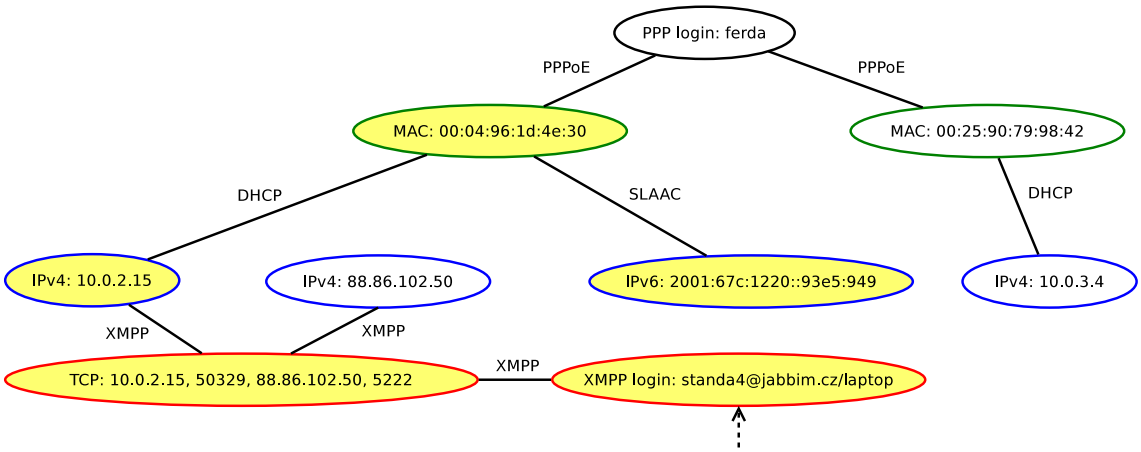
jsou zvýrazněny žlutě. NID související s odposlechem jsou tedy XMPP login a TCP spojení v rámci něhož XMPP komunikace probíhá.



Obrázek 5.3: Odposlech 1. úrovně cílený na XMPP login

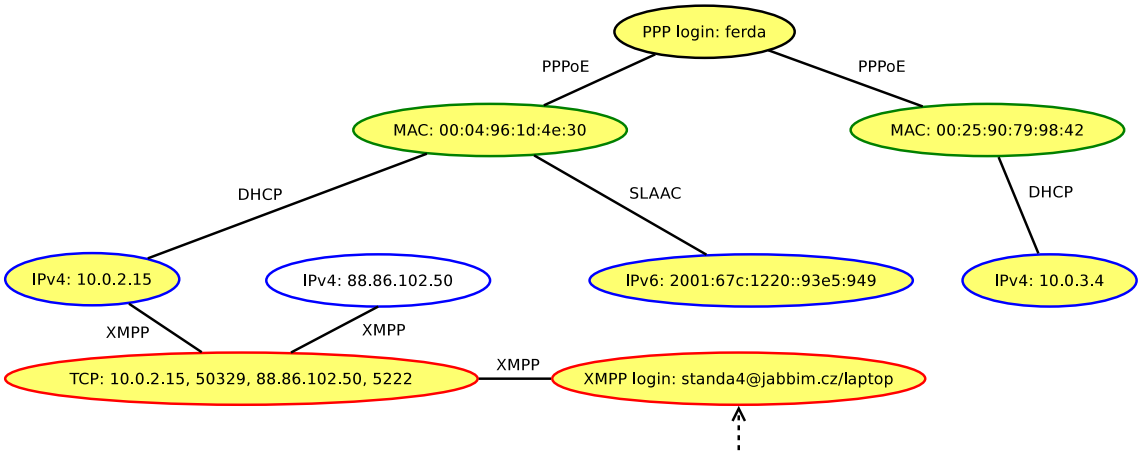
Odposlech Y má nastavenou **úroveň 2**. Na této úrovni patří do oblasi zájmu veškerá komunikace probíhající přes dané síťové rozhraní. Dle pravidel z tabulky 5.1 budou do výsledné množiny uzlů zahrnuty všechny uzly přístupné cestou přes uzly typu A, B, C, s výjimkou uzlů typu B připojených na uzly typu A jako server. Výsledek hledání je vyobrazen na obrázku 5.4. K nalezeným uzlům tedy přibyla MAC adresa *00:04:96:1d:4e:30* daného rozhraní a obě IP adresy, které jsou tomuto rozhraní přiřazeny.

Odposlech Z má nastavenou **úroveň 3**. Na této úrovni se snažíme pokrýt veškerou komunikacei daného uživatele. Dle pravidel z tabulky 5.1 budou do výsledné množiny uzlů zahrnuty všechny uzly přístupné cestou přes uzly typu A, B, C, D, s výjimkou uzlů typu



Obrázek 5.4: Odposlech 2. úrovně cílený na XMPP login

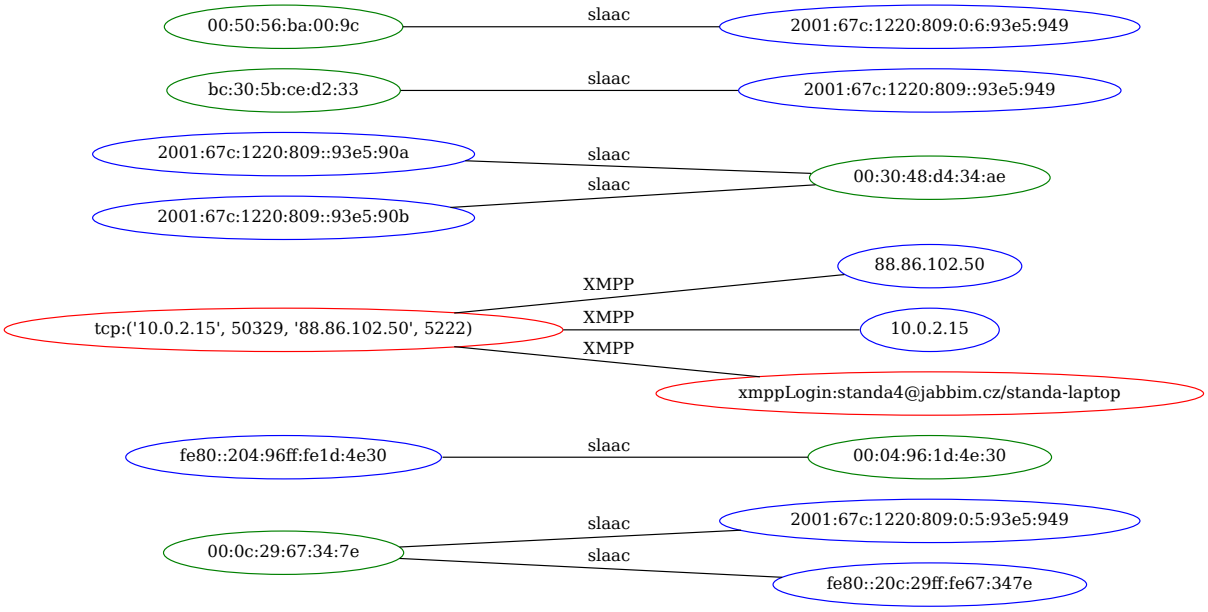
B připojených na uzly typu A jako server. Výsledek hledání je vyobrazen na obrázku 5.5. K nalezeným uzlům tedy přibyla MAC adresa *00:25:90:79:98:12* druhého rozhraní patřícího danému uživateli a také IPv4 adresa, která je tomuto rozhraní přiřazena.



Obrázek 5.5: Odposlech 3. úrovně cílený na XMPP login

5.2.5 Zobrazení grafu skrz webové rozhraní

Skrz webové rozhraní systému SLIS je možné celý graf vizualizovat a poskytnout tak operátorovi oblushujícímu systém lepší představu o aktuálním stavu sítě. Ukázka reálného grafu, která byla získána z webového rozhraní systému, je na obrázku 5.6.



Obrázek 5.6: Ukázka reálného grafu NID zobrazitelného skrz webové rozhraní

Obrázek byl získán při testování nové implementace jádra IRI-IIF s modulem pro XMPP a ukazuje, že graf reprezentující stav sítě nemusí být spojitý. V našem případě uvažujeme, že graf by měl mít tolik komponent, kolik je v síti známých uživatelů.

V obrázku vidíme přidělení IPv6 adres bezstavovou konfigurací SLAAC. Např. síťovému rozhraní s MAC adresou *00:30:48:d4:34:ae* byly přiděleny dvě různé IPv6 adresy: *2001:67c:1220:809::93e5:90a* a *2001:67c:1220:809::93e5:90b*.

V prostřední části vidíme komunikaci pomocí protokolu XMPP [21]. Červený uzel v levé části představuje zmíněnonu 5-tici, kde protokolem transportní vrstvy je TCP. V grafu vidíme asociace TCP spojení s uzly představujícími IP klienta a serveru. Z obrázku je také zřejmé, že pro komunikaci uživatel používal XMPP login: *standa4@jabbim.cz/standa-laptop*.

5.3 Podpora aplikačních protokolů v MF&CCTF

5.3.1 Problém dosavadního řešení při zavedení aplikačních identifikátorů

Jedním z hlavních cílů bloku MF&CCTF je mapování LIID na SID a následná konfigurace sondy CC-IIF pro jednotlivé SID. V příloze D je popsán dosavadní algoritmus a oobecné řešení tohoto mapování pro rozsahy IP adres. Uvedený postup počítal, že pro všechny platné rozsahy adres A, B platí $A \cap B = \emptyset \vee A \subseteq B \vee B \subseteq A$.

Se zavedením nových NID_{CC} v podpově aplikačních identifikátorů (3-jic a 5-tic) však postup uvedený v příloze D obecně přestává platit. Pro vysvětlení uvažujme, že A a B jsou libovolné existující NID. U 3-jic a 5-tic uvažujme pro jednoduchost pouze transportní protokol TCP.

Zaveďme nyní binární relaci *in*. Tato relace je použita také v reálné implementaci systému SLIS a slouží k porovnávání dvou NID. Relace *in* je reflexivní a tranzitivní. Relace *in* není symetrická. A *in* B platí, pokud je splněna právě jedna z následujících situací:

- A i B jsou totožné NID (symetrie).
- A představuje IP adresu. B představuje rozsah IP adres. A patří do rozsahu B.
- A představuje rozsah IP adres. B představuje rozsah IP adres. A je podrozsahem B.
- A představuje 3-jici (IP_X, port, TCP) a platí IP_X *in* B.
- A představuje 5-tici (IP_X, port_X, IP_Y, port_Y, TCP) a platí právě jedna z následujících situací:
  - IP_X *in* B
  - IP_Y *in* B
  - B je 3-jice (IP_Z, port_Z, TCP) a platí: (IP_X *in* $\text{IP}_Z \wedge \text{port}_X = \text{port}_Z$) $\vee$ (IP_Y *in* $\text{IP}_Z \wedge \text{port}_Y = \text{port}_Z$)

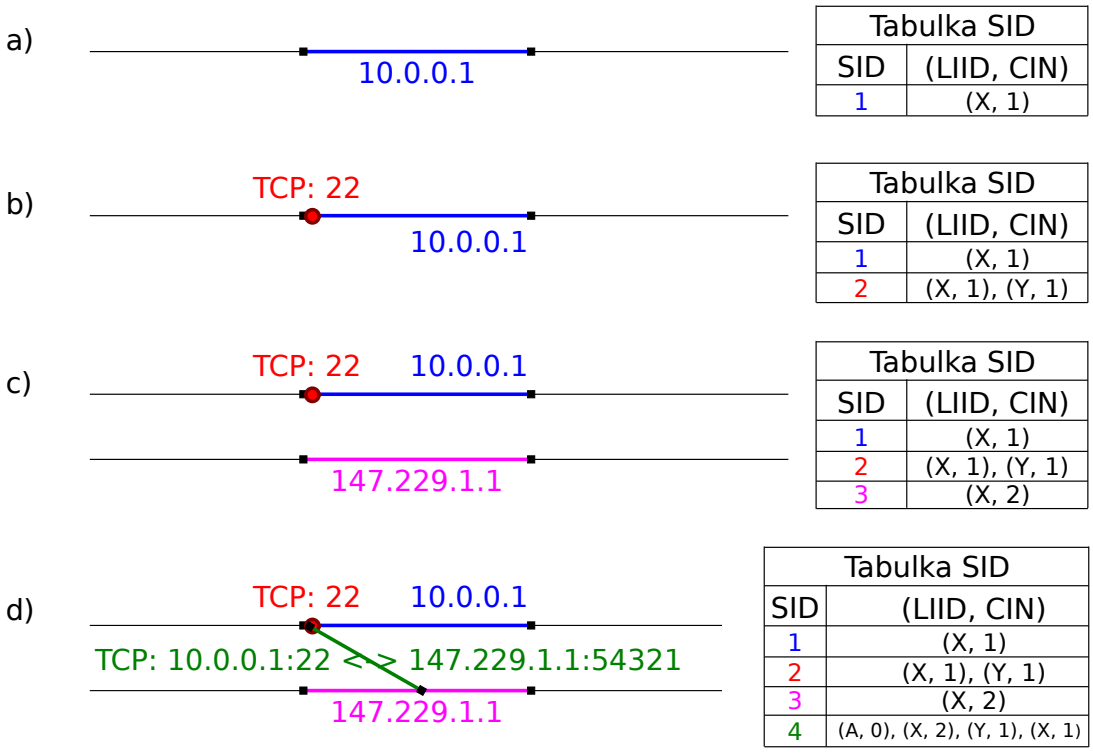
Dokud jsme neuvažovali 3-jice a 5-tice, platilo, že pokud A *in* B a zároveň A *in* C, musí platit buď B *in* C nebo C *in* B. Pokud je ale A např. 5-tice představující TCP spojení a B a C např. IP adresy, výše uvedený výrok platit nemusí.

5.3.2 Nový algoritmus pro mapování LIID na SID

Koncept přidělování SID byl rozšířen pro SID pro podporu 3-jic a 5-tic. Vysvětleme nyní nově zavedený algoritmus přidělování SID. Blok IRI-IIF oznámí nový odposlech s LIID_N cílený na NID_N. Pro NIDCC_N v rámci s LIID_Nodposlechu přiřadil blok IRI-IIF CIN_N. Postupujeme následovně:

- Pokud neexistuje žádný odposlech cílený na NID, kde NID *in* NID_N nebo NID_N *in* NID, pak je vygenerován nový SID: SID_N. K SID_N se v tabulce přiřadí dvojice (LIID_N, CIN_N).
- Pokud existuje nějaký odposlech cílený na NID_V, takový, že NID_N *in* NID_V, přičemž NID_N a NID_V nejsou totožné, pak je vygenerován nový SID: SID_N. K SID_N se v tabulce přiřadí dvojice (LIID_N, CIN_N). K SID_N se ovšem také přiřadí všechny dvojice, které představují odposlech cílený na NID_V takový, že NID_N *in* NID_V.
- Pokud již existuje odposlech cílený na NID, který je totožný s NID_N, pak nevytváříme nový SID. Dvojice (LIID_N, CIN_N) je přidána ke všem SID, ke kterým patří odposlechy cílené na NID_M takové, že NID_M in NID_N.
- Pokud existuje odposlech na NID_M takový, že NID_M *in* NID_N a zároveň NID_M a NID_N nejsou totožné, pak je vygenerován nový SID: SID_N. K SID_N se v tabulce přiřadí dvojice (LIID_N, CIN_N). Dvojice (LIID_N, CIN_N) se však přidá také ke všem SID, ke kterým patří odposlechy cílené NID_M takové, že NID_M in NID_N.

Uvedený postup nemusí být na první pohled příliš jasný, proto jej demonstrujme na příkladu. Ukázkový postup je znázorněn na obrázku 5.7. Uvažujeme následující kroky:



Obrázek 5.7: Ukázka mapování LIID na SID pro rozsahy aplikační protokoly

1. Byl přidán odposlech s LIID X, jehož cílem je adresa *10.0.0.1*. Uvažujme, že blok IRI-IIF této adrese v rámci LIID X přidělil CIN 1. MF&CCTF v *tabulce SID* vytvoří nový záznam: SID 1 a do tabulky uloží, že data označená SID 1 se budou vztahovat k LIID X (pro CIN 1), jak je znázorněno v části a) obrázku 5.7.

2. Poté byl přidán odposlech s LIID Y, jehož cílem je 3-jice (*10.0.0.1*, 22, TCP). Uvažujme, že blok IRI-IIF této 3-jici v rámci LIID Y přidělil CIN 1. Blok MF & CCTF tedy vytvoří nový SID 2. Do *tabulky SID* uloží, že data označená SID 2 se budou vztahovat k odposlechu s LIID X (pro CIN 1) a také k odposlechu s LIID Y (pro CIN 1). Situaci znázorňuje část b) obrázku 5.7.

3. Blok IRI-IIF detekoval, že k odposlechu s LIID X nově patří také NID_{CC} v podpobě adresy *147.229.1.1*. Uvažujme, že blok IRI-IIF této adrese v rámci LIID X přidělil CIN 2. Blok MF & CCTF tedy vytvoří nový SID 3. Do /*tabulky SID* uloží, že data označená SID 3 se budou vztahovat k odposlechu s LIID X (pro CIN 2). Protože neexistuje žádný NID, který by byl v relaci *in* s nově přidanou adresou *147.229.1.1*, nebude provedena žádná další činnost. Situaci znázorňuje část c) obrázku 5.7.

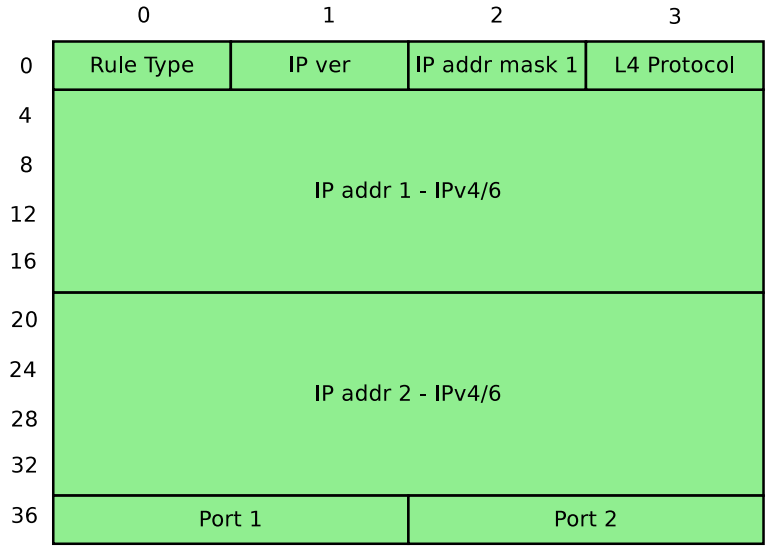
4. Nakonec byl přidán nový odposlech s LIID A, jehož cílem je 5-tice (*10.0.0.1*, 22, *147.229.1.1*, 54321, TCP). Uvažujme, že blok IRI-IIF této 5-tici v rámci LIID A přidělil CIN 0. Blok MF&CCTF v *tabulce SID* vytvoří nový záznam: SID 4 a do tabulky uloží, že data označená SID 4 se budou vztahovat k LIID A (pro CIN 0). Zároveň však blok MF&CCTF přidá k SID 4 všechny odposlechy cílené na takové NID, pro které platí (*10.0.0.1*, 22, *147.229.1.1*, 54321, TCP) *in* NID. Do *tabulky SID* tedy blok MF&CCTF uloží informaci, že data označená SID 4 se budou vztahovat také k LIID X

(pro CIN 1), k LIID X (pro CIN 2) a k LIID Y (pro CIN 1). Situaci znázorňuje část d) obrázku 5.7.

5.4 Podpora aplikačních protokolů na rozhraní CCCI a v sondě CC-IIF

5.4.1 Podpora na rozhraní CCCI

Dalším krokem k úspěšně realizaci odposlechu aplikačních protokolů byla úprava rozhraní CCCI. Rozhranní CCCI slouží ke komunikaci mezi sondou CC-IIF a blokem MF&CCTF. Vzájemná komunikace funguje na principu požadavek-odpověď. MF&CCTF pošle sondě požadavek jehož součástí je kromě akce (přidání/odebrání) odposlechu a identifikátorů RID a SID také pravidlo (rule). Pravidlo má pevně definovanou strukturu, která je ukázána na obrázku 5.8.



Obrázek 5.8: Struktura požadavku na rozhraní CCCI

Význam jednotlivých polí je následující:

- **Rule Type** (1 B) může nabývat hodnot 0 až 3:
  - 0 - Součástí požadavku je konkrétní IP adresa a maska
  - 1 - Odposlech je cílen na konkrétní komunikaci (spojení)
  - 2 - Přistupujeme k pevně zadanému portu pro všechny IP adresy
  - 3 - Přistupujeme k pevně zadanému portu pro konkrétní IP adresu
- **IP ver** (1 B) představuje verzi protokolu IP (4 - IPv4, 6 - IPv6). Položka je platná pro pravidla typu 0, 1, 3.
- **L4 Protocol** (1 B) představuje číslo transportního protokolu. Položka je platná pro pravidla typu 1, 2, 3.

- **IP addr mask 1** (1 B) představuje masku IP adresy 1 v rozsahu 0 - 128. Položka je platná pro pravidla typu 0. Pro pravidla typu 1 a 3 musí být nastavena na hodnotu 32 pro IPv4 a 128 pro IPv6
- **IP addr 1** (16 B) představuje IPv4 nebo IPv6 adresu (dle IP ver). Položka je platná v pravidlech typu 0, 1, 3.
- **IP addr 2** (16 B) představuje IPv4 nebo IPv6 adresu (dle IP ver). Položka je platná pro pravidla typu 1.
- **Port 1** (2 B) představuje číslo transportního protokolu. Položka je platná pro pravidla typu 1, 2, 3.
- **Port 2** (2 B) představuje číslo transportního protokolu. Položka je platní pro pravidla typu 1.

Při filtrování se uvažuje vždy obousměrná komunikace. Tzn. buď nastane situace, kdy se IP addr 1, IP addr 2, Port 1, Port 2 porovnává se zdrojovu, cílovou IP adresou, zdrojovým a cílovým portem, nebo se IP addr 1, IP addr 2, Port 1, Port 2 porovnává s cílovou, zdrojovou IP adresou a cílovým a zdrojovým portem.

Až do současnosti systém podporoval pouze pravidla typu 0 - tzn. odposlech konkrétní IP adresy a masky. (Využívala se pouze pole Rule Type, IP ver, IP addr maks 1 a IP addr 1). S rozšířením systému o podporu aplikačních protokolů bylo nutné doplnit do implementace rozhraní CCCI podporu pro korektní zpracování druhé IPv4/6 adresy a také portů (např. pro protokol TCP). Vzhledem ke skutečnosti, že rozhraní CCCI (narozdíl od vnitřních rozhraní systému) využívá komunikaci skrz protokol TCP, bylo nutné zajistit správný převod adres z vnitřní reprezentace v Pythonu do binární podoby k přenosu po síti. Podobným způsobem bylo třeba zajistit správný převod jednotlivých položek z přijatého paketu zpět do interní podoby využívané systémem SLIS.

5.4.2 Podpora v sondě CC-IIF

Společně s doplněním potřebné funkcionality do rozhraní CCCI bylo nutné implementovat podporu odposlechu aplikačních protokolů také do softwarové verze sondy CC-IIF.

Soda CC-IIF si udržuje vnitřní tabulku pravidel (`cc_table`). S každým přijatým paketem probíhá analýza tohoto paketu. Součástí analýzy je ověření, zda paket neopovídá některému z evidovaných pravidel. Pokud ano - přepošle jej sonda skrz rozhraní INI3 bloku MF&CCTF [18].

Až dosud byla ověřována pouze IP adresa přijatého paketu a nebylo tedy nutné řešit, zda paket obsahuje uvnitř zapouzdřen TCP segment a případně jej dále analyzovat.

Pro plnou podporu odposlechu aplikačních protokolů (možnosti selekce kontrétního TCP spojení) bylo nutné doplnit implementaci o ověřování obou IP adres a především portů transportní vrstvy. Tato úprava vyžadovala jak zásah do implementace samotné sondy CC-IIF, tak úpravu funkce `findSID()`, která je součástí objektu představujícího tabulku `cc_table`.

Kapitola 6

Optimalizace mediační funkce

Tato kapitola je zaměřena na analýzu výkonnosti bloku MF&CCTF, identifikaci kritických míst a jejich optimalizaci. Sekce 6.1 popisuje jednak vnitřní strukturu bloku MF&CCTF, jednak profilaci výkonnosti a identifikaci kritických míst. V sekci je popsána navržená a implementovaná metoda optimalizace tohoto bloku.

6.1 Profilace mediační funkce a identifikace kritických míst

6.1.1 Vnitřní implementace mediační funkce

Aby bylo možno se zaměřit na identifikaci kritických míst, bude nutné nejprve popsat vnitřní strukturu bloku MF&CCTF a její vazby na sondu CC-IIF a výstupní rozhraní systému. V kapitole 3 bylo vysvětleno, že mezi blokem CC-IIF a MF&CCTF existují dvě rozhraní:

- CCCI - pro zadávání požadavků na odposlech (CC-IIF ← CCTF),
- INI3 - pro předávání zachycených dat (CC-IIF → MF).

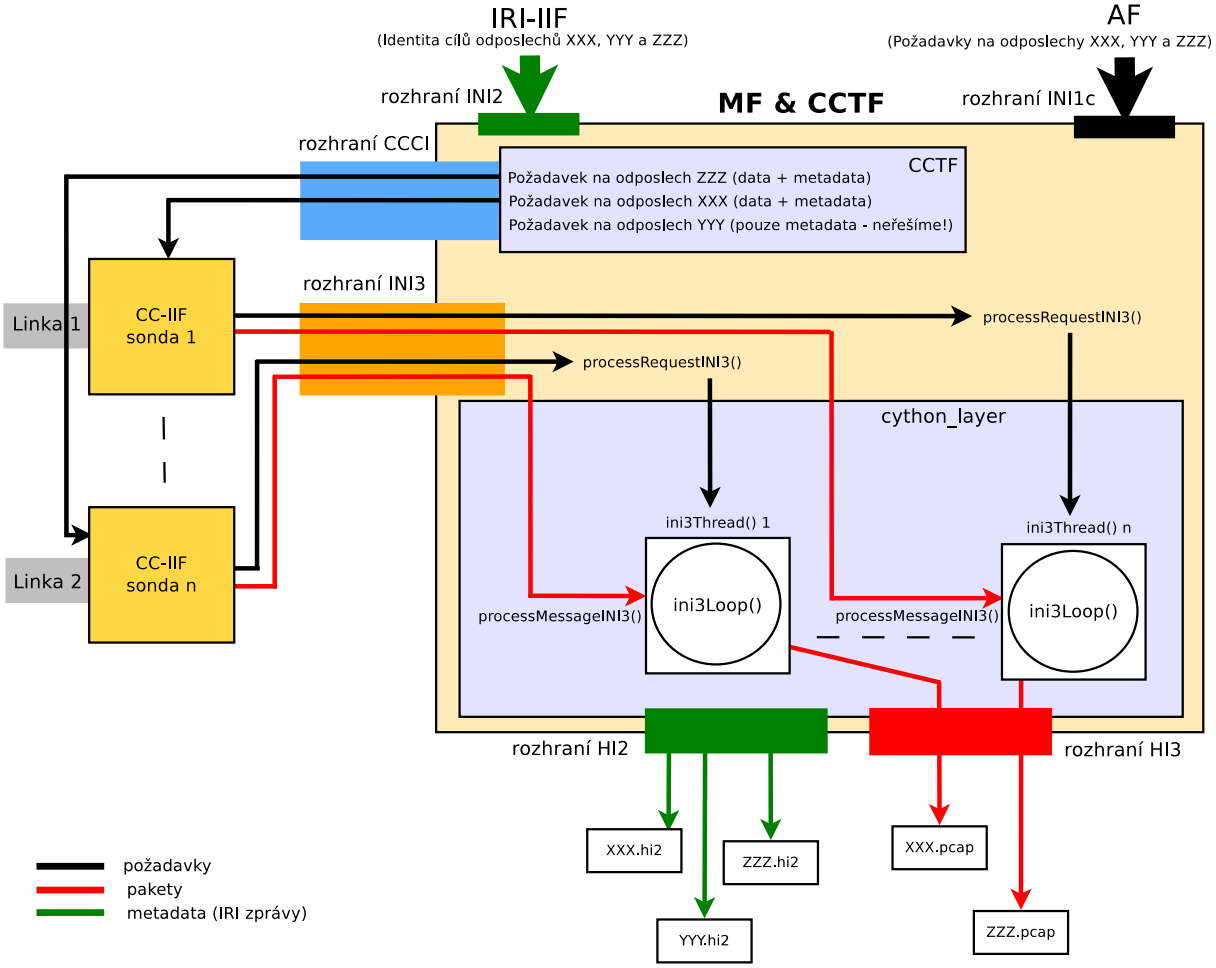
Rozdělení na CCTF a MF zde bylo uvedeno záměrně, neboť CCTF je část bloku MF&CCTF, jejímž účelem je spouštění odposlechů (konfigurace sondy CC-IIF), zatímco účelem MF je především zpracování přijatých paketů. Sonda CC-IIF může existovat buď jako softwarová část centrálního zařízení SLIS (viz kapitolu 3), tak jak v hardwarové podobě jako samostatné zařízení. Aktivních sond může být k bloku MF&CCTF připojeno i více, přičemž pro všechny sondy uvažujeme jednotné rozhraní [18]. Blok MF&CCTF je dále připojen na výstupní rozhraní celého systému:

- HI2 - pro předání metadat (IRI zpráv) na výstup systému,
- HI3 - pro předání obsahu zachycené komunikace (paketů) na výstup systému.

V současné době jsou tato rozhraní implementačně řešena zápisem do souborů. V případě rozhraní HI2 se vytvoří soubor `[LIID].hi2`, do kterého jsou v textové podobě ukládany IRI zprávy související s daným odposlechem. V případě rozhraní HI3 se vytvoří soubor `[LIID].pcap` ve formátu PCAP[1], do kterého jsou ukládány zachycené pakety související s daným odposlechem. `[LIID]` je identifikátor konkrétního odposlechu [18].

[1] http://www.tcpdump.org/pcap.html

V kapitole 4 bylo vysvětleno, že v rámci urychlení zpracování paketů je obsluha rozhraní INI3 řešena samostatným vláknem `ini3Thread()` pro každé spojení se sondou. Celý princip princip komunikace bloku MF&CCTF se sondou je znázorněn na obrázku 6.1.



Obrázek 6.1: Komunikace sondy CC-IIF a Mediační funkce

Uvedené schéma je značně zjednodušené a uvažuje určitou úroveň abstrakce. Není zde řešena řada interních záležitostí MF&CCTF (konfigurace odposlechů s ohledem na časové intervaly, interní tabulky, apod.). Cílem obrázku je především poskytnout názorný pohled na vzájemnou komunikaci sond CC-IIF a bloku MF&CCTF.

Uvažujeme 1 až n sond CC-IIF. Pokud sonda požádá o navázání spojení na rozhraní INI3, je v MF zavolána funkce `processRequestINI3()`. V rámci této funkce je vytvořeno vlákno `ini3Thread()` v rámci nějž běží smyčka `ini3Loop()`, která zpracovává data zaslaná sondou. Při každém přijetí paketu na rozhraní INI3 je pak zavolána obslužná funkce `processMessageINI3()`. Jak bylo zmíněno v kapitole 4, implementace vlákna `ini3Thread()` a obslušných podfunkcí je řešena v Cythonu[2]. Při instalaci systému SLIS dojde nejprve k automatickému překladu do jazyka C a poté ke kompilaci do statických knihoven. Z tohoto důvodu je také tato část bloku MF&CCTF v samostaném modulu nazvaném `cython_layer`.

[2] http://cython.org/

V modelové situaci na obrázku 6.1 je znázorněna sonda 1 a sonda n, přičemž pro každou z nich v MF existuje samosatné vlákno `ini3Thread()`, ve kterém běží smyčka `ini3Loop()`. Uvažujme, že na rozhraní INI1c byly od *Administrační funkce* (AF) přijaty postupně tři požadavky na odposlech. Identifikátory (LIID) těchto odposlechů jsou *XXX*, *YYY* a *ZZZ*. V rámci odposlechu s LIID *XXX* jsou cílem zájmu jak metadata (IRI zprávy), tak celý obsah komunikace (pakety). V rámci odposlechu s LIID *YYY* jsou cílem zájmu pouze metadata. V rámci odposlechu s LIID *ZZZ* jsou cílem zájmu opět jak metadata tak obsah komunikace.

Blok *Funkce dynamické identity* (IRI-IIF) nalezl informace o identitě těchto cílů (IP adresy, resp. identifikátory TCP spojení) a tyto informace zaslal bloku MF&CCTF skrz rozhraní INI2 [18]. Protože cílem ukázky je ilustrovat možnost současné existence více sond, uvažujme, že sonda 1 je schopna odposlouchávat cíl odposlechu *XXX* a sonda n je schopna odposlouchávat cíl odposlechu *ZZZ*.

Blok MF&CCTF na základě obdržených požadavků a znalostech o identitě cílů rozhodl o další činnosti. U odposlechů *XXX* a *ZZZ* byl vyžadován záznam obsahu komunikace. Funkce CCTF (která je součástí bloku MF&CCTF) tedy poslala skrz rozhraní CCCI sondě 1 požadavek na odposlech *XXX* a sondě n požadavek na odposlech *ZZZ*. U odposlechu *YYY* nebyl záznam obsahu komunikace požadován, žádný požadavek sondám tedy poslán nebyl.

V rámci první smyčky `ini3Loop()` jsou obsluhovány pakety ze sondy 1, tedy pakety týkající se odposlechu *XXX*. V rámci druhé smyčky jsou pak obsluhovány pakety ze sondy n, tedy pakety týkající se odposlechu *ZZZ*. Každý paket, kterého se záznam týká je tedy příslušným vláknem ukládán do výstupního PCAP souboru. Vlákno 1 tedy ukládá pakety týkající se odposlechu *XXX* do souboru *XXX.pcap* a vlákno n ukládá pakety týkající se odposlechu *ZZZ* do souboru *ZZZ.pcap*.

Kromě ukládání obsahu komunikace (pro *XXX* a *ZZZ*) jsou pro všechny tři odposlechy ukládána metadata o zachycené komunikaci. V uvedeném případě jde o soubory *XXX.hi2*, *YYY.hi2* a *ZZZ.hi2*. Zdrojem metadat není sonda CC-IIF, nýbrž *Funkce dynamické identity* - blok IRI-IIF (viz kapitoly 3 a 5). Metadata jsou čtena ve formě IRI zpráv [18] z rozhraní INI2. Vnitřní implementace MF poté řeší jejich kategorizaci a zápis do výstupních *.hi2* souborů. Tato činnost s děje zcela nezávisle na vláknech `ini3Thread()`.

6.1.2 Nalezení optimálního způsobu profilace

Pro analýzu výkonnosti a identifikaci kritických míst bylo v rámci této práce experimentálně vyzkoušeno několik metod. Tyto zahrnovaly měření výkonnosti bloku MF&CCTF v reálných i simulovaných situacích - např. profilaci při současném „přehrávání“ referenčního PCAP souboru pomocí nástroje TCPreplay[3] na odposlouchávané síťové rozhraní. Jako optimální řešení se ukázalo využití jednoduchého generátoru paketů implementovaného v jazyce C. Tento generátor je schopen připojit se k MF&CCTF na rozhraní INI3 a emulovat sondu CC-IIF.

Další otázka se týkala samotné profilace - jaký způsob měření výkonnosti použít? Vzhledem ke skutečnosti, že některé části systému jsou implementovány v Cythonu, rozhodl jsem se použít nástroj cProfile[4]. Za předpokladu, že na začátek zdrojového souboru (*.pyx*) v Cythonu přidáme direktivu pro kompilátor: `# cython: profile=True`, je cProfile schopen měřit i výkonnost funkcí zkompilovaných do statických binárních knihoven.

[3] http://tcpreplay.synfin.net/

[4] https://docs.python.org/2/library/profile.html

Záhy jsem však identifikoval další problém - cProfile nepodporuje profilaci funkcí volaných v rámci nově vytvořených vláken. Po vyzkoušení řady dalších způsobů (včetně implementace vlastního modulu pro manuální měření systémového času) jsem se rozhodl pomocí cProfile profilovat vlákno `ini3Thread()` samostatně. Pro tyto účely jsem do systému SLIS přidal možnost běhu bloku MF&CCTF v režimu automatické profilace vlákna `ini3Thread()`. Tento režim se aktivuje, pokud v konfiguračním souboru `profiling.ini` nastavíme v sekci `[mf]` volbu `profile_ini3thread = True` a spustíme systém SLIS.

Kdykoli v tomto režimu přijde od sondy žádost o navázání spojení na rozhraní INI3, ihned po vytvoření vlákna `ini3Thread()` se spustí profilace. V rámci této je měřena výkonnost funkce `ini3Loop()` a všech dalších funkcí volaných v tomto vlákně. Po ukončení spojení se sondou je profilace ukončena, výsledky jsou automaticky setříděny a uloženy do souboru pojmenovaného: `profile_ini3thread_Thread-[ID vlákna]`.

6.1.3 Profilace a identifikace kritických míst

Způsobem popsaným výše byla provedena řada profilací vlákna `ini3Thread()`. Zkrácený výpis jednoho z naměřených výstupů profilace pomocí nástroje cProf ukazuje výstup 6.2. Celkem bylo na rozhraní INI3 posláno $10 * 2^{20}$ paketů s daty o náhodné velikosti v rozmezí 64 B až 1500 B. Každý řádek představuje jedno volání funkce. Výpis je řazen sestupně dle času stráveného prováděním daných funkcí a je omezen na prvních 7 volání, které mají z hlediska doby zpracování největší význam.

```
62914689 function calls in 237.429 seconds
Ordered by: internal time

  ncalls  tottime  percall  cumtime  filename:lineno(function)
10485760  145.908    0.000  187.471  cython_layer.pyx:62(processMessageINI3)
       1   49.958   49.958  237.429  cython_layer.pyx:136(ini3Loop)
10485760   15.341    0.000   19.339  liid_table.pyx:111(getPCAPFileWriters)
10485760   10.874    0.000   22.224  cython_layer.pyx:47(GetLIIDsForRID)
10485760    7.434    0.000    7.434  sid_table.py:67(getLIIDs)
10485760    3.998    0.000    3.998  liid_table.pyx:90(getPeriodUnix)
10485760    3.916    0.000    3.916  sid_table.py:148(EntryExists)
...
```

Obrázek 6.2: Ukázka profilace vlákna ini3Thread()

Sloupec `ncalls` ukazuje počet volání každé funkce. Jak již bylo uvedeno v sekci 6.1.2, profilujeme jeden konkrétní běh vlákna `ini3Thread()`. V rámci tohoto vlákna pak vznikne smyčka `ini3Loop()`. Počet volání této funkce je tedy roven 1. Zbylý počet volání odpovídá počtu zaslaných paketů: $10 * 2^{20} = 10485760$. Z hlediska měření výkonnosti jsou nejvíce podstatné sloupce `tottime` a `cumtime`. Sloupec `tottime` představuje celkový čas (v jednotkách sekund) strávený prováděním dané funkce (pouze uvedené funkce - nikoli funkcí z ní volaných). Sloupec `cumtime` pak představuje celkový kumulovaný čas (v jednotkách sekund) strávený prováděním příslušné funkce včetně funkcí z ní volaných. Poslední uvedený sloupec má formát `název-souboru:řádek(volaná-funkce)`.

Při pohledu na sloupec `tottime` vidíme, že nejvíce času je stráveno prováděním funkce `processMessageINI3()` - ta se volá ze smyčky `ini3Loop()` při každém přijetí paketu na

rozhraní INI3 (viz obrázek 6.1 a související popis). Ostatní uvedené funkce jsou pak volány právě z funkce `processMessageINI3()`.

Celkový kumulovaný čas provádění smyčky `ini3Loop()` je 237,429 s. Z Toho 187,471 s (tedy zhruba 79%) je stráveno prováděním funkce `processMessageINI3()`. Zbylých 49,958 s představuje především čtení dat za socketu INI3. Nejvíce kritickým místem je tedy funkce `processMessageINI3()`.

Účelem funkce `processMessageINI3()` je analýza přijatého paketu, jeho kategorizace a nakonec zápis do příslušného výstupního PCAP souboru. Zápis do souboru je vstupně/výstupní operace a její čas trvání silně závisí na použité architektuře (typu disku, souborového systému, apod.). V rámci projektu Sec6Net již proběhl nezávislý výzkum efektivity zápisu na disk (mechanický vs. SSD disk, EXT4 vs. XFS, apod.), nicméně uvedená problematika není náplní této práce.

Zaměřme se tedy na časový interval mezi voláním funkce `processMessageINI3()` a samotným zápisem do PCAP souboru. Z kapitoly 3 víme, že každý paket zaslaný sondou CC-IIF je identifikován unikátním identifikátorem: RID (reason ID). Abychom získali název výstupního PCAP souboru (resp. ukazatel na objekt typu PCAPFileWriter(), který slouží k zápisu do něj), je třeba provést řadu transformací a prohledávání interních tabulek bloku MF&CCTF [18]:

- (**ProbeID**, **RID**) → **SID** - Na základě identifikátoru sondy (probeID) a identifikátoru zachyceného paketu (RID) získáme identifikátor množiny odposlechů (SID). Z kapitoly 3 víme, že existuje surjektivní zobrazení (ProbeID, RID) → SID. Toto zobrazení zajišťuje interní tabulka `rid_sid_table` která je součástí bloku MF&CCTF.
- **SID** → množina **LIID** - Tento převod zajišťuje funkce `getLIIDs()`, kterou vidíme ve výpisu 6.2. Funkce `getLIIDs()` je metodou objektu třídy `SIDTable` - viz kapitoly 3 a 5. `SIDTable` představuje další interní tabulku bloku MF&CCTF.
- **LIID** → **PCAPFileWriter** - Pro každý identifikátor odposlechu(LIID) z výše uvedené množiny získáme ukazatel na objekt typu PCAPFileWriter(). K tomuto účelu slouží funkce `getPCAPFileWriters()`, kterou rovněž vidíme ve výpisu 6.2. Funkce `getPCAPFileWriters()` je metodou objektu třídy `LIIDTable`, tedy další interní tabulky bloku MF&CCTF.
- **LIID** → (**start_unix**, **stop_unix**) - Pro každý identifikátor odposlechu(LIID) potřebujeme získat také interval (v UNIX timestamp[5]) - čas začátku a konce odposlechu. Zápis do souboru bude proveden pouze tehdy, pokud čas přijetí paketu spadá do nalezeného intervalu. Pro získání tohoto intervalu slouží funkce `getPeriodUnix()`, kterou rovněž vidíme ve výpisu 6.2 a která je také metodou objektu třídy `LIIDTable`. Tato kontrola je prováděna, protože může nastat situace, kdy odposlech již není aktivní a stále existuje příslušný záznam v tabulce LIID. Příčin může být několik: jednak z důvodu konzistence datových struktur dochází k zamykání interních tabulek mezi vlákny (pomocé zámku `internal_tables_lock`) - přístup k tabulce by mohl být blokován jiným vláknem. A jednak jsou v systému zavedena úmyslná zpoždění - viz kapitolu 3.

Jedním z kritických míst z hlediska výkonnosti bloku MF&CCTF je obrovská režie postupného prohledávání několika tabulek, které je prováděno pro každý přijatý paket. Výpis 6.2

[5] http://en.wikipedia.org/wiki/Unix_timestamp

navíc vznikl na základě experimentu, kdy v sytému byl aktivní pouze jeden odposlech. Interní tabulky tedy obsahovaly minimum záznamů. V případě experimentů s více odposlechy (především několika se stejným SID) byla tato režie ještě vyšší (viz kapitolu 7).

6.2 Optimalizace nalezených kritických míst

6.2.1 Návrh způsobu optimalizace

V sekci 6.1.2 byla identifikována kritická místa při zpracování paketů v bloku MF&CCTF. Obrovskou režii má prohledávání:
(**ProbeID**, **RID**) → **SID** → množina **LIID**

A pro každé LIID z této množiny pak:
LIID → **PCAPFileWriter**,
LIID → (**start_unix**, **stop_unix**).

Po důkladném zvážení různých možností optimalizace jsem se rozhodl využít vlastní způsob optimalizace. Tento způsob je vzdáleně inspirován technologií IP CEF[6] od firmy Cisco. Myšlenky optimalizace spočívá ve vytvoření rychlé vyhledávací tabulky, která bude přímo poskytovat mapování:

(**ProbeID**, **RID**) → (**PCAPFileWriter**, **start_unix**, **stop_unix**)

Tato tabulka bude obsahovat automaticky předpočítané všechny validní kombinace hodnot z původních tří tabulek:

- tabulka RID-SID,
- tabulka SIDů,
- tabulka LIID.

Tato rychlá vyhledávací tabulka pak při zpracovávání paketů z rozhraní INI3 zcela nahradí volání `getLIIDsForRID()`, `getPCAPFileWriters()`, `getLIIDs()`, `getPeriodUnix()`, apod.

Při změně kterékoli z původních tří tabulek dojde k invalidaci hodnot rychlé vyhledávací tabulky a proběhne opětovný přepočet všech validních kombinací. Tato myšlenka je přijatelná, protože ke změnám některé ze tří tabulek dochází pouze, pokud:

- bude přidán/odebrán odposlech,
- dojde ke změně identity cíle některého z odposlechů.

Efekt této optimalizace bude tím vyšší, čím méně změn vnitřních tabulek (a následných přepočtů) bude prováděno v porovnání s množstvím paketů zaznamenávaných sondami. Změna identity silně závisí na charakteru dané sítě. Pokud budeme uvažovat např. protokol DHCP [3], např. v systémech Microsoft Windows je výchozí délka platnosti přidělení IP adresy stanovena na 8 dní[7]. Jiná situace bude v případě, kdy bude uživatel mobilní (např.

[6] http://www.cisco.com/c/en/us/support/docs/routers/12000-series-routers/47321-ciscoef.html

[7] http://technet.microsoft.com/en-us/library/dd183602%28v=ws.10%29.aspx

bezdrátové připojení skrz mobilní telefon, apod.). Další otázkou je frekvence přidávání a odebírání odposlechů. Uvažme však, že i v extrémním případě, kdy by byly neustále přidávány a odebírány odposlechy, je pro přidání/odebrání odposlechu nutná interakce uživatele (skrz webové rozhraní systému SLIS). Doba vyplnění webového formuláře pro zadání nového odposlechu je v řádech sekund až minut (ověřeno experimentálně). Efektivita popisované optimalizace bude samozřejmě také záviset na charakteru odposlouchávané komunikace. Ale např. u VoIP telefonie při použití kodeku G.723.1 je přenášeno průměrně 33.3 paketů/s, při použití kodeku G.711 pak průměrně 50 paketů/s[8]. V souvislosti s těmito rychlostmi je frekvence změn identity cíle, či konfigurace odposlechů zanedbatelná. Z výše uvedeného usuzuji že navržený optimalizace má při reálném nasazení (tedy ve většině reálných situací) smysl.

6.2.2 Realizace navrženého způsobu optimalizace

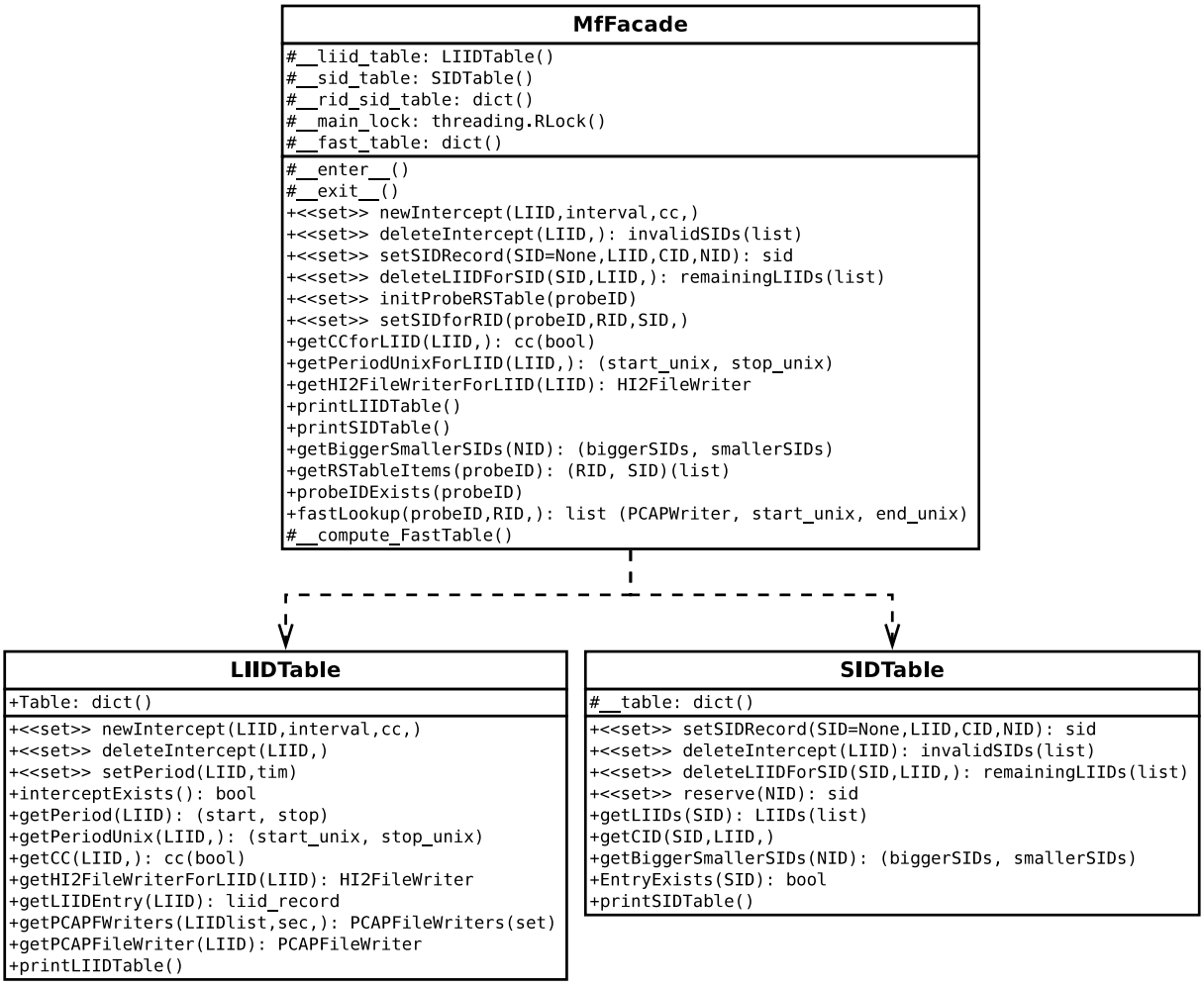
Po zvážení různých možností implementace jsem se rozhodl vytvořit novou třídu MfFacade(). Tato třída odpovídá návrhovému vzoru *fasáda* (Facade pattern)[13]. Z pohledu koncepce systému SLIS a principů objektově orientovaného programování splňuje tato třída následující vlastnosti:

- Zapouzdřuje všechny tři interní tabulky využívané ke zpracování paketů v bloku MF&CCTF. Tedy:
  - tabulka RID-SID,
  - tabulka SIDů,
  - tabulka LIID.
- Uchovává v sobě rychlou vyhledávací tabulku `__fast_table`. Přístup k ní je zajištěn pomocí metody `fastLookup()`. Metoda `fastLookup()` pro konkrétní dvojici (**ProbeID**, **RID**) vrátí trojici (**PCAPFileWriter**, **start_unix**, **stop_unix**) dle konceptu uvedeného výše.
- Poskytuje abstraktní rozhraní pro operace s původními tabulkami (`newIntercept()`, apod.). Některé metody přitom slučuje do jedné - např. poskytuje jednu metodu `deleteIntercept()`, která aktualizuje jak tabulku SIDů, tak tabulku LIID.
- Zajišťuje možnost souběžného přístupu z více vláken (z hlavního vlákna MF i z vláken `ini3Thread()` pro zpracování paketů z jednotlivých sond).
- Zajišťuje konzistenci vnitřních struktur a vzájemné vyloučení pomocí vnitřního zámku `__main_lock`.
- Zajišťuje automatický výpočet a aktuálnost obsahu rychlé vyhledávací tabulky. V případě úpravy některé z vnitřních struktur je proveden automatický přepočet chráněnou metodou `__compute_FastTable()`.
- Kromě vnitřního reentrantního zámku (pro zajištění konzistence) používaného uvnitř metod třída MfFacade implementuje také obecné metody `__enter__` a `__exit__`. To poskytuje možnost využití bloků kódu uvnitř klauzule `with mf_facade`, kde `mf_facade` je objekt třídy `MfFacade`. Toto použití se provádí v situacích, kdy potřebujeme získat

[8] http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html

odkaz na HI2Writer metodou `getHI2FileWriterForLIID()` nebo odkaz na PCAPFileWriter metodou `fastLookup()`, přičemž s těmito objekty chceme po určitou dobu pracovat. Pokud tyto operace provádíme uvnitř klauzule `with mf_facade`, je zajištěno, že i po vrácení odkazu na HI2/PCAPWriter můžeme s těmito objekty pracovat a žádné jiné vlákno nemůže vyvolat jejich zánik (např. při odebrání odposlechu ze systému).

Obrázek 6.3 ukazuje UML diagram třídy MfFacade společné s jejími atributy a metodami. Je zde také zakreslena závislost na původní (nyní zapouzdřené) třídy LIIDTable() a SIDTable(). Metody stereotypu «set» představují takové operace, které mají za následek úpravu některé z vnitřních tabulek. Na konci každé z těchto metod je proveden nový automatický předpočet rychlé vyhledávací tabulky. Díky tomu při volání `fastLookup()` získáme vždy aktuální platné výsledky.



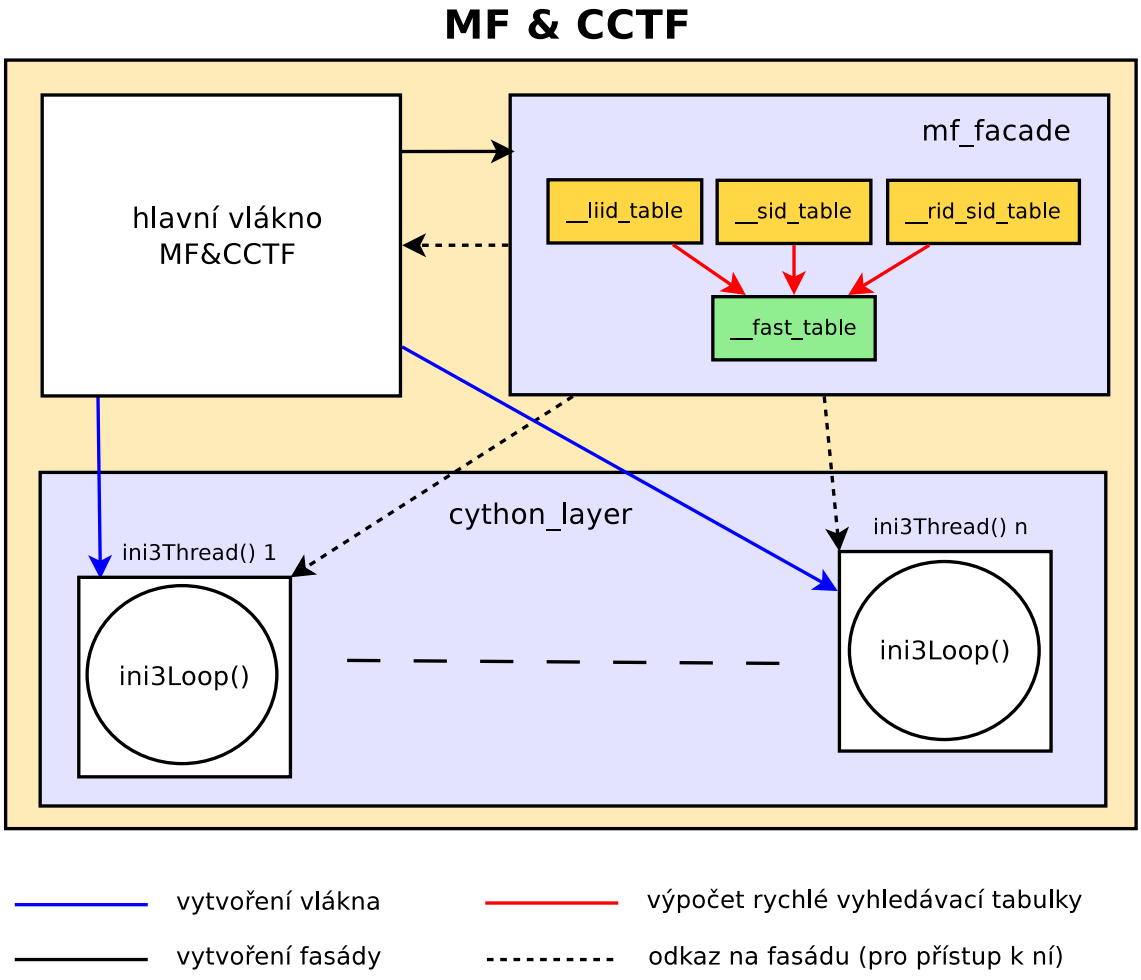
Obrázek 6.3: UML diagram fasády MfFacade se závislostmi na LIIDTable a SIDTable

Navržená fasáda poskytuje jednak optimalizaci (díky předpočítané rychlé vyhledávací tabulce), jednak daleko jednodušší práci se strukturami uvnitř bloku MF&CCTF. Při navázájí spojení se sondou je vytvořenému vláknu `ini3Thread()` předán odkaz na objekt

fasády a toto vlákno s ní pak může pracovat. Až dosud bylo nutno složitě (i uvnitř vlákna `ini3Thread()`) předávat formou argumentů odkazu na každou z tabulek zvlášť a stejně tak bylo nutné předávat odkaz na zámek `internal_tables_lock`.

Celý princip požití fasády uvnitř bloku MF&CCTF je znázorněn na obrázku 6.4. Je zde viditelná souvislost se schématem na obrázku 6.1. Hlavní vlákno bloku MF&CCTF vytvoří objekt fasády (`mf_facade = MfFacade()`) a od této chvíle má přístup k jeho metodám. Pokud přijde požadavek na přidání/odebrání odposlechu, volá hlavní vlákno metody fasády.

Pokud přijde od sondy CC-IIF požadavek na navázání spojení na rozhraní INI3, vytvoří hlavní vlákno MF&CCTF nové vlákno `ini3Thread()` a jako argument mu předá odkaz na existující fasádu. Objekt `mf_facade` existuje po celou dobu běhu systému jako jediná instance třídy `MfFacade`. Fasáda tak tvoří jednotný celek a zapouzdřuje všechny podstaté vnitřní datové struktury bloku MF&CCTF.



Obrázek 6.4: Ilustrace použití fasády uvnitř bloku MF&CCTF

Efektivita optimalizace Mediační funkce pomocí rychlé vyhledávací tabulky a fasády je ukázána na experimentech v kapitole 7.

Kapitola 7

Experimenty a nasazení v praxi

V této kapitole je řada experimentů a praktických ukázek práce se systémem SLIS. V sekci 7.1 je demonstrována správná činnost systému při detekci identity na základě aplikačního identifikátoru - IRC loginu a při realizaci samotného záznamu obsahu zájmové komunikace (IRC komunikace v rámci nalezeného TCP spojení). Sekce 7.2 ukazuje praktické nasazení implementovaného rozšíření při další výzkumné činnosti v rámci projektu Sec6Net. V sekci 7.3 je krátce popsána jedna z demonstrací funkčnosti systému pro Policii ČR. Sekce 7.4 pak popisuje řadu experimentů, jejichž cílem bylo určit efektivitu optimalizace bloku MF&CCTF.

7.1 Detekce identity a odposlech aplikačních protokolů

7.1.1 Detekce identity na základě IRC loginu a odposlech zájmové IRC komunikace

Cílem tohoto experimentu bylo ověřit funkčnost nové implementace jádra IRI-IIF s podporou detekce identity na základě aplikačních identifikátorů. Dalším cílem bylo ověřit provedené implementační úpravy v bloku MF&CCTF, na rozhraní CCCI a v softwarové sondě CC-IIF a zjistit tak, zda byl skutečně uložen pouze obsah zájmové komunikace.

Pro účely experimentu byl pridán jeden odposlech s LIID `wwwww`. Cílem odposlechu byl IRC login *vmlemon@sterling.freenode.net*. Požadavek byl zadán skrz webové rozhraní systému SLIS - viz obrázek - 7.1.

Current interceptions

Active interceptions

LEA	LIID	NID	Level	Start	End	CC
PolicieCR	wwwww	'ircLogin:vmlemon@sterling.freenode.net'	1	Tue Mar 11 20:58:00 2014	Wed Mar 11 00:00:00 2015	True

Obrázek 7.1: Nastavení odposlechu s cílem na IRC login

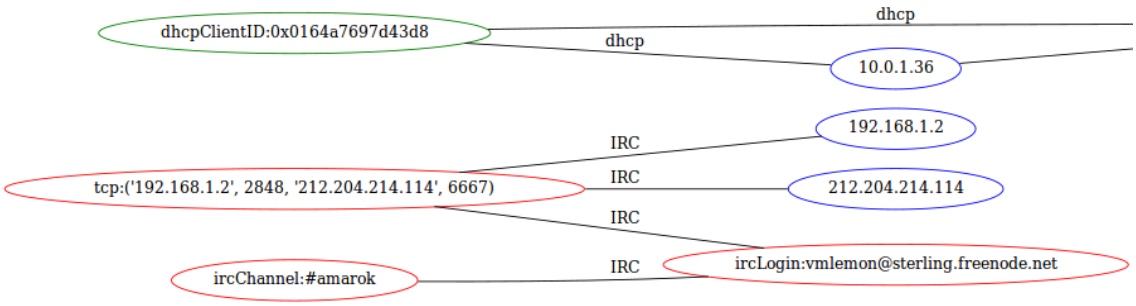
Následně byl programem TCPreplay[1] přehrán na síťové rozhraní referenční PCAP se zá-

[1] http://tcpreplay.synfin.net/

znamem Skype[2] a IRC [17] komunikace: `SkypeIRC.pcap` - k dispozici na CD (viz přílohu A).

Funkce dynamické identity (Blok IRI-IIF) detekovala, že uživatel s IRC loginem *vmlemon@sterling.freenode.net* má IP adresu 192.168.1.2 a je připojen k serveru 212.204.214.114 na IRC kanál *#amarok*. Dále bylo blokem IRI-IIF zjištěno, že komunikace probíhá v rámci TCP spojení 192.168.1.2:2848 ↔ 214.204.214.114:6667. Výsledek detekce ukazuje výřez grafu na obrázku 7.2.

Dále bylo třeba ověřit předpoklad, že nalezený identifikátor TCP spojení byl úspěšně předán Mediační funkci a že blok CCTF nakonfiguroval sondu CC-IIF právě k odposlechu tohoto TCP spojení.



Obrázek 7.2: Grafová reprezentace stavu sítě vytvořená jádrem IRI-IIF

Obrázek 7.3 ukazuje, že byly vytvořeny soubory `wwwww.hi2` (pro ukládání metadat o komunikaci) a `wwwww.pcap` (pro ukládání zachycené komunikace). Z výpisu je zřejmé, že oba soubry mají nenulovou velikost. Zbývá tedy ověřit, zda sonda CC-IIF odfiltrovala skutečně pouze obsah zájmové komunikace.

K tomuto účelu byl použit program Wireshark[3], ve kterém byl otevřen PCAP soubor se zachycenými daty (viz obrázek 7.3). Ze záznamů bylo zjištěno, že ačkoli referenční PCAP soubor obsahovat i jiný provoz, systém úspěšně zaznamenal pouze zájmové TCP spojení.

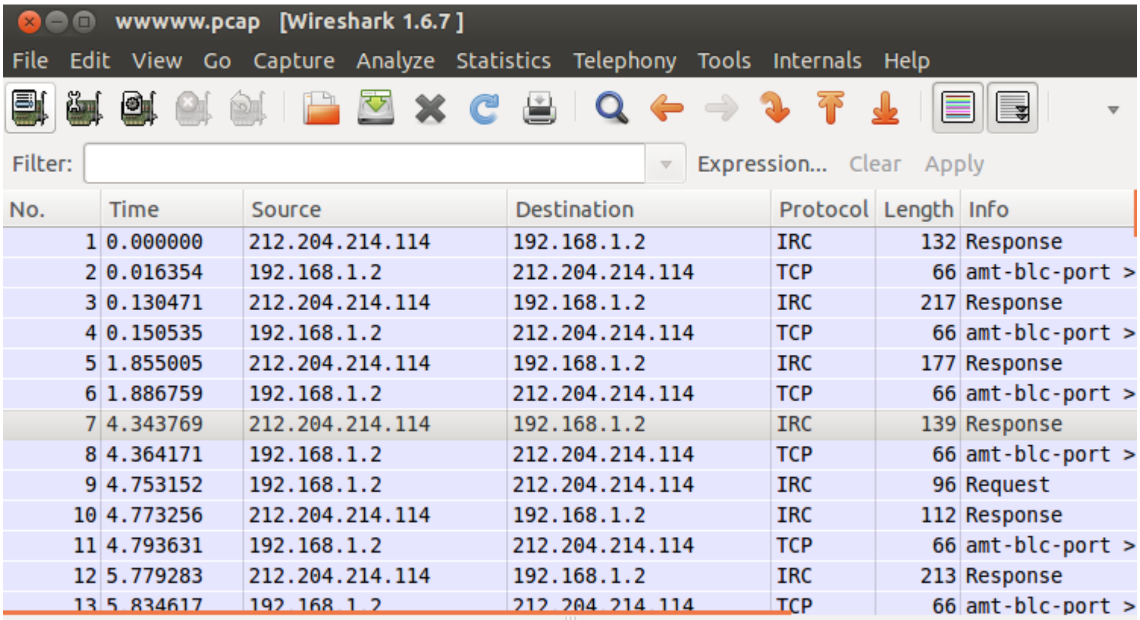
```
-rw-r--r--  1 root root   272 bře 13 00:52 wwwww.hi2
-rw-r--r--  1 root root 61464 bře 13 00:53 wwwww.pcap
slis@slis-vm:/opt/slis/light$
```

Obrázek 7.3: Soubory se zachycenými metadaty a obsahem komunikace

7.2 Praktické nasazení v rámci projektu

Systém SLIS s rozšířením o jádro IRI-IIF s grafovou reprezentací (viz kapitolu 5) se již prakticky používá v rámci projektu Sec6Net. Pro tyto účely byla vytvořena řada virtuálních serverů, z nich každý má svůj účel a specifickou topologii. Seznam aktivně používaných testovacích strojů je znázorněn v tabulce 7.1. Vidíme, že se servery se liší jak účelem, tak instalovanou verzí operačního systému (sloupec OS).

[2] http://www.skype.com
[3] http://www.wireshark.org/



Obrázek 7.4: Zaznamenaný obsah IRC komunikace v programu Wireshark

Název serveru	OS	Účel serveru
sec6net-mv3.fit.vutbr.cz	Ubuntu 10.04 LTS	Vývoj systému SLIS
sec6net-mv4.fit.vutbr.cz	Ubuntu 10.04 LTS	Vývoj systému SLIS
sec6net-mv5.fit.vutbr.cz	Ubuntu 12.04 LTS	Radek Hranický - test. podpory apl. protokolů
sec6net-mv6.fit.vutbr.cz	Ubuntu 12.04 LTS	Barbora Franková - test. Clock skew modulu
sec6net-mv7.fit.vutbr.cz	Ubuntu 12.04 LTS	Testování vysokorychlostní sondy
sec6net-mv8.fit.vutbr.cz	Ubuntu 12.04 LTS	Vývoj systému SLIS
sec6net-mv18.fit.vutbr.cz	Ubuntu 13.04 - Mininet+ODL	Libor Polčák - testování ODL
sec6net-mv19.fit.vutbr.cz	Ubuntu 13.04 - Mininet+ODL	Barbora Franková - testování ODL

Tabulka 7.1: Virtuální servery používané v rámci projektu Sec6Net

Virtuální servery jsou zapojeny do sítě FIT pouze pomocí protokolu IPv6 s adresami 2001:67c:1220:809::X:93e5:949, na které ukazují v DNS záznamy sec6net-mvX.fit.vutbr.cz. Počítače mají několik síťových rozhraní. Jedním jsou připojeny k testovací síti, která je nakonfigurovaná jako broadcastové médium, tzn. „všichni vidí vše“.

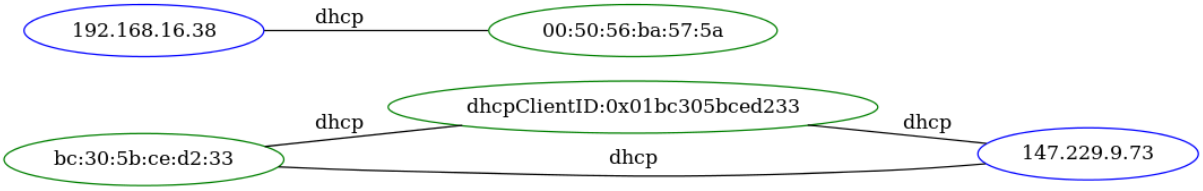
Druhým rozhraním jsou servery připojeny k jiné testovací síti, která slouží pro experimenty s DHCP relay [3]. Na počítači sec6net-mv3.fit.vutbr.cz běží DHCP server přidělující IPv4 adresy zbytku sítě, na počítači sec6net-mv5.fit.vutbr.cz běží DHCP relay, který naslouchá požadavkům z testovací sítě a přeposílá je DHCP serveru.

Kromě těchto dvou rozhraní má každý server k dispozici ještě speciální rozhraní, kterým je připojen k síti NAT, kde je možné zajistit přístup k IPv4 Internetu pomocí překladu na počítači sec6net-mv2.fit.vutbr.cz (pod správavou DRaP, není zde vyobrazen).

Obrázek 7.5 ukazuje schéma grafu vygenerovaného jádrem IRI-IIF při testování modulu pro Clock skew a platformy OpenDaylight[4]. Vidíme, že není problém rozšířit systém o nové identifikátory NID. V tomto případě byl přidán identifikátor sdnConnector. Na obrázku 7.6 vidíme graf z průběhu testování modulu pro DHCP.

[4] http://www.opendaylight.org/

Obrázek 7.5: Graf stavu sítě při testování modulu pro Clock skew



Obrázek 7.6: Graf stavu sítě při testování modulu pro DHCP

7.3 Demonstrace činnosti systému pro policii ČR

Dne 22. 2. 2013 proběhla v praze demonstrace systému SLIS pro policii České republiky. Účelem demonstrace bylo seznámit odpovědné osoby z řad příslušných útvarů policie ČR se systémem SLIS a předvést jeho činnost v praxi. Byla vysvětlena celá architektura systému a účel jednotlivých funkčních bloků. Součástí byla ukázka možné síťové topologie při nasazení systému SLIS.

Dále byl věnován prostor vyvíjeným hardwarovým prototypům mikro-sondy a vysokorychlostní sondy. Součástí této demonstrace bylo představení použité platformy, schématu zapojení a praktických ukázek.

V průběhu demonstrace byla vysvětlena problematika detekce identity IPv4 a IPv6 sítích. Byla zde vysvětlena činnost bloku IRI-IIF s praktickou ukázkou činnosti jednotlivých modulů systému. Mezi možnostmi detekce identity byla zmíněna analýza událostí na síti v rámci protokolů DHCP, RADIUS, PPPoE. V sítích IPv6 pak šlo o protokoly SLAAC a DHCPv6. Nastíněna byla také možnost detekce počítačů pomocí časových značek.

Jak lze vidět z propagačního plakátu na obrázku 7.7, byl zde popsán princip jádra IRI-IIF založeného na grafové reprezentaci. Součástí bylo vysvětlení celkového konceptu grafu, identifikátorů NID a možnosti konfigurovatelných úrovní odposlechu. Grafová reprezentace stavu sítě je popsána v dolní části tohoto obrázku.

Celá demonstrace proběhla úspěšně a výsledkem byla spokojenost ze strany pracovníků policie ČR, kteří měli možnost si práci se systémem vyzkoušet také osobně.

Sec6Net Lawful Interception System (SLIS)

Skupina pro zákonné odposlechy projektu Sec6Net

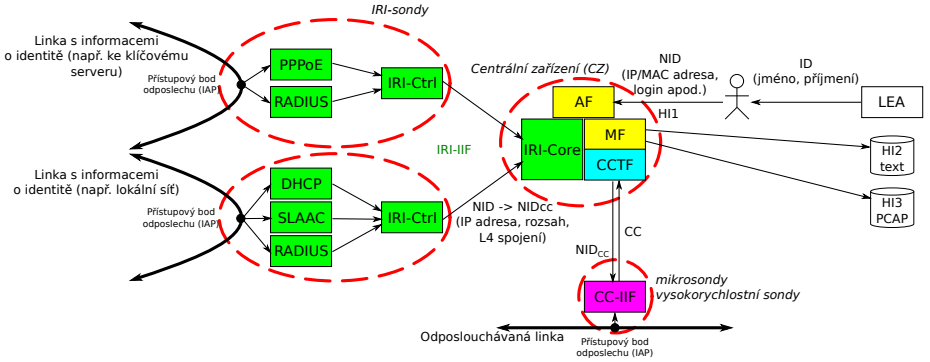
{ipolcak, martinto, matousp}@fit.vutbr.cz
http://www.fit.vutbr.cz/~ipolcak/grants.php?id=517

Architektura systému pro zákonné odposlechy (SLIS)

- Založená na ETSI a RFC 3924
- 1 Centrální zařízení
- 1-n IRI sond
  - mohou mít různé konfigurace
  - může být součástí centrálního zařízení
- Sondy pro odposlech provozu
  - Mikrosonda
  - Vysokorychlostní sonda

1) Na základě databáze operátora se identifikace odposlouchávaného převede na neměnný identifikátor NID.
2) V rámci IRI-IIF je detekováno přiřazení IP adresy, rozsahu adres, nebo konkrétní spojení na úrovni transportní vrstvy (TCP/UDP) - NID_{CC}.
3) Konfigurace CC-IIF pomocí NID_{CC}.
4) Odposlechnutá data (CC) jsou do MF zaslaná pouze jednou (nezávisle na počtu agentur a odposlechů, kterých se konkrétní CC týká).
- Duplikace dat v MF

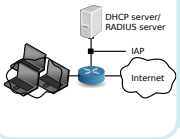
- Šifrování přenosu pomocí SSH
- IAP řešeny pomocí:
  - TAPů
  - Port mirroring
  - Vestavěný tap v mikrosondě



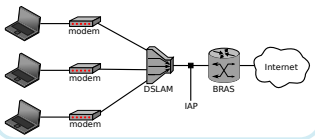
Detekce identity v systému pro zákonné odposlechy (SLIS)

Detekce identity v IPv4

- DHCP/RADIUS



- PPPoE

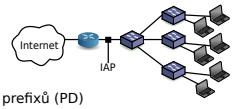


Časové značky

- Není nutné zařízení pro odposlech v lokální síti
  - Přiřazení toků k počítačům za překladačem adres
  - Přiřazení dočasných adres IPv6 ke konkrétním počítačům
- TCP, automaticky: MAC OS, Linux, BSD, Android, iOS
- JavaScript: nutnost přístupu ke konkrétnímu serveru, nezávislé na OS

Detekce identity v IPv6

- Objevování sousedů (novinka v IPv6)
  - SLAAC (bezestavová konfigurace), statické adresy, DHCPv6
  - Dočasné adresy, více adres na jednom rozhraní, adresy generované koncovými počítači



- DHCPv6
  - Možnost delegace prefixů (PD)
  - Počítač může používat i adresy získané jiným způsobem
- RADIUS
  - Nové položky pro IPv6 adresy
- PPPoE
  - Modem získává pouze adresu pro komunikaci v lokální síti
  - Často kombinované s DHCPv6-PD

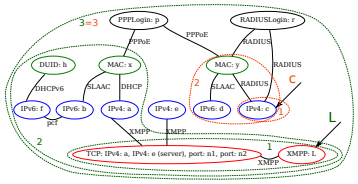
Úrovně odposlechů v systému pro zákonné odposlechy (SLIS)

Podporované identifikátory (NIDy)

A) Aplikační identifikátor
  - Aplikační login (e-mail, XMPP login, ICQ # apod.)
  - 5-tice zdrojové/cílové IP adresy a porty, transportní protokol
  - 3-jice IP adresa, transportní protokol a port
B) Adresa síťové vrstvy (IPv4/IPv6)
C) Adresa síťového rozhraní/identifikátor konkrétního počítače
  - MAC adresa
  - DHCPv6 DUID
D) Ostatní identifikátory (typicky využívané pro autentizaci)
  - RADIUS login
  - PPP login, nebo číslo spojení

Podporované úrovně odposlechů

1) Nejmenší rozsah
2) Data konkrétního síťového rozhraní či počítače
3) Všechna data daného uživatele



Tato práce je součástí projektu VG20102015022 podporovaného Ministerstvem vnitra České republiky (Sec6Net).

Obrázek 7.7: Plakát z demonstrace prototypu systému SLIS

7.4 Měření výkonnosti MF&CCTF před a po optimalizaci

Cílem následujících experimentů bylo změřit efektivitu optimalizace Mediační funkce [18] popsané v kapitole 6.

7.4.1 Použitý postup

Vzhledem k použité metodě optimalizace a k implementaci vnitřních struktur v bloku MF&CCTF odhaduji, že výsledný efekt optimalizace bude silně závislý na počtu záznamů ve vnitřních tabulkách MF. Z těchto důvodů jsem se rozhodl provést několik experimentů s různými počty přidaných odposlechů v systému. Vzhledem k současné implementaci metody `getLIIDs()` v tabulce SIDů odhaduji, že efekt optimalizace bude tím vyšší, čím více odposlechů se stejným SID bude v systému existovat [18]. Dvěma odposlechům je přiřazen stejný SID, pokud se shodují, či překrývají jejich cíle - podrobněji je tato problematika popsána v kapitolách 3 a 5.

Celkem bylo provedeno pět experimentů s následujícími konfiguraci odposlechů:

- V systému je pouze 1 odposlech s příznakem CC (je žádoucí záznam obsahu komunikace).
- V systému je kromě 1 odposlechu s příznakem CC ještě další odposlech bez příznaku CC, ovšem se stejnou přiřazenou hodnotou SID.
- 1 odposlech s příznakem CC + 2 další odposlechy (bez příznaku CC) se stejným SID.
- 1 odposlech s příznakem CC + 5 další odposlechů (bez příznaku CC) se stejným SID.
- 1 odposlech s příznakem CC + 10 další odposlechů (bez příznaku CC) se stejným SID - extrémní případ, který je v praxi velice málo pravděpodobný.

Důvod, proč u ostatních odposlechů není nastaven příznak odposlechu obsahu komunikace (CC) vychází ze samotné podstaty optimalizace. Jejím cílem není optimalizovat zápis do výstupního souboru, nýbrž zkrátit čas mezi přijetím paketu z rozhraní INI3 a samotným zápisem.

Měření výkonu probíhalo na virtuálním počítači se systémem Ubuntu 12.04 LTS. Pro účely virtualizace byla použita aplikace VMWare player[5]. Požitým hostitelským počítačem byl notebook *Toshiba Qosmio X70-A-12X*. Virtuálnímu počítači bylo vyhrazeno 16 GB z dostupných 32 GB paměti RAM DDR3L a přiděleny 3 ze 4 jader procesoru *Intel Core i7 4700MQ Haswell*.

Při experimentech bylo použito stejné metody profilace jako byla popsána v kapitole 6 - tedy automatická profilace každého vytvořeného vlákna `ini3Thread()` pomocí nástroje cProfile[6].

Obdobný byl i způsob generování paketů. Pro experimenty byl použit stejný generátor jako v kapitole 6. Jak bylo uvedeno výše, cílem optimalizace není zefektivnit zápis do souboru, či čtení dat z rozhraní INI3. Z tohoto důvodu byla pro všechny generované pakety nastavena stejná hodnota obsahu(payload): 64 B. Náhodné (nedeterministické) velikosti paketů

[5] http://www.vmware.com/products/player
[6] https://docs.python.org/2/library/profile.html

```
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./addint.sh 5
Adding main incerception.
Adding 5 other interceptions with same SID.
Adding extra interception no. 1
Adding extra interception no. 2
Adding extra interception no. 3
Adding extra interception no. 4
Adding extra interception no. 5
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./cc_test 127.0.0.1 21103
Sent bytes: 83886086
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./cc_test 127.0.0.1 21103
Sent bytes: 83886086
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./cc_test 127.0.0.1 21103
Sent bytes: 83886086
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./cc_test 127.0.0.1 21103
Sent bytes: 83886086
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ./cc_test 127.0.0.1 21103
Sent bytes: 83886086
slis@slis-vm:~/hranicky_li_optimized/src/mf_perf/mf$ ▯
```

Obrázek 7.8: Přidání odposlechů pomocí připraveného skriptu a generování paketů

v tomto případě nejsou žádoucí. Důvod je zřejmý - paket s větším obsahem bude zapisován déle než paket s kratším obsahem dat. Tato skutečnost by mohla zkreslovat naměřené výsledky. Z těchto důvodů byly ve všech pěti experimentech použity pakety statické velikosti.

V rámci každého experimentu bylo provedeno 5 nezávislých měření. V rámci každého měření bylo vygenerováno a posláno na rozhraní INI3 2^{20} paketů.

Pro přidání odposlechů byl vytvořen jednoduchý automatizovaný skript. Jeho použití v rámci jednoho z experimentů (společně z generátorem paketů) je možné vidět na obrázku 7.8. Obrázek 7.9 pak ukazuje ladící výpisy bloku MF&CCTF kde můžeme spatřit obsah všech interních tabulek včetně předpočítaného obsahu rychlé vyhledávací tabulky. Z výpisu lze vyčíst obsah:

- **Tabulky LIID** - označeno na obrázku 7.9 jako sekce `==> LIID TABLE <==`, kde vidíme postupně identifikátor (LIID) daného odposlechu, datum a čas zahájení (resp. ukončení) odposlechu. Poslední sloupec udává příznak CC - tedy zda má být zaznamenáván obsah komunikace. V tabulce LIID jsou uloženy také objekty tříd `HI2Writer` a `PCAPFileWriter`, ty však nejsou ve výpisu uvedeny.

- **Tabulky SIDů** - sekce `==> SID TABLE <==`, kde vidíme identifikátor SID a k němu spadající identifikátor odposlechu (LIID), síťoví identifikátor odposlouchávaného cíle (NID, resp. NID_{CC} - viz kapitolu 3) a identifikátor komunikace (CID).

- **Tabulky RID_SID** - sekce `==> ProbeID->RID->SID TABLE <==` - implementačně řešenou jako slovník, kde klíčem je identifikátor sondy (ProbeID) a hodnotou opět slovník (RID → SID).

- **Předpočítané rychlé vyhledávací tabulku** - sekce `==> FAST LOOKUP TABLE <==`, představující opět dva vnořené slovníky. Ve vnějším slovníku je klíčem identifikátor sondy (ProbeID), ve vnitřním slovníku pak identifikátor zachyceného paketu RID. Hodnotou je potom skutečně trojice zahrnující PCAPFilewriter a unixový čas začátku a konce odposlechu.

Obrázky 7.8 a 7.9 jsou však spíše ilustrativní a jejich cílem je přiblížit vnitřní funcionalitu celého systému - tedy ukázat, že koncept popsaný v kapitole 6 skutečně funguje v reálné implementaci.

```
============================================================
============= MF FACADE - PRINTING ALL TABLES =============
============================================================
==> LIID TABLE <===
LIID Table Content:
================================================================================
LIID            > Start                          | End                  | Intercept CC?
--------------------------------------------------------------------------------
'zz_li_other_1' > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | False
'zz_li_main'    > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | True
'zz_li_other_3' > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | False
'zz_li_other_2' > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | False
'zz_li_other_5' > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | False
'zz_li_other_4' > '03-05-2014 00:00:00'          | '19-01-2038 04:14:07' | False
==> SID TABLE <===
SID Table Content:
================================================================================
SID                     > LIID  | NID   | CID
--------------------------------------------------------------------------------
0                       zz_li_other_2   |IPv4 8.8.8.8   | CID('BUT', '8.8.8.8', '1', 'CZ')
0                       zz_li_other_1   |IPv4 8.8.8.8   | CID('BUT', '8.8.8.8', '1', 'CZ')
0                       zz_li_other_4   |IPv4 8.8.8.8   | CID('BUT', '8.8.8.8', '1', 'CZ')
0                       zz_li_main      |IPv4 8.8.8.8   | CID('BUT', '8.8.8.8', '1', 'CZ')
0                       zz_li_other_3   |IPv4 8.8.8.8   | CID('BUT', '8.8.8.8', '1', 'CZ')
==> ProbeID->RID->SID TABLE <===
{1: {0: 0}}
==> FAST LOOKUP TABLE <===
{1: {0: {(<modules.mf.cy_pcap_writer.CyPcapWriter object at 0x95b6c98>, 1399068000.0, 2147483647.0),
(None, 1399068000.0, 2147483647.0)}}}
============================================================
```

Obrázek 7.9: Ukázka ladícího výpisu interních tabulek bloku MF&CCTF

7.4.2 Dosažené výsledky

Kompletní výsledky měření výkonu v rámci všech experimentů jsou k dispozici na přiloženém CD (viz přílohu A).

Z hlediska experimentů je pro účely zjištění efektivity optimalizace podstatný celkový kumulativní čas strávený ve funkci `processMessageINI3()` - tedy čas, kdy zrovna „nečekáme na paket“. Výsledky měření tohoto času v rámci všech experimentů jsou znázorněny v tabulce 7.2.

První sloupec tabulky značí počet odposlechů evidovaných v systému v době měření. Druhý (resp. třetí) sloupec ukazuje celkový kumulativní čas strávený prováděním funkce `processMessageINI3()`, tedy čas od doby přijetí paketu z rozhraní INI3 až po jeho zápis do výstupního PCAP souboru (včetně) - před optimalizací (resp. po optimalizaci). Poslední sloupec pak ukazuje dosažené zrychlení díky provedené optimalizaci.

V rámci každého z experimentů je na konci každého z 5-ti měření (1 řádek = 1 měření) navíc přidán řádek udávající průměrné hodnoty časů i zrychlení pro daný dílčí experiment.

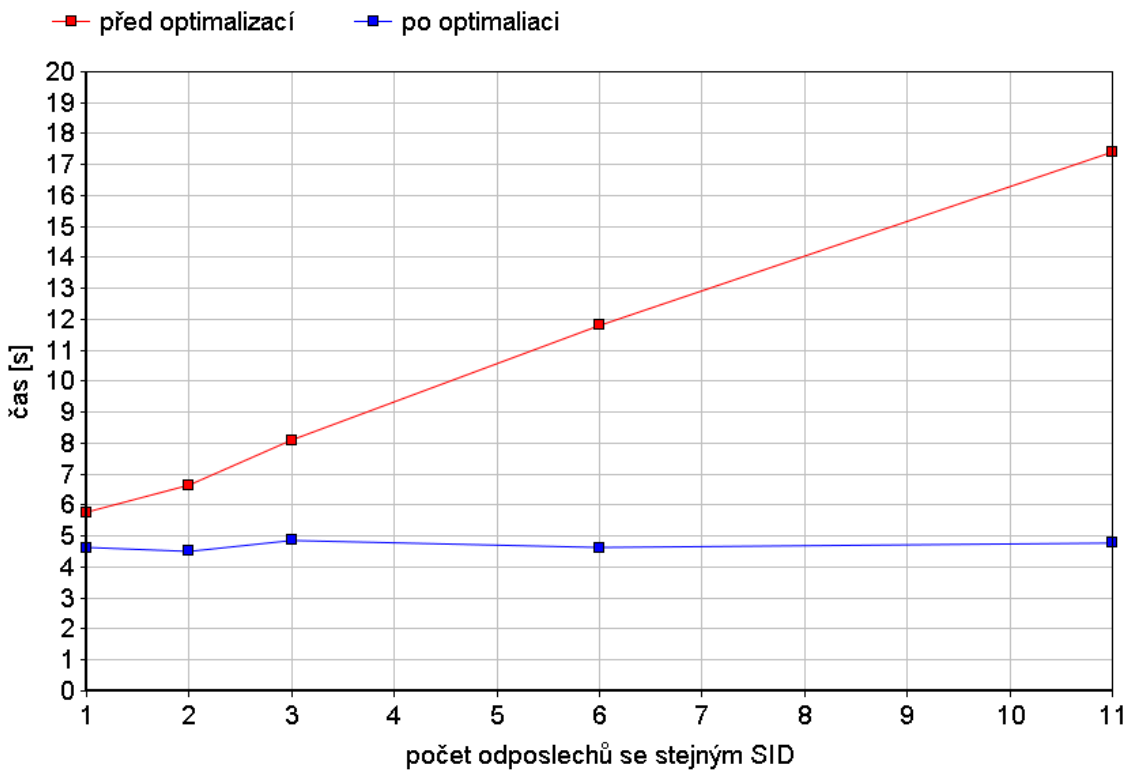
Z tabulky je zřejmé, že i v nejobecnějším případě (1 odposlech) jsme dosáhli zrychlení v průměru 21,27 %. Experimentálně tedy bylo dokázano tvrzení, že optimalizace popsaná v kapitole 6 má význam i v případě jednoho odposlechu na daný cíl.

U dalších experimentů vidíme, že s rostoucím množstvím odposlechů se stejným SID v pů-

počet odposlechů	Kumulativní čas strávený processMessageINI3()		
	před optimalizací	po optimalizaci	zrychlení
1	5,570 s	4,223 s	24,18 %
	5,624 s	4,417 s	21,46 %
	5,689 s	4,726 s	16,93 %
	5,981 s	4,922 s	16,54 %
	6,009 s	4,899 s	27,24 %
průměr	5,775 s	4,637 s	21,27 %
1 + 1 se stejným SID	6,534 s	4,276 s	34,56 %
	6,530 s	4,628 s	29,12 %
	6,940 s	4,598 s	33,74 %
	6,631 s	4,338 s	34,58 %
	6,626 s	4,698 s	29,09 %
průměr	6,652 s	4,508 s	32,22 %
1 + 2 se stejným SID	7,837 s	4,660 s	40,54 %
	8,204 s	4,693 s	42,80 %
	8,060 s	5,140 s	36,23 %
	7,874 s	5,227 s	33,62 %
	8,558 s	4,586 s	46,41 %
průměr	8,107 s	4,861 s	39,92 %
1 + 5 se stejným SID	11,577 s	4,892 s	57,74 %
	11,832 s	4,516 s	61,83 %
	12,121 s	4,585 s	62,17 %
	11,561 s	4,895 s	57,66 %
	11,936 s	4,285 s	64,10 %
průměr	11,805 s	4,635 s	60,70 %
1 + 10 se stejným SID	17,032 s	4,474 s	73,73 %
	17,434 s	4,662 s	73,26 %
	17,216 s	4,510 s	73,80 %
	17,594 s	4,490 s	74,48 %
	17,769 s	5,749 s	67,65 %
průměr	17,409 s	4,777 s	72,58 %

Tabulka 7.2: Výsledky měření výkonu před a po optimalizaci Mediační funkce

vodní implementaci také výrazně roste doba zpracování. Zatímco v optimalizované verzi není tento efekt znatelný. Čas zpracování paketů v optimalizované verzi je u všech experimentů velmi podobný: průměrně 4.684 s s mediánem v hodnotě 4.613 s. Tento pozitivní efekt lze vidět i na grafu 7.10, kde jsou znázorněny průměrné časy zpracování paketů funkcí `processMessageINI3()` před a po optimalizaci. Zatímco červená křivka (čas před optimalizací) má rostoucí trend, modrá (čas po optimalizaci) balancuje v intervalu od 4,223 s do 5,749 s.



Obrázek 7.10: Průměrný čas zpracování paketů v rámci jednotlivých experimentů.

7.4.3 Závěr měření

Cílem této sady experimentů bylo zjistit efektivitu optimalizace navržené v kapitole 6. V nejobecnější situaci (jeden odposlech na daný cíl) bylo dosaženo zrychlení v průměru 21,27 %. Bylo také experimentálně dokázáno, že při stoupajícím počtu překrývajících se odposlechů rapidně stoupá i efektivita provedené optimalizace. Na základě výsledků provedených měření považuji optimalizaci bloku MF&CCTF za úspěšnou.

Kapitola 8

Závěr

Systém pro zákonné odposlechy (*Lawful Interception System* - LIS) je nástroj, který umožňuje oprávněným orgánům sledovat komunikaci podezřelých subjektů v počítačové, či telefonní síti [7]. V rámci projektu Moderní prostředky pro boj s kybernetickou kriminalitou na internetu nové generace (Sec6Net) byl vyvinut prototyp systému LIS - *Sec6Net Lawful Interception System* - SLIS.

Dosavadní implementace prototypu se však vyznačovala řadou nedostatků. Nebylo možné automaticky detekovat identitu na základě aplikačního identifikátoru (např. XMPP loginu [21]). Nebylo možné selektivně zachytit pouze zájmová data v podobě komunikace na aplikační nebo transportní vrstvě [14], ani stanovit rozsah odposlechu - co chceme odposlouchávat a co nikoli. Dalším problémem byla malá propustnost z hlediska rychlosti pracování paketů z odposlouchávané linky.

Implementované rozšíření systému SLIS umožňuje oprávněným orgánům lépe a detailněji specifikovat cíl odposlechu. Podpora aplikačních protokolů umožňuje cílit odposlech na konkrétní aplikační identifikátor a na základě něj automaticky detekovat identitu cíle. Dalším přínosem je možnost z odposlouchávané linky vybrat pouze konkrétní zájmovou komunikaci na úrovni aplikační či transportní vrstvy. Do zavedení tohoto rozšíření bylo nutné zachytávat veškeré pakety s danou zdrojovou/cílovou IP adresou a další filtrace byla až v režii oprávněných orgánů. Tento přístup má kromě usnadnění práce s odposlouchávanými daty ze strany LEA také pozitivní dopad na výkonnost - pokud jsme schopni již na úrovni bloku CC-IIF vybrat např. konkrétní TCP spojení, můžeme tím výrazně zredukovat počet paketů nutných ke zpracování v bloku MF&CCTF.

Nově implementované jádro IRI-IIF založené na grafové reprezentaci přineslo jednak konfigurovatelné úrovně odposlechů umožňující přesněji specifikovat jeho rozsah, jednak usnadnilo určování cílového NID_{CC} pro vstupní NID. V neposlení řadě umožňuje IRI-IIF vizualizovat graf NID přes webové rozhraní systému SLIS. Toto umožní oblushujícímu operátorovi vytvořit si lepší představu o aktuálním stavu sítě.

Díky důkladné analýze a profilaci výkonnosti byla nalezena kritická místa v bloku MF&CCTF. Následně byl navržen a realizován způsob optimalizace, který značným způsobem urychlil zpracování paketů systémem SLIS. Efekt optimalizace byl tím vyšší, čím více odpolechů na stejný cíl bylo v systému evidováno.

Během této analýzy však byla objevena řada dalších možností zvýšení propustnosti systému. Z těchto důvodů bych chtěl v rámci své další práce implementovat možnost paralelního zápisu do více PCAP souborů z různých vláken, který ani současná implementace dosud neumožňuje. Dále bych se chtěl zaměřit na optimalizaci rozhraní CCCI a INI3 a další

zdokonalování systému.

V rámci projektu Sec6Net byl také sepsán článek na 6. ročník mezinárodní konference ICDF2C[1], která se soustřeďuje na digitální forenzní analýzu a kybernetickou bezpečnost. Článek má název *On Identities in Modern Networks* a pojednává o problematice detekce identity v počítačových sítích. Jeho obsah vychází také z poznaků získaných při psaní této práce.

[1] http://d-forensics.org/2014/show/home

Literatura

[1] Baker, F.; Foster, B.; Sharp, C.: *RFC 3924: Cisco Architecture for Lawful Intercept in IP Networks*. Říjen 2004.

[2] Behnel, S.; Bradshaw, R.; Seljebotn, D. S.; aj.: *Cython 0.19.2 - dokumentace*. 2012. URL http://docs.cython.org/

[3] Droms, R.: *RFC 2131: Dynamic Host Configuration Protocol*. Březen 1997.

[4] Droms, R. et al.: *RFC 3315: Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. Červenec 2003.

[5] European Telecommunications Standards Institute: *ETR 331: Security Techniques Advisory Group (STAG); Definition of user requirements for lawful interception of telecommunications; Requirements of the law enforcement agencies*. Prosinec 1996.

[6] European Telecommunications Standards Institute: *ES 201 158 : Telecommunications security; Lawful Interception (LI); Requirements for network functions*. Květen 1998, verze 1.1.2.

[7] European Telecommunications Standards Institute: *ETSI TR 101 943: Telecommunications security; Lawful Interception (LI); Concepts of Interception in a generic Network Architecture*. Červenec 2001, verze 1.1.1.

[8] European Telecommunications Standards Institute: *ETSI ES 201 158; Telecommunications security; Lawful Interception (LI); Requirements for network functions*. Duben 2002, verze 1.2.1.

[9] European Telecommunications Standards Institute: *ETSI TS 102 232; Telecommunications security; Lawful Interception (LI); Handover specification for IP delivery*. Únor 2004, verze 1.1.1.

[10] European Telecommunications Standards Institute: *ETSI TS 102 232-3; Lawful Interception (LI); Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 3: Service-specific details for internet access services*. Leden 2009, verze 2.2.1.

[11] European Telecommunications Standards Institute: *ETSI TS 101 671; Lawful Interception (LI); Handover interface for the lawful interception of telecommunications traffic*. Srpen 2010, verze 3.6.1.

[12] European Telecommunications Standards Institute: *ETSI TS 102 232-1; Lawful Interception (LI); Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 1: Handover specification for IP delivery*. Srpen 2010, verze 2.5.1.

[13] Gamma, E.; Helm, R.; Johnson, R.; aj.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995, iSBN 0-201-56882-9.

[14] International Organization for Standardization: *ISO/IEC international standard 7498-1:1994 Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. 1994.

[15] International Organization for Standardization: *International Standard ISO 3166-1, Codes for the representation of names of countries and their subdivisions - Part 1: Country codes, ISO 3166-1: 2006 (E/F)*. 2006.

[16] Martínek, T.; Kramoliš, P.; Holkovič, M.; aj.: Dynamická identifikace uživatelů v prostředí sítí IPv4 a IPv6. Technická zpráva, 2012.
URL http://www.fit.vutbr.cz/research/view_pub.php.cs?id=10210

[17] Oikarinen, J.; Reed, D.: *RFC 1459: Internet Relay Chat Protocol*. Květen 1993.

[18] Polčák, L.; Kramoliš, P.; Kajan, M.; aj.: Architektura systému pro zákonné odposlechy. Technická zpráva, 2011.
URL http://www.fit.vutbr.cz/research/view_pub.php.cs?id=9829

[19] Python Software Foundation: *Dokumentace k jazyku Python*. 1990-2013.
URL http://www.python.org/doc/

[20] Rigney, C.; et al., S. W.: *RFC 2865: Remote Authentication Dial In User Service (RADIUS)*. Červen 2000.

[21] Saint-Andre, P.: *RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core*. Březen 2011.

[22] Sbírka zákonů č. 259/2010: *Zákon o elektronických komunikacích*. 2010, hlava V, díl 1.

[23] Simpson, W.: *RFC 1661: The Point-to-Point Protocol (PPP)*. Červenec 1994.

[24] The Council of European Union: *Council Resolution of 17 January 1995 on the lawful interception of telecommunications*. Listopad 1996, official Journal C 329, 04/11/1996 str. 1-6.

Příloha A

Obsah CD

Na přiloženém CD se nachází:

- tento dokument v interaktivní podobě ve formátu .pdf,
- verze systému SLIS před provedenými úpravami,
- verze systému SLIS po provedených úpravách,
- instalační manuál k systému SLIS,
- referenční PCAP soubory použité při měření,
- výsledky všech profilací při měření výkonnosti bloku MF&CCTF.

Popis jednotlivých položek a strukturu adresářů poskytuje soubor `README.txt`, který je umístěn přímo v kořenovém adresáři přiloženého CD.

Příloha B

Požadavky na prototyp systému pro sběr dat

Dle normy ETSI ES 201 158 [8] by měl systém podporovat až 100 LEMF (viz sekci 2.3) a být schopen zasílat data týkající se jednoho odposlechu alespoň do třem LEMF. Další požadavky Ministerstva vnitra ČR jsou následující:

- Přípustná jsou řešení s agregací i bez agregace datového toku.
- Výstup dat na rozhraních HI1, HI2 a HI3 (viz sekci 2.3) musí být v souladu s aktuálními doporučeními ETSI TS 102 232 [9].
- Systém musí být schopen zachytit provoz DSL na protokolech IPv4 i IPv6 podle aktivovaných filtračních kriterií a předat jej na výstup tak, aby všechny pakety byly seřazeny v pořadí, ve kterém byly přenášeny sítí, a přitom nedošlo ke ztrátě žádného z nich.
- Systém musí být schopen zpracovávat data i z plně vytížené 10 Gbps linky bez ztráty paketů.
- Všechny sondy systému musí mít společné řízení.
- Systém musí být schopen zpracovat souhrnně alespoň 500 Mbps filtrovaného provozu.
- Časové značky u každého zachyceného paketu musí přesně odpovídat okamžiku jeho zachycení (přednost časových značek s tolerancí nejvýše 1 ms, přičemž společný systémový zdroj časových značek se může od mezinárodního časového standardu UTC+1 odchylovat nejvýše o 1 s).
- Systém musí být schopen aktivovat jako filtrační kritérium alespoň následující identifikátory: telefonní číslo, login, pevná IPv4 adresa, pevná IPv6 adresa, rozsah IP adres.
- Systém musí být schopen ukončit filtrační kritéria aktivovaná s časovou platností automaticky po jejich vypršení.
- Systém bude předávat informace o aktivaci/deaktivaci/modifikaci filtračního kritéria rozhraním HI1.
- Systém bude předávat informace o dynamicky přidělené IP adrese rozhraním HI1.

- Systém musí být schopen zahájit zachytávání zájmového provozu bezprostředně po aktivaci filtračního kritéria, a to i v případě, že IP adresa je dynamicky přidělována Radius serverem (telefonní číslo, login).
- Systém musí být schopen nezávisle sledovat totožné zájmové uživatelské adresy pro minimálně 5 LEA (viz sekci 2.3).
- Systému bude možné zadat minimálně 1000 filtračních kritérií (minimálně 200 na LEA).
- Bude existovat možnost budoucího rozšiřování systému bez rizika zmaření případně již realizovaných investic do technologie dodané předkladatelem nabídky.

Příloha C

Příklad tabulky NID v bloku IRI-IIF

ID modulu	MAC	DUID	IPv4	IPv6	PPP_L	$RADIUS_L$	...
PPPoE	00:25:90:0f:81:37				radek		
RADIUS	08:00:69:02:01:fc					standa	
DHCP	00:13:a9:a7:ef:4b		10.1.0.6				
RADIUS	00:80:c7:0f:4b:22		10.2.0.4			tomas	
SLAAC	00:80:c7:e4:81:1f			fd02..			
DHCPv6		0003..		fd02..			

Tabulka C.1: Příklad tabulky NID v bloku IRI-IIF

Každý záznam v tabulce je výsledkem skutečnosti, že určitý modul pomocí *IRI BEGIN / IRI CONTINUE* ohlásil určitou událost. Záznamy v tabulce C.1 byly vytvořeny na základě následujících hlášení modulů:

1. Modul PPPoE: počítači s MAC adresou 00:25:90:0f:81:37 byla udělena autorizace na základě PPP loginu „radek“ a hesla.
2. Modul RADIUS: počítači s MAC adresou 08:00:69:02:01:fc byla udělena autorizace na základě RADIUS loginu „standa“ a hesla.
3. Modul DHCP: počítači s MAC adresou 00:13:a9:a7:ef:4b byla přidělena IPv4 adresa 10.1.0.6.
4. Modul RADIUS: počítači s MAC adresou 00:80:c7:0f:4b:22 byla udělena autorizace na základě RADIUS loginu „tom“ a hesla a současně mu byla také přidělena IPv4 adresa 10.2.0.4.
5. Modul SLAAC: počítači s MAC adresou 00:80:c7:e4:81:1f byla přidělena IPv6 adresa fd02...
6. Modul DHCPv6: počítači s DUID identifikátorem 0003.. byla přidělena IPv6 adresa Ifd02...

Příloha D

Mapování LIID na SID v bloku MF&CCTF

V této příloze bude vysvětleno mapování LIID na SID. Pro ilustraci uvažujme následující modelovou situaci:

1. Do systému SLIS jsou vloženy dva požadavky na odposlech: X a Y. Požadavky odpovídají odposlechům tabulce 3.4.
2. Dne 1.1.2011 zahájí AF odposlech požadavku X a předá příslušné zprávy blokům IRI-IIF a MF&CCTF.
3. Sledováním protokolu DHCP detekuje blok IRI-IIF, že byla počítači s MAC adresou 00:11:22:aa:bb:cc přiřazena IPv4 adresa 10.0.0.1. Dále bylo zjištěno, že počítač začal používat IPv6 adresu 2001:db8::5 (např. sledováním protokolu DHCPv6 nebo SLAAC).
4. IRI-IIF informuje MF&CCTF, že v odposlechu X došlo k přiřazení IPv4 adresy 10.0.0.1 a že bude tato komunikace označována CID = (LEA_1, MAC: 00:11:22:aa:bb:cc, 1, CZ). CID představuje identifikátor komunikace a byl popsán v sekci 3.4. Třetí hodnota zleva představuje CIN, který je zde roven 1. Toto přiřazení provedl blok IRI-IIF (viz sekci 3.6).
5. MF&CCTF přiřadí tomuto odposlechu SID 1 a uloží tuto informaci do *tabulky SID*.
6. IRI-IIF ohlásí MF&CCTF, že v rámci odposlechu s X došlo k přiřazení IPv6 adresy 2001:db8::5 a že bude používán CID = (LEA_1 , MAC:00:11:22:aa:bb:cc, 2, CZ).
7. Jelikož nedochází ke shodě s některou z již existujících IP adres v tabulce SID, přiřadí MF&CCTF tomuto odposlechu SID 2 a uloží tuto informaci do *tabulky SID*.
8. Dne 2.1.2011 zahájí AF odposlech požadavku Y a předá příslušné zprávy blokům IRI-IIF a MF&CCTF.
9. IRI-IIF může okamžitě ohlásit odposlech IPv4 adresy 10.0.0.1 s CID např. (LEA_2, IPv4: 10.0.0.1/32, 5, CZ).
10. Jelikož dochází ke shodě s již odposlouchávanou IPv4 adresou z požadavku X, přidělí MF&CCTF pro tento odposlech SID 1 a uloží tuto informaci do *tabulky SID*.

Po provedení předchozích kroků bude obsah *tabulky SID* odpovídat tabulce D.1. První sloupec představuje CID přiřazený blokem MF&CCTF. Druhý sloupec je NID_{CC}, který byl nalezen blokem IRI-IIF. Dále je v tabulce LIID osposlechu a CID v rámci něj probíhající komunikace. Tabulka ukazuje názorný příklad situace, kdy jeden NID_{CC} patří do rozsahu dvou různých odposlechů (X a Y).

SID	NID_{CC}	LIID	CID
1	IPv4: 10.0.0.1/32	X Y	(LEA_1, MAC: 00:11:22:aa:bb:cc, 1, CZ) (LEA_2, IPv4: 10.0.0.1/32, 5, CZ)
2	IPv6: 2001:db8::5/128	X	(LEA_1, MAC: 00:11:22:aa:bb:cc, 2, CZ)

Tabulka D.1: Příklad tabulky SID v bloku MF&CCTF

Tabulka D.2 pak ilustruje transformaci NID zadaných v požadavcích na konkrétní NID_{CC}, které lze odposlouchávat. Tato ilustrace představuje v zásadě spojení dvou předchozích tabulek. Vidíme, že např. k odposlechu X na MAC adresu MAC: 00:11:22:aa:bb:cc byly nalezeny dva související NID_{CC}, z nichž každý pochází z jiné komunikace (odlišný CID).

LIID	NID z požadavku	NID_{CC}	SID	CID
X	MAC: 00:11:22:aa:bb:cc	IPv4: 10.0.0.1/32 IPv6: 2001:db8::5	1 2	(LEA_1, MAC: 00:11:22:aa:bb:cc, 1, CZ) (LEA_1, MAC: 00:11:22:aa:bb:cc, 2, CZ)
Y	IPv4: 10.0.0.1	IPv4: 10.0.0.1	1	(LEA_2, IPv4: 10.0.0.1/32, 5, CZ)

Tabulka D.2: Transformace NID z požadavku na konkrétní NID_{CC}, které lze odposlouchávat

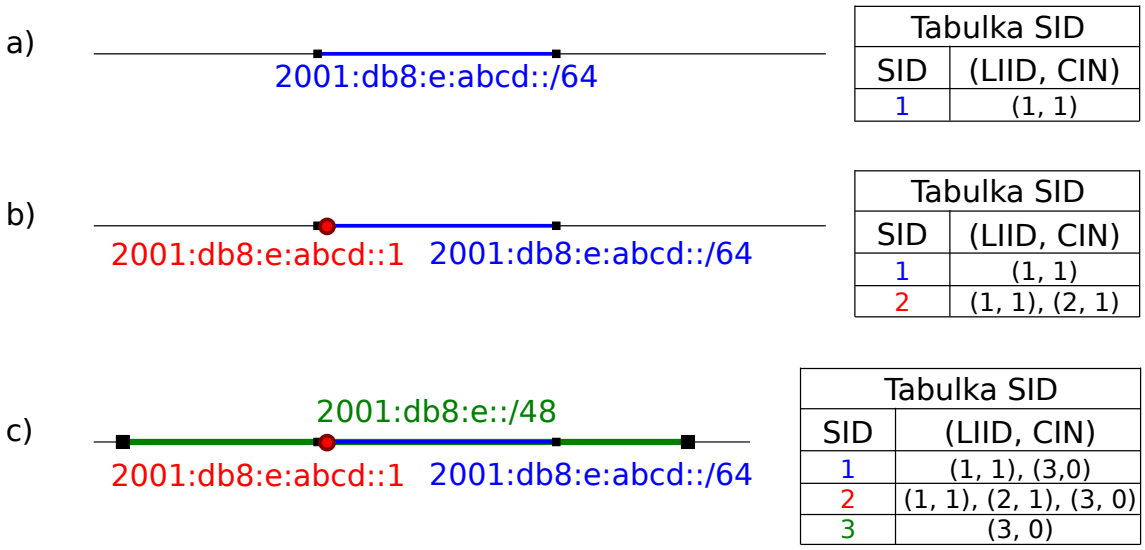
V případě odposlechů rozsahu IP adres přidělování SID funguje způsobem znázorněným na obrázku D.1

V ukázkové situaci byly postupně přidány tři odposlechy. Dle pravidel v kapitole 3, sekci 3.7 byly provedeny následující kroky:

1. Nejdříve byl aktivován odposlech s LIID 1, jehož cílem je rozsah *2001:db8:e:abcd::/64*. Uvažujme, že blok IRI-IIF tomuto rozsahu adres v rámci LIID 1 přidělil CIN 1. Blok MF&CCTF v *tabulce SID* vytvoří nový záznam: SID 1 a do tabulky uloží, že data označená SID 1 se budou vztahovat k LIID 1 (pro CIN 1), jak je znázorněno v části a) obrázku D.1.

2. Následně je aktivován odposlech LIID 2, jehož cílem je IPv6 adresa *2001:db8:e:abcd::1*. Uvažujme, že blok IRI-IIF této IPv6 adrese rámci LIID 2 přidělil CIN 1. Blok MF & CCTF tedy vytvoří nový SID 2. Do */tabulky SID* uloží, že data označená SID 2 se budou vztahovat k odposlechu s LIID 1 (pro CIN 1) a také k odposlechu s LIID 2 (pro CIN 1). Situaci znázorňuje část b) obrázku D.1.

3. Poté je aktivován odposlech s LIID 3, jehož cílem je celý rozsah *2001:db8:e::/48*. Uvažujme, že blok IRI-IIF tomuto rozsahu adres v rámci LIID 3 přidělil CIN 0. Blok



Obrázek D.1: Ukázka mapování LIID na SID pro rozsahy IP adres

MF&CCTF tedy vytvoří nový SID 3. Do *tabulky SID* uloží, že data označená SID 3 se budou vztahovat k osdposlechu s LIID 3 (pro CIN 0). Zároveň však MF&CCTF do tabulky doplní, že data označená SID 1 i SID 2 se budou vztahovat také k odposlechu s LIID 3 (pro CIN 0). Situaci znázorňuje část c) obrázku D.1.

Mapování LIID na SID pro rozsahy IP adres

Předchozí příklad ukazoval překryv konkrétních IP adres. V reálné situtaci se však mohou překrývat nejen konkrétní IP adresy, ale také rozsahy IP adres. Může se stát, že jeden odposlech zahrnuje takový rozsah IP adres, který je podrozsahem jiného odposlechu. Při požadavku na odposlech nového rozsahu R_N se tedz kontroluje, zda již v *tabulce SID* neexistuje odpovídající větší nebo menší rozsah a podle potřeby se upraví informace v ní uložené. Blok MF&CCTF v takovém případě postupuje dle následujících pravidel:

- Pokud není žádná adresa z rozsahu adres R_N odposlouchávána, pak je vygenerován nový SID a odposlech je uložen do tabulky SID.
- Pokud je R_N vlastní podmnožinou jiného rozsahu adres, např. rozsahu R_V (tzn. R_N obsahuje nějakou část adres obsažených v R_V, ale ne všechny), pak se vygeneruje nový SID a do tabulky SID se k nově přidávanému odposlechu uloží i odposlechy vztahující se k R_V .
- Pokud se již odposlouchává stejný rozsah adres, pak se pouze nový odposlech uloží do tabulky SID ke stávajícím odposlechům pro daný rozsah a všechny podrozsahy adres.
- Při existenci odposlechů pro menší rozsahy adres se vytvoří nový SID pro nově vkládaný rozsah a zároveň se informace o nově přidávaném odposlechu uloží i ke všem odposlouchávaným podrozsahům adres.