VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTITA V TUNELOVANÝCH A PŘEKLÁDANÝCH SÍTÍCH

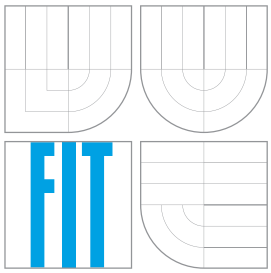
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL ŠEPTUN

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTITA V TUNELOVANÝCH A PŘEKLÁDANÝCH SÍTÍCH

IDENTITIES IN TUNNELED NETWORKS AND DURING NETWORK ADDRESS TRANSLATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL ŠEPTUN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LIBOR POLČÁK

BRNO 2015

Zadání diplomové práce/13207/2014/xseptu00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2014/2015

Zadání diplomové práce

Řešitel: **Šeptun Michal, Bc.**
Obor: Počítačové sítě a komunikace
Téma: **Identita v tunelovaných a překládaných sítích**
Identities in Tunelled Networks and during Network Address Translation
Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se se systémem pro zákonné odposlechy vyvíjeném v rámci projektu Sec6Net.
2. Seznamte se s principy tunelování síťového provozu a překladu adres.
3. Navrhněte rozšíření mechanismu pro monitorování identity vyvíjeném v projektu Sec6Net o vybrané mechanismy nastudované v předchozím bodě.
4. Návrh implementujte.
5. Otestujte implementaci.
6. Zhodnoťte dosažené výsledky.

Literatura:

- POLČÁK Libor, HRANICKÝ Radek a MARTÍNEK Tomáš. On Identities in Modern Networks. *The Journal of Digital Forensics, Security and Law*. 2014, roč. 2014, č. 2, s. 9-22. ISSN 1558-7215.
- POLČÁK Libor. Challenges in Identification in Future Computer Networks. In: *ICETE 2014 Doctoral Consortium*. Wien: SciTePress - Science and Technology Publications, 2014, s. 15-24.
- SATRAPA Pavel. Internetový protokol verze 6.Edice CZ.NIC 2011.
- A další podle dohody s vedoucím.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Polčák Libor, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2014
Datum odevzdání: 27. května 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

V této práci se seznámíme s návrhem a implementací rozšíření části systému pro zákonné odposlechy. Systém je vyvíjen v projektu Sec6Net na FIT VUT v Brně a poskytuje platformu pro výzkumnou činnost v problematice určovaní identity v počítačových sítích. Rozšířena bude část, která má za úkol sledování změn identity uživatele tak, aby byl systém schopen určovat identitu i v tunelovaných a překládaných sítích. Poznáme problémy, které se vyskytly v průběhu implementace a jejich řešení. Jsou jde popsány mechanismy pro tunelování sítí, hlavně virtuální privátní sítě a přechodové mechanismy pro IPv6, překlad IP adres NAT a jeho varianty. Na závěr jsou uvedeny testy jednotlivých modulů.

Abstract

This thesis introduces the design and implementation of the extension of the system for lawful interception. The system is developed as a part of the Sec6Net project at FIT BUT and provides a platform for research activities in determining identities in computer networks. Parts which has the task of monitoring changes in a user's identity will be extended, so that the system is able to determine the identity even in the tunneled and translated networks. It describes the problems encountered during implementation and their solutions. There are described mechanisms for tunneling networks, mainly virtual private networks and transition mechanisms for IPv6, IP addresses and NAT variants. In the end the tests of the individual modules are described.

Klíčová slova

Sec6Net, zákonné odposlechy, identita, síťové tunelování, VPN, přechodové mechanismy IPv6, NAT

Keywords

Sec6Net, lawful interception, identities, tunelled networks, VPN, IPv6 transition mechanisms, NAT

Citace

Michal Šeptun: Identita v tunelovaných a překládaných sítích, diplomová práce, Brno, FIT VUT v Brně, 2015

Identita v tunelovaných a překládaných sítích

Prohlášení

Prohlašuji, že jsem tento diplomový projekt vypracoval samostatně pod vedením pana Ing. Libora Polčáka
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Šeptun
27. května 2015

Poděkování

Tímto bych rád poděkoval svému vedoucímu Ing. Liboru Polčákovi za vedení a pomoc při psaní této práce.

Obsah

Kapitola 1

Úvod

V dnešní době má k internetu a službám s ním spojeným přístup téměř kdokoli. Většina uživatelů využívá služby v souladu s tím, jak byly navrženy a nijak neohrožují ostatní. Jako v každé oblasti se ovšem najde pár jedinců, kteří chtějí věci využít pro svůj prospěch a obohacení, a to i nelegálním způsobem. Internet využívají zkušení hackeři, kteří přes něj útočí na servery státních či soukromých organizací, jako jsou banky, úřady, databáze s tisíci položek citlivých údajů. Pomocí sítě komunikují i pachatelé různé jiné trestné činnosti a v některých případech je nutné zaznamenávat jejich aktivity.

Pro monitorování nezákonné činnosti slouží systémy pro zákonné odposlechy. Nutnost existence zákonných odposlechů vyplývá z některých zákonů České republiky, např. zákon 127/2005 Sb. ve znění pozdějších předpisů, o elektronických komunikacích [13], a ustanovení Evropské unie [1].

S jedním takovým systémem vyvíjeném v projektu Sec6Net se seznámíme v části 2. Poznáme celkovou architekturu, komunikační rozhraní systému, jednotlivé části a jejich funkce. Blíže se zaměříme na část pro určování dynamické identity. Popíšeme jak komunikují jednotlivé moduly s jádrem a jaké zprávy jsou použity.

V kapitole 3 se seznámíme s dnešními technikami pro tunelování síťového provozu, obecným protokolem GRE, IPSec, protokoly a programy používanými pro vytvoření virtuálních privátních sítí a některými přechodovými mechanismy IPv6.

Jak funguje překlad IP adres a jaké jsou varianty se dozvíme v kapitole 4.

Hlavním záměrem této práce je rozšíření systému pro zákonné odposlechy o některé z uvedených mechanismů. Návrh rozšíření si popíšeme v části 5.

Vlastní implementace je pak prováděna pouze s části systému pro zákonné odposlechy. V kapitole 6 se dozvíme, jakým způsobem je část spravující identitu rozšířena, aby systém mohl určovat identitu v tunelovaných a překládaných sítích. Implementovaná funkcionalita se zaměřuje hlavně na tunelování pomocí VPN, jako například protokol pptp nebo openVPN, a na informace ze zařízení překládající síťové adresy získané pomocí Syslogu.

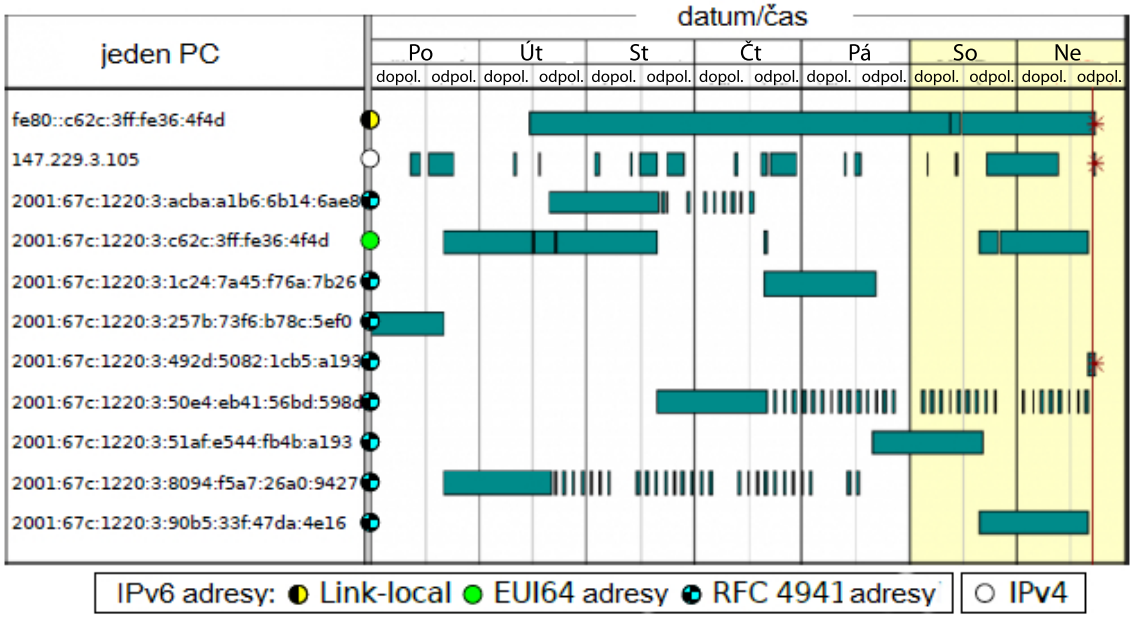
V kapitole 7 zaměřené na testování se seznámíme s prostředím GNS3, které je použito pro vytvoření jednoduché síťové topologie. Popíšeme testování jednotlivých modulů a výstupy v podobě grafů zobrazující propojení identifikátorů uživatele nebo počítače vyskytující se v počítačové síti.

Kapitola 2

Systém pro zákonné odposlechy

V této kapitole nejdříve obecně popíšeme systém pro zákonné odposlechy vyvíjený na FIT VUT v Brně a následně se blíže zaměříme na část sledující identitu uživatelů v síti, která může být určena různými identifikátory a může se dynamicky měnit. Architektura, jednotlivé moduly a jejich význam, rozhraní a některé používané identifikátory jsou popsány v kapitole 2.1. Funkce dynamické identity zajišťující hlavní část funkčnosti systému pro zákonné odposlechy je blíže popsána v kapitole 2.2.

Jak ukazuje obrázek 2.1 publikovaný v článku User Identification in IPv6 Network [9], jediný počítač může v průběhu týdne vystřídat několik IP adres, u IPv6 má síťové rozhraní zároveň více adres.

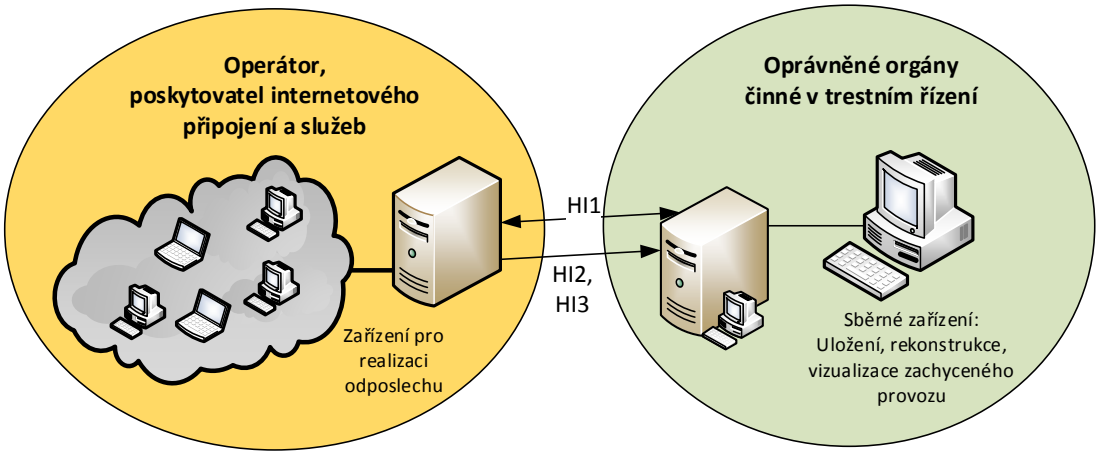


Obrázek 2.1: Různé IP adresy v průběhu týdne

Uživatel také může využívat v průběhu času více zařízení, různé služby a proto je důležité jednotlivé identity v počítačové síti zaznamenávat, spojovat a vytvářet tak ucelený obraz skutečnosti.

2.1 Architektura

V současné době existují dvě hlavní architektury systémů pro zákonné odposlechy, v USA využívaný standard J-STD-025 a ETSI [2] standard využívaný v Evropské unii. Dále jsou zde pak také komerční systémy, např. společnost Cisco kombinuje oba standardy.



Obrázek 2.2: Zařízení pro zákonné odposlechy

Systém vyvíjený na FIT si neklade za cíl vytvořit zcela novou architekturu nebo podporovat oba standardy, ale spíše poskytnout prostředí pro zkoumání problematiky zákonných odposlechů [10]. Sec6Net Lawful Interception System (dále jen SLIS) vychází z evropského standardu. Pro komunikaci mezi zařízením pro odposlech a zařízením pro sběr a vizualizaci dat, viz obrázek 2.2, používá tyto komunikační rozhraní:

HI1: přenáší požadavky na odposlech od oprávněných orgánů a případné dodatečné informace o odposlechu,

HI2: přenos informací (přidělení IP adresy, připojení-odpojení ze sítě, apod.) o sledovaných subjektech,

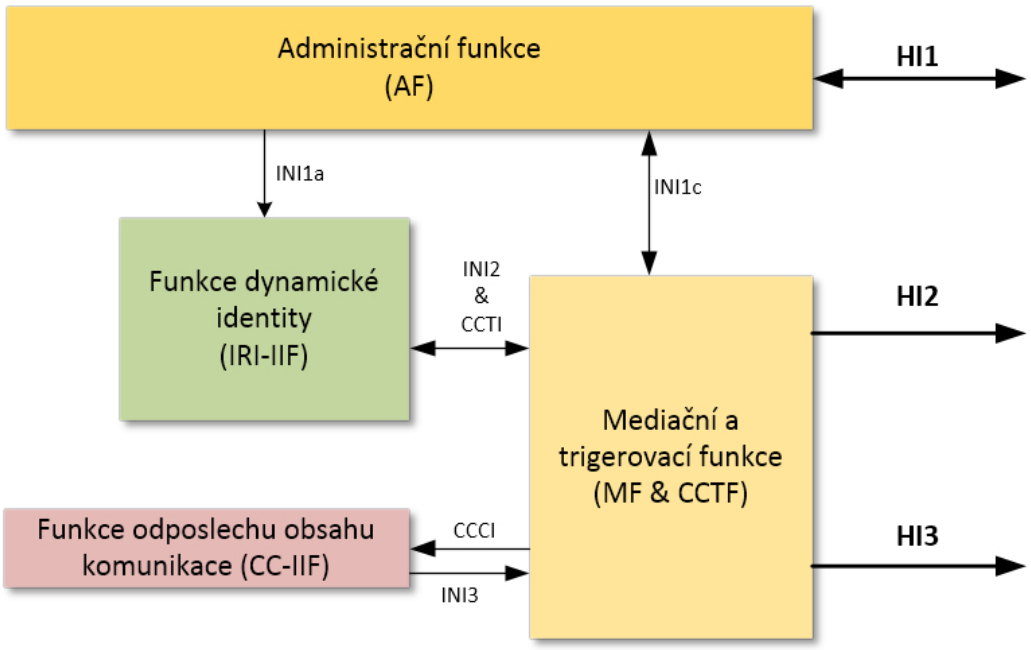
HI3: přenos obsahu komunikace sledovaného (data, obsah e-mailu, apod.).

Vlastní odposlech probíhá na straně poskytovatele internetového připojení a odposlouchávací zařízení je tedy zodpovědné za přijetí a ověření požadavku, sledování aktivit (připojení-odpojení, dynamická změna identity, viz část 2.2) a odesílání zachycených dat oprávněným orgánům.

Architektura zařízení pro realizaci odposlechu se skládá ze 4 základních částí, viz obrázek 2.3:

- *AF - Administrační funkce* se stará o přijetí a zpracování vstupních požadavků skrz rozhraní HI1. Provádí kontrolu správnosti vyplněných údajů a následné nastavení ostatních částí systému.
- *MF - Mediační funkce* zpracovává data právě probíhajících odposlechu. Data získaná od IRI-IIF a CC-IIF kombinuje a pomocí rozhraní HI2 a HI3 zasílá oprávněným orgánům. *CCTF - trigerovací funkce* má na starosti konfiguraci CC-IIF sond.

- *IRI-IIF - Funkce dynamické identity* udržuje informace o identitě uživatelů, která se může v průběhu času měnit.
- *CC-IIF - Funkce odposlechu obsahu komunikace* zajišťuje kopírování celého obsahu síťového provozu odposlouchávané osoby.



Obrázek 2.3: Architektura prototypu odposlouchávacího zařízení

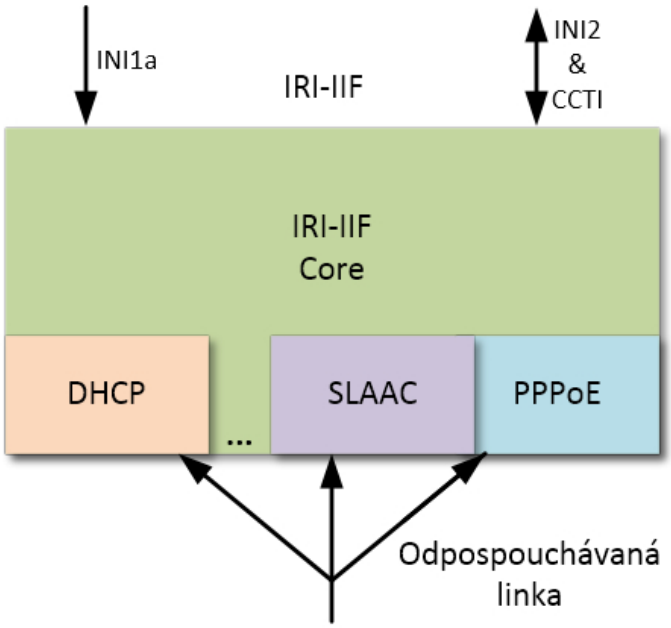
Pro komunikaci mezi výše popsanými částmi jsou využívána rozhraní viz obrázek 2.3. Jednotlivé části a rozhraní používají pro přenos informací následující identifikátory:

- *NID (Network IDentifier)*: identifikátor označující účastníky komunikace, spojení nebo konkrétní zařízení. Tento identifikátor může obsahovat MAC adresu, IP adresu, přihlašovací jméno atd.
- *HI1ID (Handover Interface 1 IDentifier)*: identifikace cíle odposlechu, tento identifikátor musí pověřený pracovník jednoznačně převést na jeden z NID identifikátorů. Převedení identifikátoru ve většině případů provede podle interní databáze obsahující seznam zákazníku (jména, rodná čísla, ...).
- *CID (Communication IDentifier)*: identifikátor sezení nebo komunikace v rámci jednoho odposlechu.
- *SID (System IDentifier)*: identifikátor množiny odposlechů udávaný 32-bitovým celým číslem.
- *LIID (Lawful Interception IDentifier)*: jednoznačný identifikátor odposlechu reprezentovaný řetězcem alfanumerických znaků. Všechna data na rozhraních HI2 a HI3 musí být tímto identifikátorem označeny.

2.2 Funkce dynamické identity (IRI-IIF)

Nyní si blíže popíšeme část systému pro zjišťování a udržování identity (IRI-IIF), která bude hlavní částí rozšíření systému pro zákonné odposlechy.

Pro možnost rozšiřování o další podporované protokoly je blok IRI-IIF navrhnut modulárně. Skládá se z jádra IRI-IIF, které zpracovává zprávy z jednotlivých modulů a případně posílá zprávy do mediační a trigerovací funkce, viz obr 2.4, a modulů hledajících různým způsobem identity vyskytující se v počítačové síti.



Obrázek 2.4: Modulární architektura IRI-IIF

V současné verzi systému jsou podporovány tyto protokoly a zdroje identity [11]: DHCP, RADIUS, PPPoE, DHCPv6, objevování sousedu (ND) včetně bezstavové autokonfigurace adres (SLAAC), Extensible Messaging and Presence Protocol (XMPP), Internet Relay Chat (IRC), Open System for Communication in Realtime (OSCAR), Yahoo! Messenger Protocol (YMSG), Simple Mail Transfer Protocol (SMTP), identifikace počítače pomocí odchylky v měření času a zjišťování identity z kontroléru SDN.

Každý modul starající se o nějaký z výše uvedených protokolů nebo zdrojů identity po zjištění události zasílá jádru zprávy o detekovaných změnách identity ve formátu: *(Jméno modulu, Časová značka, Typ zprávy, Popis zprávy, Seznam NIDů, jejichž vazeb se zpráva týká)*. Typy zpráv můžeme vidět v tabulce 2.1.

Celková zpráva pak může vypadat například takto:

```
('pptp', 1403819400.0, 'BEGIN', 'Klient se pripojil do VPN site',
[('PPTP', '1234'), ('MAC', '00:0d:10:11:8a:15'), ('PPTP Login', 'user'),
('IP', '192.168.100.6')])
```

Za seznamem NIDů ve zprávě mohou volitelně následovat další 2 seznamy. Pokud takovýto případ nastane význam seznamů je následující:

- 1. seznam: NIDy určené pro zpracování jádrem IRI-IIF k určování identity. Tyto NIDy jsou při odeslání zprávy typu end z jádra IRI-IIF odstraněny.

IRI zpráva	Popis
Begin	Oznamuje úspěšnou autentizaci nebo přidělení IP adresy
End	Ukončení období pro autentizaci nebo přidělení IP adresy
Continue	Obnova IP adresy
Report	Informativní zpráva

Tabulka 2.1: IRI zprávy předávané mezi moduly a jádrem bloku IRI-IIF

- 2. seznam: NIDy nejsou určeny pro zpracování, ale využijí se pouze pokud jádro generuje zprávy IRI.
- 3. seznam: NIDy určené pro zpracování jádrem IRI-IIF. Tyto identifikátory by však měly po odeslání zprávy end zůstat v jádru.

Po přijetí seznamu NIDů se v jádru IRI-IIF zpracují podle typu zprávy. Pokud je zpráva typu *BEGIN* přidá se hrana mezi vrcholy identifikátorů do grafu, případně jsou vytvořeny vrcholy nové. V případě, že je zpráva typu *END* hrany mezi vrcholy definovanými ve zpravě jsou zrušeny.

Každý NID patří do jedné z kategorií, která představuje rozdílné typy informací na různých vrstvách síťového modelu:

- typ A - Aplikační identifikátor, identifikátor aplikačních spojení,
  – 5-tice (IP klienta, IP serveru, port klienta, port serveru, typ transportního protokolu),
  – 3-jice (IP, port, typ transportního protokolu),
  – login IRC, název kanálu v rámci serveru IRC,
  – login login XMPP,
  – login YMSG,
  – login OSCAR,
  – login SIP,
  – e-mailová adresa,
- typ B - Adresa síťové vrstvy
  – IPv4 adresa,
  – IPv6 adresa,
- typ C - Adresa síťového rozhraní, nebo identifikátor konkrétního počítače,
  – MAC adresa,
  – DHCP client ID,
  – DHCPv6 DUID,
- typ D - Ostatní identifikátory (především pro autentizaci),
  – RADIUS login,
  – PPP login,
  – Číslo PPP sezení

Podle úrovně odposlechu se pak vyhledávají spojovací hrany mezi různými typy NIDů. Úrovně odposlechu jsou definovány 3 a to následovně. **I. úroveň** *Odposlech v rozsahu síťové adresy* - zachytává pouze data přímo spojená se zadanou IP adresou. **II. úroveň** *Odposlech v rozsahu rozhraní nebo počítače* - jsou zachytávána data jak s výskytem konkrétní adresy,

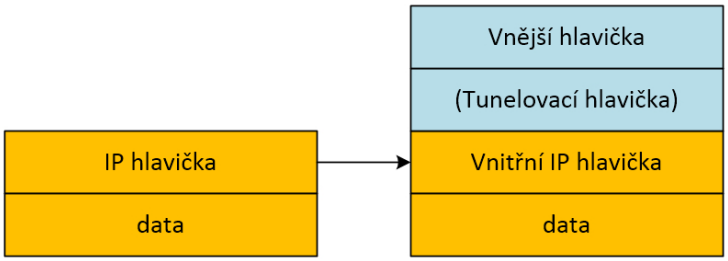
ale také data s jinými IP adresami, které náleží ke stejné MAC adrese. **III. úroveň** *Odposlech v rozsahu uživatele* - zachytávána je veškerá komunikace uživatele, data mohou být spojená s různými IP i MAC adresami, které spojuje například přihlašovací jméno.

Kapitola 3

Tunelování síťového provozu

V této kapitole si popíšeme principy některých tunelovacích mechanismů, které se v dnešní době více či méně používají. Tunelování síťového provozu lze využívat v mnoha případech, pro ochranu přenášených dat po nedůvěryhodné síti, v případě, že existující síť nepodporuje protokoly nebo služby, které chceme využít, přechodové mechanismy pro IPv6, viz kapitola 3.4, nebo například pro propojení poboček firmy pomocí VPN[1].

Tunelování je technika kdy je paket zapouzdřen do nového paketu [16].



Obrázek 3.1: Zapouzdření původního IP paketu

Tunelováním lze v podstatě přenášet jakýkoli protokol, např. IPX, AppleTalk, IPv4, IPv6.

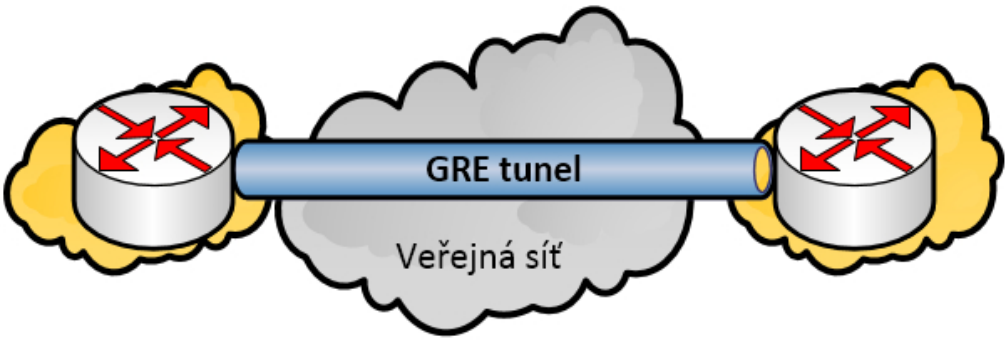
Pro tunelování-zapouzdření lze využít také různé protokoly. Využívá se například GRE, IPSec, L2TP, PPTP. Tunelovací protokol je pak přenášen po existující síti jako je Frame-relay nebo Ethernet.

Tunelovat lze pomocí tunelovacích protokolů, ale také bez nich. Pokud nějaký pomocný protokol využijeme a obalíme jím originální data, přináší nám to výhodu autentizace, šifrování a například několikanásobného tunelu mezi stejnými zařízeními, ale tyto vlastnosti mohou také přinést větší režii do celého přenosu. Tunelovat bez pomocných protokolů lze například pomocí IP-in-IP nebo IPv6-in-IPv4.

[1] VPN - Virtual Private Network

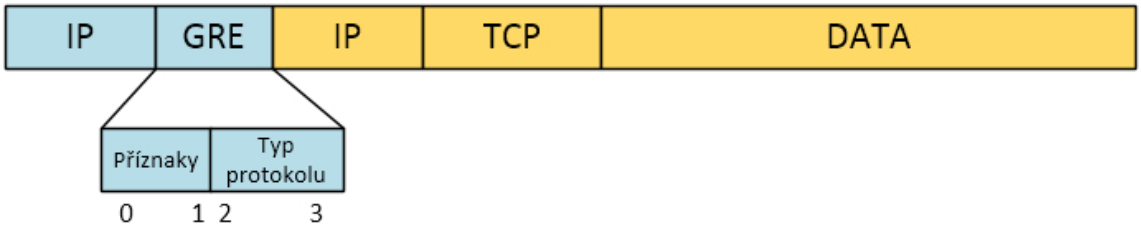
3.1 GRE

GRE[2] je zapouzdřovací protokol s číslem IP protokolu 47. Nejprve byl vyvinut společností Cisco, nyní je to otevřený standard specifikovaný RFC 2784 [3], podporuje přenášení mnoha protokolů a využívá IP jako nosný protokol.



Obrázek 3.2: GRE tunel

GRE je bezstavový, bez flow control a neposkytuje žádné zabezpečení (bez autentizace, šifrování,...). Režie GRE tunelu je 24B (20B pro novou IPv4 hlavičku a 4B pro GRE hlavičku). V hlavičce, obr. 3.3, je mimo jiné identifikátor přenášeného protokolu, pro IPv4 0x0800, PPTP 0x880b.



Obrázek 3.3: Struktura GRE paketu

Protože podporuje multicast, využívá se spolu s IPsec zabezpečením pro šíření směrovacích informací skrz tunel veřejnou sítí a dále se používá s PPTP pro vytvoření VPN, k tunelování IPv6 paketů a tunelování obecně.

3.2 IPSec

Sada standardů (původně pro IPv6, zpětně implementovaná v IPv4), která řeší jakým způsobem zabezpečit přenos dat v počítačových sítích. Dovoluje systému zvolit jaké chce použít bezpečnostní protokoly a algoritmy. Lze také vytvořit několik tunelů pro různé účely mezi dvěma zařízeními [6].

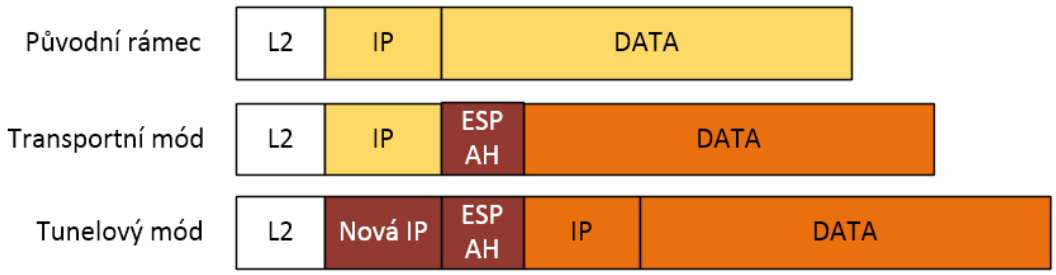
[2]GRE - Generic Routing Encapsulation

IPSec se používá s jinými tunelovacími protokoly pro zajištění:

- důvěrnosti dat – nikdo nemůže v průběhu komunikace mezi odesilatelem a příjemcem data přečíst,
- integrity dat – nikdo nemůže data pozměnit, když jsou přenášena sítí,
- autentizace původu dat – víme přesně, kdo data poslal,
- anti-replay ochrany - nikdo nemůže poslat odchycená data znovu.

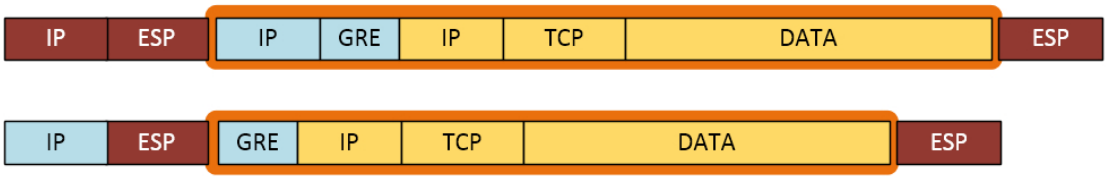
IPSec pro svou funkčnost dále využívá tyto protokoly:

- Internet Key Exchange (IKE) pro bezpečný přenos sdílených klíčů a NAT-T podporu (UDP porty 500 a 4500),
- Authentication Header (AH) pro autentizaci odesilatele, integritu dat a volitelnou ochranu proti replay útokům,
- Encapsulating Security Payload (ESP) pro šifrování dat, autentizaci odesilatele, integritu dat a a volitelnou ochranu proti replay útokům.



Obrázek 3.4: IPSec módy

Na obrázku 3.5 lze vidět použití IPSec pro zabezpečení GRE tunelu.



Obrázek 3.5: Struktura IPSec paketu s GRE

3.3 VPN

VPN je virtuální privátní síť propojující koncové zařízení (uživatelův počítač, smartphone, atd.) většinou s neveřejnou sítí, ve většině případů i s privátním rozsahem adres udržovaným

vlastním administrátorem, přes veřejný internet. K propojení komunikujících stran využívá různé tunelovací protokoly, tento tunel bývá většinou autentizován a šifrován. VPN lze použít pro projení dvou vzdálených sítí nebo pro připojování uživatelů do podnikové sítě.

3.3.1 PPTP

Point-to-Point Tunneling Protocol (PPTP) využívá protokol PPP a ten zapouzdřuje do GRE. Je používán v sítích Microsoft jako standardní typ VPN. Protokol podporuje šifrování, ale jelikož není moc bezpečné[3], často se nahrazuje L2TP/IPSec.

Obrázek 3.6: Struktura PPTP paketu

PPTP server

Aplikace umožňující vytvořit na počítači PPTP server se nazývá pptpd (pptp deamon). Při konfiguraci je potřeba určit lokální IP adresu, kam se klienti mohou připojovat, vzdálené IP adresy, které bude pptpd server přidělovat klientům po připojení k VPN. Definovat jaké autentizační protokoly jsou povolené a které se použít nesmí (PAP, CHAP, MS-CHAP, MS-CHAPv2 a MPPE). Jednotlivým uživatelům přidělit hesla a také je nutné na zařízení povolit směrování.

Konfigurace pptpd serveru je uvedena v příloze B.1

3.3.2 openVPN

Jedná se o Open Source alternativu k uvedeným typům VPN sítí. Umožňuje značnou interoperabilitu, software poskytující tunelování veřejnou sítí existuje pro velké množství platforem. V případě blokování některých protokolů providerem stále umožňuje zajistit VPN spojení přes vybrané porty. IP pakety nebo Ethernetové rámce jsou po síti přenášeny šifrovaným UDP spojením [8].

Paket OpenVPN lze vidět na obrázku 3.7.

Obrázek 3.7: OpenVPN paket

[3] V PPTP používaný šifrovací protokol MS-CHAPv2 byl prolomen na konci července 2012.

OpenVPN server

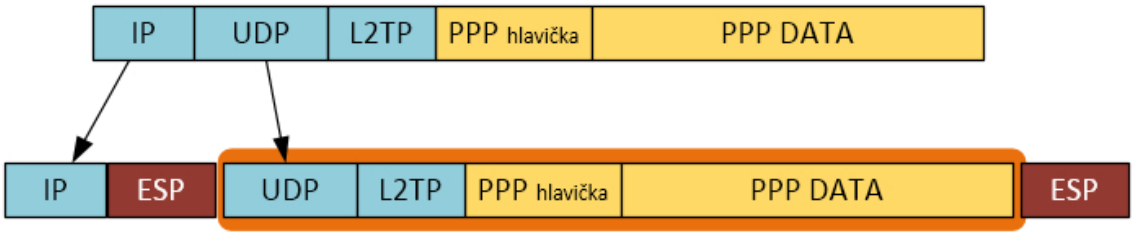
OpenVPN pro vytvoření spojení využívá program openssl a pro připojení klientů je výhodné využít certifikát. K vytvoření uživatelského certifikátu je potřeba nejprve vytvořit certifikační autoritu. Certifikační autorita, v našem případě server, zajišťuje důvěryhodnost připojení uživatelů k serveru.

Po vytvoření certifikační autority lze generovat další certifikáty pro server a klienty. Vygenerované žádosti o certifikát pro server a klienty jsou podepsány certifikační autoritou a tím vznikne plnohodnotný certifikát.

Konfigurace serveru je uvedena v příloze B.2. Klientovi pošleme konfigurační soubor, viz příloha B.3, klientské certifikáty spolu s klíčem, certifikátem certifikační autority a následně se může přihlásit na funkční VPN server.

3.3.3 L2TP

Protokol vzniklý z Cisco L2F a Microsoft PPTP. L2TP sám o sobě nenabízí žádnou možnost šifrování, nešifruje vnitřní PPP pakety jako PPTP. Pro možnost šifrování používá protokol IPSec v transportním režimu a tato varianta je známá jako L2TP/IPSec [7].



Obrázek 3.8: Struktura L2TP paketu šifrovaného pomocí IPSec

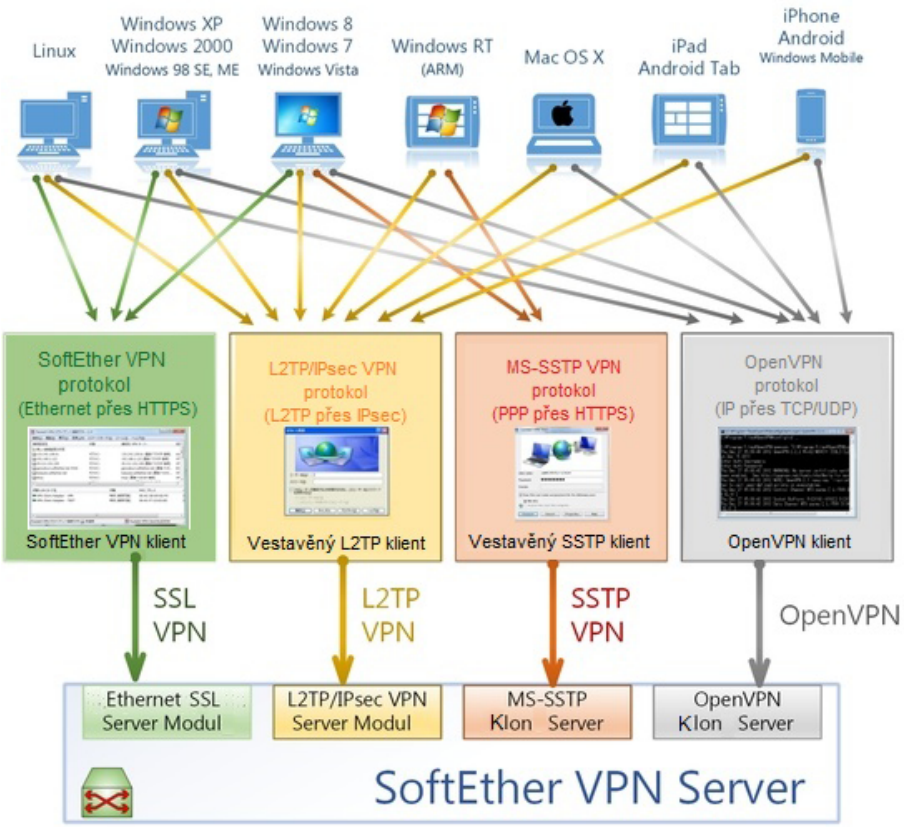
SoftEther VPN server

SoftEther je volný, open-source program vyvíjený v akademickém projektu Japonské univerzity v Tsukuba. První vydání bylo v roce 2013. Dává si za cíl vyvíjet a distribuovat univerzální VPN software. Program je dostupný na mnoha platformách a podporuje několik protokolů:

- OpenVPN
- L2TP/IPsec
- L2TPv3/IPsec
- EtherIP
- Microsoft SSTP
- VPN přes HTTPS (SSL-VPN)

- VPN přes DNS
- VPN přes ICMP

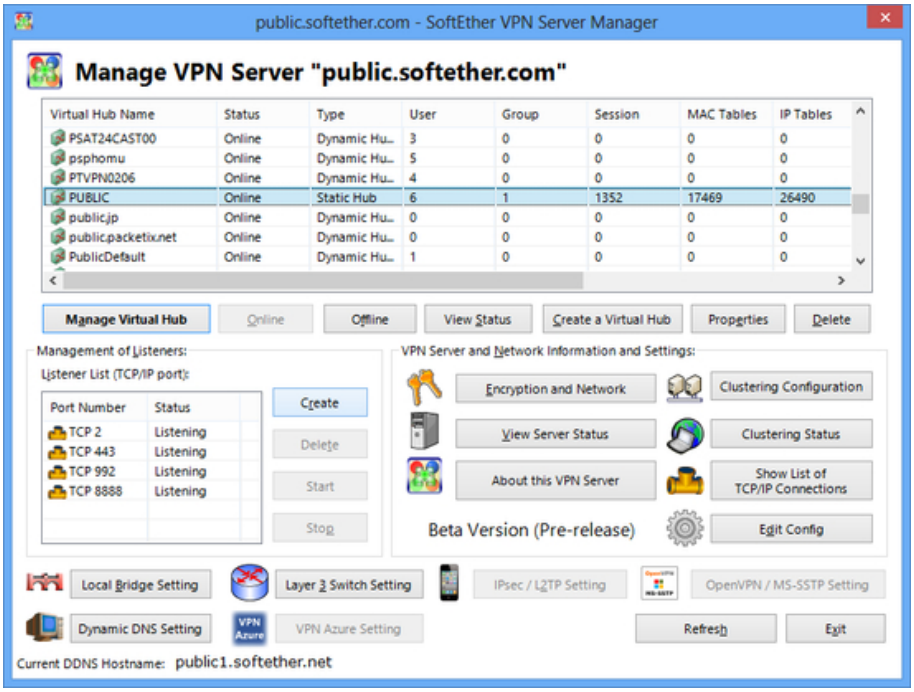
Jelikož L2TP má v mnoha systémech (Windows, Mac OS X, iOS, Android) nativně zabudovaného VPN klienta, není potřeba v případě využití tohoto protokolu doinstalovávat na klientské zařízení žádný další software. Zabudovaní VPN klienti mohou mít v některých případech velké pozitivum, například když uživatel nemůže spouštět programy jako správce. Dá se ovšem použít také s programem OpenVPN, lze využít certifikáty, RADIUS, Active Directory. V případě, že bychom se chtěli připojit do internetové sítě přes síť, která pro zabezpečení využívá Captive portál, například na letišti, je možné využít tunelování VPN přes DNS a ICMP.



Obrázek 3.9: Možnosti připojení k SoftEther serveru

Na obrázku 3.9 z oficiální stránky SoftEther projektu [15] lze názorně vidět možnosti připojení k VPN serveru z různých zařízení. SoftEther také poskytuje vlastní protokol SSL-VPN využívající tunelování přes HTTPS. Tento protokol může mít výhodu v prostředí s firewallem a NATem, protože není potřeba doplňovat další pravidla pro VPN spojení.

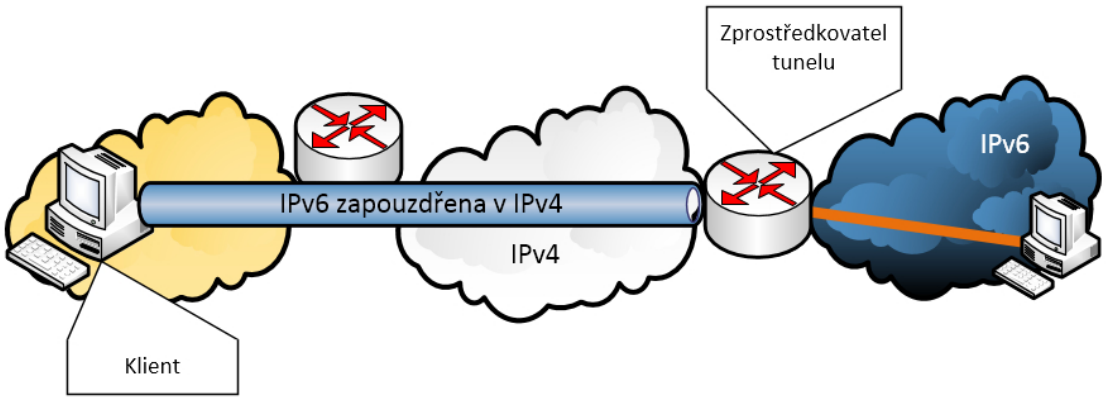
Nastavení jednotlivých protokolů lze snadno spravovat z aplikace s grafickým rozhraním SoftEther VPN Server Manager, viz obrázek 3.10, což také dává výhodu oproti nutnosti zadávat konzolové příkazy při konfiguraci dříve zmíněných protokolů.



Obrázek 3.10: Hlavní obrazovka SoftEther VPN Server Manager

3.4 Přechodové mechanismy IPv6

Již několik let je standardizována nová verze IP protokolu. Jelikož vyměnit všechny směrovače za kompatibilní s novým protokolem není v dnešní síti, obsahující tisíce zařízení, snadné, existuje několik mechanismů umožňujících uživatelům využívat IPv6 přes stávající IPv4 síť. Všechny tyto mechanismy využívají tunelování, viz obr. 3.11.



Obrázek 3.11: IPv6 tunelováno v IPv4

Základem většiny dále popsaných mechanismů je dostupnost duální implementace protokolu IP tzv. *Dual stack* na uživatelském zařízení. V případě, že máme nativně dostupnou IPv4 i IPv6 síť používají se obě rovnocenně. Pokud je ovšem dostupná pouze síť s protokolem IPv4, pro dosažení zařízení dostupných v IPv6 síti nastupují přechodové mechanismy.

Mimo níže zmíněných protokolů jsou také známé 6over4 nebo IPv6 tunel protokolem AYIYA.

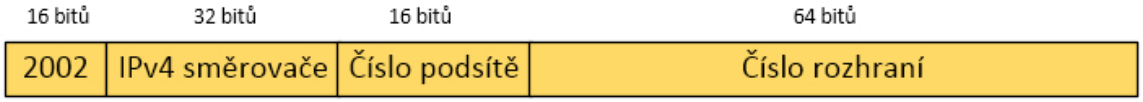
3.4.1 6to4

Přechodový mechanismus 6to4 je definován RFC 3056 a má sloužit k propojení jednotlivých IPv6 ostrůvky přes IPv4.

Tento mechanismus rozlišuje 3 základní typy uzlů v síti:

- 6to4 směrovač (6to4 router) - zařízení s veřejnou IPv4 adresou. Většinou hraniční směrovač mezi menší IPv6 sítí a IPv4 sítí.
- Předávací směrovač (relay router) - 6to4 směrovač nakonfigurovaný pro směrování provozu mezi 6to4 adresami a nativními IPv6 adresami.
- 6to4 klient (6to4 host) - zařízení s alespoň jednou 6to4 adresou. Jinak standardní IPv6 klient.

Z IPv4 adresy 6to4 směrovače se vytvoří IPv6 adresa s prefixem 2002, viz obrázek 3.12. Tím vznikne 48 bitů dlouhý prefix, kterým se dají adresovat koncové podsítě a počítače.



Obrázek 3.12: IPv6 adresa v 6to4

Možným mínusem pro tento způsob může být nutnost veřejné adresy

3.4.2 ISATAP

ISATAP[4] je přechodový mechanismus určený pro přenos IPv6 paketů mezi zařízeními s duální implementací IP přes lokální IPv4 síť. Podobně jako 6to4 vytváří IPv6 adresu z IPv4 adresy.

K prefixu `fe80:0000:0000:0000:0000:5efe::/96` se přidá 32 bitů původní IPv4 adresy.

IP vrstva je používána jako *nonbroadcast multiple-access network*, tedy jako síť kde se přenáší data přímo mezi dvěma počítači, bez možnosti poslat hromadnou zprávu všem. Kvůli tomu nemůže být pro objevování sousedů využito ICMPv6 a zařízení tak nemohou automaticky nalézt směrovač. Pro zjištění směrovače se používá *seznam potenciálních směrovačů - PRL*[5]. Tento seznam se v praxi získává pomocí DNS, dotazem v lokální doméně na isatap.domena.cz.

3.4.3 Teredo

Pro překonání NATu, kvůli kterému nemusí některé přechodové mechanismy fungovat, byl vyvinut mechanismus Teredo.

Ke správnému fungování potřebuje tři součásti:

[4]ISATAP - Intra-Site Automatic Tunnel Addressing Protocol
[5]PRL - Potencial Router List

- Teredo klient - zařízení využívající Teredo.
- Teredo server - uzel připojený současně do IPv4 a IPv6 sítě, sloužící k inicializaci spojení Teredo klienta s IPv6 uzlem. Přiděluje Teredo klientovi IPv6 adresu, ale nikdy nepřeposílá pakety, to dělá relay.
- Teredo relay - uzel ukončující Teredo tunely a směruje pakety mezi Teredo klienty a IPv6 uzly.

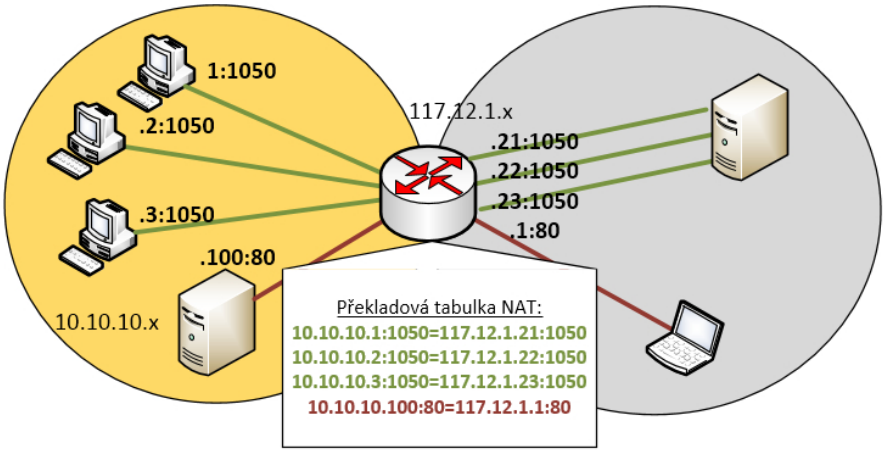
Kapitola 4

Překlad adres

Pro možnost rozšíření adresního prostoru IPv4[1] se objevovali různé způsoby. Vyvíjel se zcela nový protokol IPv6 vycházející z IPv4 a také mechanismus, jak pracovat s aktuálním protokolem, a to překlad síťových adres - NAT [14]. Směrovače na hranici privátní/podnikové sítě překládají přidělené veřejné IPv4 na adresy z privátního rozsahu, který není přístupný z veřejného internetu. Překlad adres lze provozovat několika způsoby podle počtu dostupných veřejných adres, viz dále.

4.1 Překlad na síťové vrstvě - NAT

NAT jedna ku jedné (NAT 1:1, basic NAT, static NAT) umožňuje pouze překlad adres, nikoli mapování portů. Tato možnost vyžaduje IP adresu pro každé samostatné zařízení. Tento typ NATu využívají například lokální poskytovatelé internetu v případě, že mají svoji síť adresovanou privátním rozsahem a některým klientům pak přidělí i veřejnou adresu.



Obrázek 4.1: NAT

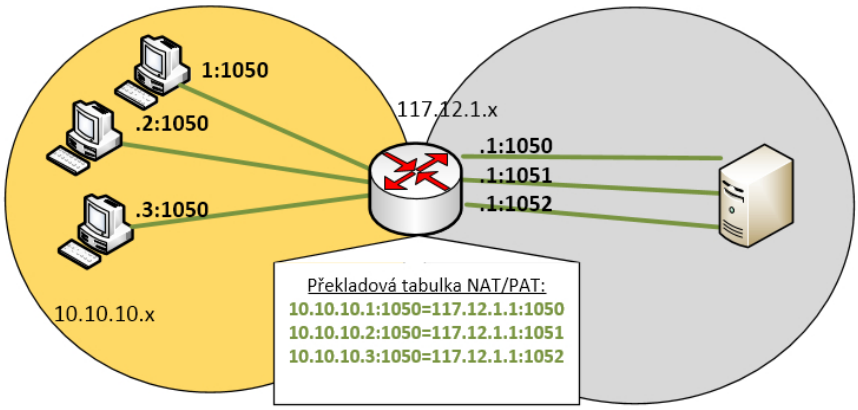
NAT 1:1 přiřazuje adresy většinou dynamicky a pokaždé může být přidělena odlišná

[1]Poslední bloky adresního prostoru IPv4 s maskou /8 byly přiděleny organizací Internet Assigned Numbers Authority (IANA) regionálním registrátorům 3. února 2011

adresa. V některých případech, například když máme na privátní sítí server a chceme, aby byl přístupný i z veřejného internetu, použijeme statický NAT. Obrázek 4.1 zobrazuje počítače připojující se z lokální sítě do internetu zelenou barvou a počítač připojující se na lokální server z veřejné sítě červenou barvou.

4.2 Překlad na síťové a transportní vrstvě - PAT

Aktuálně veřejných IPv4 adres není dostupných tolik, aby si každý mohl dovolit vlastnit takový adresní rozsah, který by dovolil používání NATu 1:1. V dnešní době vetšinou uživatelé a poskytovatelé disponují malým množstvím veřejných adres. Z tohoto důvodu se v dnešní době nejvíce používá překlad pomocí portů, tzv. overloaded NAT. Network Address Port Translation (NAPT, PAT) je technika, kdy dochází k mapování čísel portů na jednu IP adresu. Několik zařízení pak může sdílet jednu veřejnou IPv4 adresu, viz obrázek 4.2.



Obrázek 4.2: Overloaded NAT

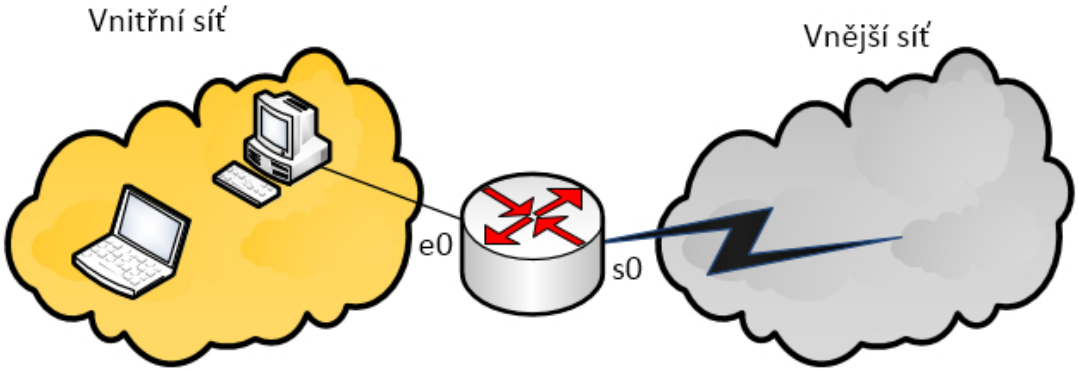
4.3 NAT a Syslog

Pro monitorování překládání síťových adres lze zasílat zprávy pomocí protokolu syslog na vzdálené zařízení.

4.3.1 Překladu síťových adres na Cisco směrovači

Při konfiguraci NATu je nejprve nutné určit rozhraní, která vedou do vnitřní sítě (inside) a vnější sítě (outside), viz obrázek 4.3.

Po určení směru rozhraní vybereme požadovaný druh NATu. Jednotlivé typy popsané v kapitole 4 se mohou i kombinovat. Pokud se jedná o dynamický NAT nebo PAT nastaví se z jakého prostoru budou adresy přidělovány. U obou typů lze zadat rozsah nebo jednu adresu. Adresy z rozsahu se přidělují postupně a pokud je přidělena poslední u NAT 1:1 už se další překlad nepovede a u PAT se začne překládat na jiná čísla portů. Pro dynamický překlad je nutné pomocí access listu určit jaké adresy se mohou překládat.



Obrázek 4.3: Vnitřní a vnější NAT rozhraní

Je také důležité určit na jakém rozhraní se bude překlad provádět, tímto se určí zároveň zda se bude měnit zdrojová či cílová adresa pro určitý směr procházení paketů.

- Překlad na vnitřním rozhraní (`inside source`):
  Přeloží zdrojovou adresu IP paketů putujících z vnitřní do vnější sítě.
  Přeloží cílovou adresu IP paketů putujících z vnější do vnitřní sítě.
- Překlad na vnějším rozhraní (`outside source`):
  Přeloží zdrojovou adresu IP paketů putujících z vnější do vnitřní sítě.
  Přeloží cílovou adresu IP paketů putujících z vnitřní do vnější sítě.

Nachází-li se ve vnitřní sítí webový server, který má být dostupný z internetu, lze povolit přesměrování portů a server tak zpřístupnit.

4.3.2 Syslog na Cisco směrovači

Syslog je protokol, původní specifikace RFC 3164, a standard pro záznam systémových a programových zpráv. Většina síťových zařízení podporuje posílání zpráv na Syslog server, ani u Cisco zařízení tomu není jinak.

Zprávám je přiřazena priorita, či úroveň závažnosti, udávající jak moc jsou reportované událostí ve zprávě závažné.

- Emergency - Nejzávažnější zprávy, program nebo celý systém může být nepoužitelný.
- Alert - Poplašné zprávy, reportovaná událost by měla být okamžitě napravena, například výpadek primárního připojení k internetu.
- Critical - Kritická hlášení, vetšinou označující výpadky sekundárních systémů, například výpadek záložního připojení k internetu.
- Error - Chybová hlášení nepříliš urgentních chyb.
- Warning - Varování před možnými chybami, například že na disku dochází místo.
- Notice - Upozornění na neobvyklé události, které ale nejsou chybami.

- Info - Informace týkající se běžného provozu.
- Debug - Ladící informace užitečné například pro vývojáře.

Jednotlivé zprávy se pak vztahují k zařízením *facility*: auth, authpriv, daemon, cron, ftp, lpr, kern, mail, news, syslog, user, uucp, local0, ..., local7. Toto zařazení se dá použít na serveru pro filtrování zpráv.

Na Cisco zařízení při konfiguraci zasílání zpráv na syslog server lze tedy zvolit jak moc závažné zprávy zasílat, jak budou zprávy označeny (*facility*) a adresu syslog serveru. Konfigurace v příloze C.2.

Pro posílání zpráv týkajících se překladu adres je důležité zapnout vypisování ladících informací.

4.3.3 Syslog server

U systémů Fedora a Ubuntu existují dvě primárně používané implementace Syslogu, a to:

- **Syslog-ng** objevil se roku 1998 a na některých systémech byl nebo stále je defaultní syslog server.
- **Rsyslog** byl vytvořen roku 2004 jako konkurent Syslog-ng. Od roku 2007 byl na Fedoře primárním syslog serverem a to do roku 2013, kdy byl nahrazen journald

Pro nastavení syslog serveru většinou stačí editovat pár řádků konfiguračního souboru, viz příloha C.3

Kapitola 5

Rozšíření systému Sec6Net Lawful Interception System

V této kapitole si popíšeme navržené rozšíření systému pro zákonné odposlechy. Rozšíříme jej o některé z výše uvedených protokolu a mechanismů. Celá práce bude rozdělena do dvou modulů, z nichž každý se bude starat o jednu z těchto kategorií:

- tunelovaný provoz - VPN: modul bude analyzovat logovací soubory nacházející se na VPN serveru,
- překlad adres - NAT: modul bude od směrovačů přijímat zprávy o jejich činnosti a kontrolovat, zda-li nedošlo k překladu adresy.

Protože v dnešní době je již nativní IPv6 konektivita relativně dostupná a využití přechodových mechanismů už je minimální, nebude tato varianta tunelování do rozšíření zahrnuta.

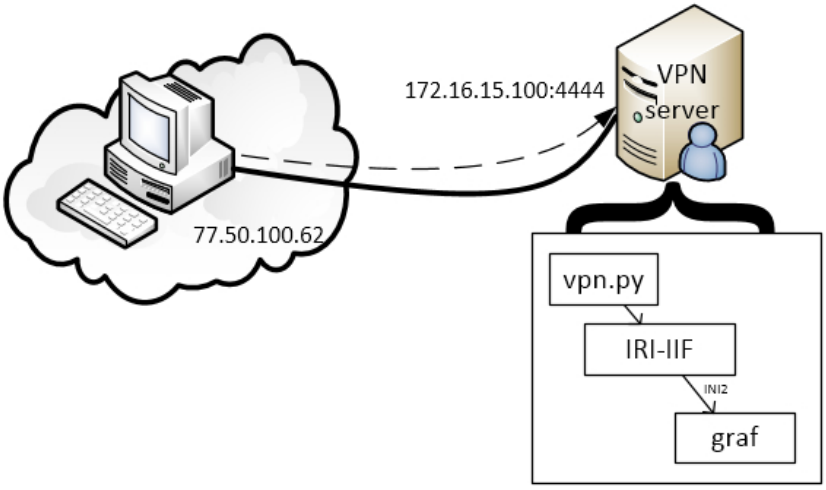
Oba moduly by pak měly podporovat zpracování několika souborů dohromady. Protože informace budou získávány z logovacích souborů, které se někdy mění s aktuálním datem, například *soubor-20150512.log*, je také důležité zabezpečit podporu rotace souborů na základě data.

5.1 Identita v tunelovaném provozu

Protože u mechanismů využívajících tunelovací protokoly jsou ve většině případů vnitřní pakety, viz obr. 3.1, šifrovány a lze zjistit bezpečně pouze IP hlavička vnějšího paketu, bude v těchto případech identita určována z logů jednotlivých protokolů. V tomto případě bude tedy nutné dodatečné nastavení poskytovatelem konce tunelu, ať už posílání logů do systému pro zákonné odposlechy nebo nainstalování softwarové sondy na server.

Umístění logů na serveru se může u různých variant a verzí software lišit. Například u rozšířené varianty VPN OpenVPN 3.3.2 lze logovací soubory nalézt v `/var/log/syslog` a u některých verzí se dá nastavit logovací soubor v nastavení `/etc/openvpn/server.conf`. V navrhovaném modulu tedy budeme kontrolovat, zda se soubor nezměnil a případně zkontrolujeme, jestli se neobjevil záznam o připojení či odpojení klienta. Kontrolu můžeme provádět pomocí balíčku `inotify-tools`, který zajistí kontrolování změny souboru.

Na obrázku 5.1 je zobrazen VPN server, který zároveň zpracovává identity uživatelů. V případě, že je detekováno připojení (odpojení) klienta do VPN sítě, modul pošle IRI zprávu Begin (End) jádru IRI-IIF, viz tabulka 2.1 v kapitole 2.2.



Obrázek 5.1: Klient se připojuje na VPN

5.1.1 Činnost modulu

Nyní se podrobněji podíváme na činnost navrhovaného modulu. Zjistíme jaké informace bude v logovacích souborech hledat a kdy bude posílat zprávy jádru IRI-IIF.

PPTP připojení

Z logovacího souboru pptp serveru lze kontrolovat následující události:

- autorizaci klienta s IPv4 adresou - *peer authorized*,
- přidělení IPv4 adresy na rozhraní VPN - *remote IP*,
- odpojení klienta s IPv4 - *connection finished*.

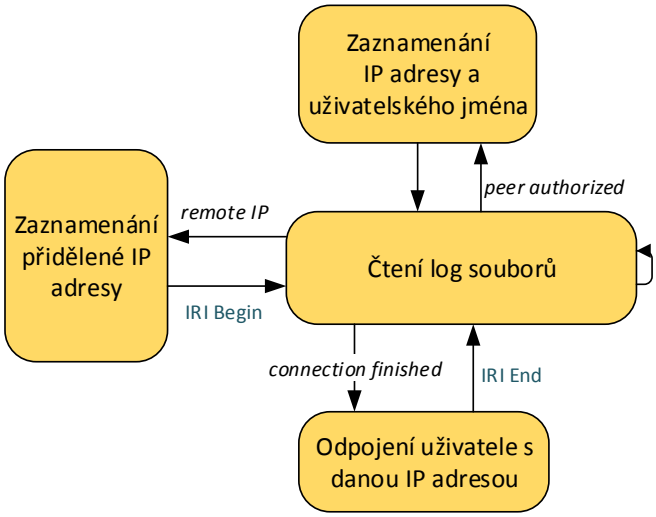
Pro získání uživatelského jména lze použít příkaz `last`, který vypisuje přihlášení uživatelů.

V případě, že se objeví záznam o autorizaci klienta, modul zaznamená jeho adresu a uživatelské jméno. Po přidělení vzdálené adresy modul generuje zprávu *Begin* pro jádro IRI-IIF. Jakmile je přečtena informace o odpojení klienta modul vytvoří zprávu *End*, viz diagram na obrázku 5.2

L2TP (SoftEther) připojení

Software SoftEther k průběhu spojení s L2TP klientem udává tyto události:

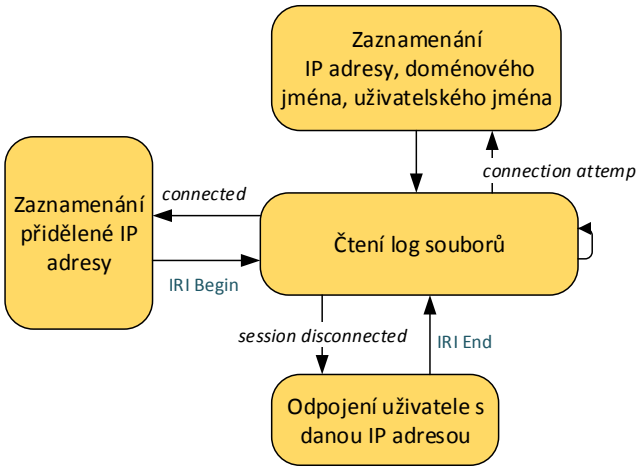
- pokus o připojení - *connection attemp*,
- pokus o připojení se zdařil - *connected*,



Obrázek 5.2: Diagram zpracování PPTP

- odpojení klienta - *session disconnected*.

Je-li zaznamenán pokus o připojení, modul si uloží informace o IPv4, doménovém jménu počítače a uživatelské jméno. Pokud je pokus o připojení úspěšný, je zaznamenána přidělená IPv4 adresa a jádru IRI-IIF je poslána zpráva typu *Begin*. IRI zpráva *End* je generována po odpojení klienta, viz obrázek 5.3.



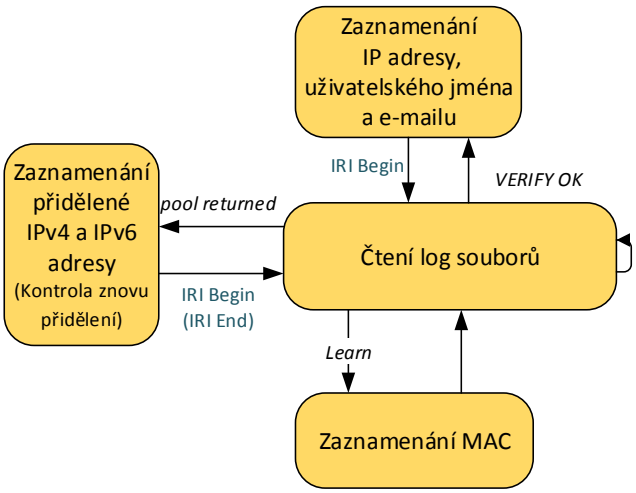
Obrázek 5.3: Diagram zpracování L2TP

OpenVPN připojení

U OpenVPN jsou poskytované informace o událostech následující:

- verifikace klienta - *VERIFY OK*,
- přidělení IPv4 a IPv6 adres - *pool returned*,
- MAC adresa klienta - *Learn*.

Při verifikaci klienta se uloží informace o IPv4 adrese, uživatelském jménu a e-mailu a vygeneruje se zpráva *Begin* pro jádro IRI-IIF. Při zjištění přidělení adres se opět zaznamenají a vygeneruje se zpráva *Begin*, to samé při zjištění MAC adresy. Bohužel OpenVPN neposkytuje informace o odpojení klientů a tudíž je nutno odpojení klientů zjistit jinak. Po připojení každý klient dostane přidělenu IPv4 adresu. Pokud je adresa přidělena znovu jinému klientovi, dá se předpokládat že předchozí klient je již odpojen a generuje se zpráva *End* o odpojení klienta. Zpracování souboru lze vidět na obrázku 5.4.

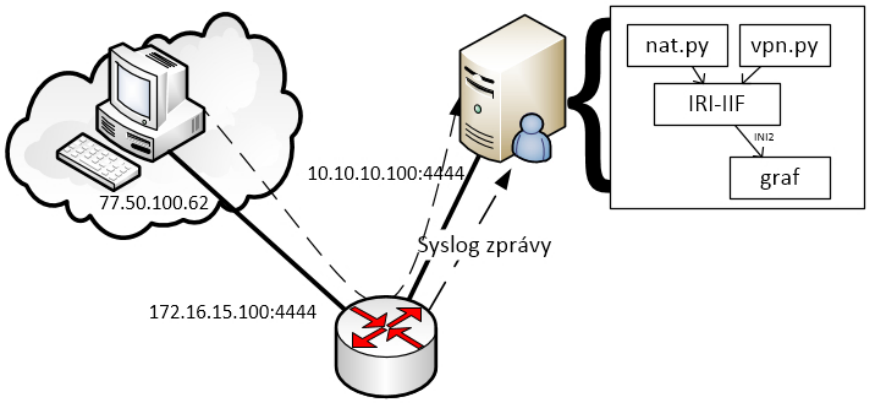


Obrázek 5.4: Diagram zpracování OpenVPN

5.2 Identita v překládaných sítích

Překlad adres probíhá výhradně na síťových směrovačích, které mají většinou specifický operační systém/software. Na tyto operační systémy nelze systém pro zákonné odposlechy nainstalovat a proto se musí identita při překladu adres určovat z logů, které router zašle na server s běžícím systémem pro zákonné odposlechy. K tomuto účelu bude použit standard Syslog, který je určen pro přenos logovacích dat a je také podporován na mnoha síťových směrovačích.

Pokud ve firmě či doma zprovozníme VPN server, často se nachází i za NATem. Aby se mohli klienti z veřejné sítě připojit na VPN, je potřeba na zařízení, které provádí překlad adres zajistit také přeposílání portů na VPN server v privátní síti. Na obrázku 5.5 je zobrazeno, jak takové připojení může vypadat.



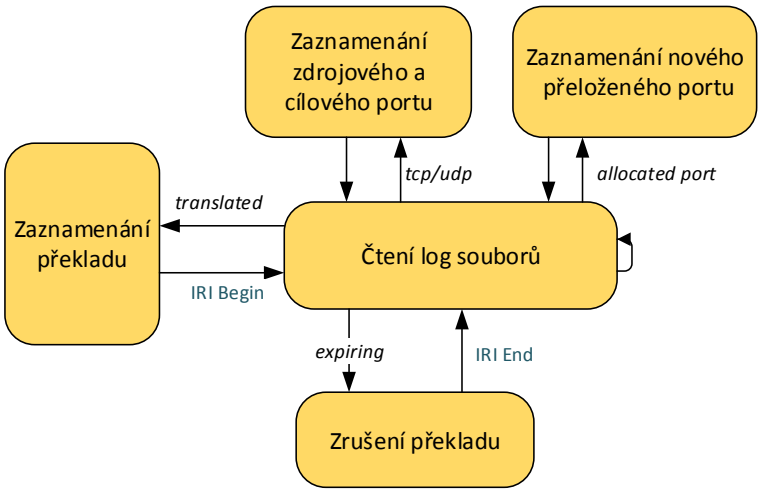
Obrázek 5.5: PC se připojuje na VPN v privátní síti

5.2.1 Činnost modulu

K zaznamenávání překladu adres modul vyhledává v logovacím souboru tyto události:

- komunikace tcp nebo udp,
- alokace portu,
- překlad adresy,
- expirace překladu adresy.

Na obrázku 5.6 lze vidět zaznamenávání jednotlivých událostí.



Obrázek 5.6: Diagram zpracování NAT

Pokud je v souboru zaznamenána komunikace tcp nebo udp, uloží se zdrojové a cílové porty. Alokace portu znamená vytvoření nového překladu adresy s překladem portu. Při přečtení informace o překladu adres se zaznamenají zdrojová, cílová adresa a přeložená

adresa. Pokud jsou uloženy tcp nebo udp porty vygeneruje se *Begin* zpráva s těmito porty a alokovaným portem přeložené adresy jinak se vygeneruje zpráva bez označení portů. Při zrušeni překladu se generuje zpráva *End*.

Kapitola 6

Implementace

V této kapitole se seznámíme s vlastní implementací rozšíření. Pro snazší možnost publikace výsledku je implementace prováděna na softwaru Správa identity z projektu Sec6Net [12], tzv. SIMS[1], což je část systému pro zákonné odposlechy obsahující Administrační funkci a Funkci dynamické identity IRI-IIF, viz kapitola 2. Jednotlivé moduly jsou jako většina systému psány v programovacím jazyku Python.

Pro implementaci bylo nejprve nutné zprovoznit vybrané mechanismy překladu adres a tunelování. Vše bylo prováděno ve virtualizovaném prostředí GNS3, více o tomto softwarovém emulátoru pro sítě v kapitole 7.1.

Funkcionalita se nachází ve dvou hlavních modulech, první zpracovává záznamy z VPN serverů a druhý zpracovává přijaté Syslog zprávy.

V modulu definujícím použitelné NIDy byly přidány dva identifikátory:

- HOST NAME - reprezentující jméno počítače vyskytující se v informacích ze SoftEther VPN,
- VPN Login - přihlašovací jméno použité pro přihlášení do VPN sítě.

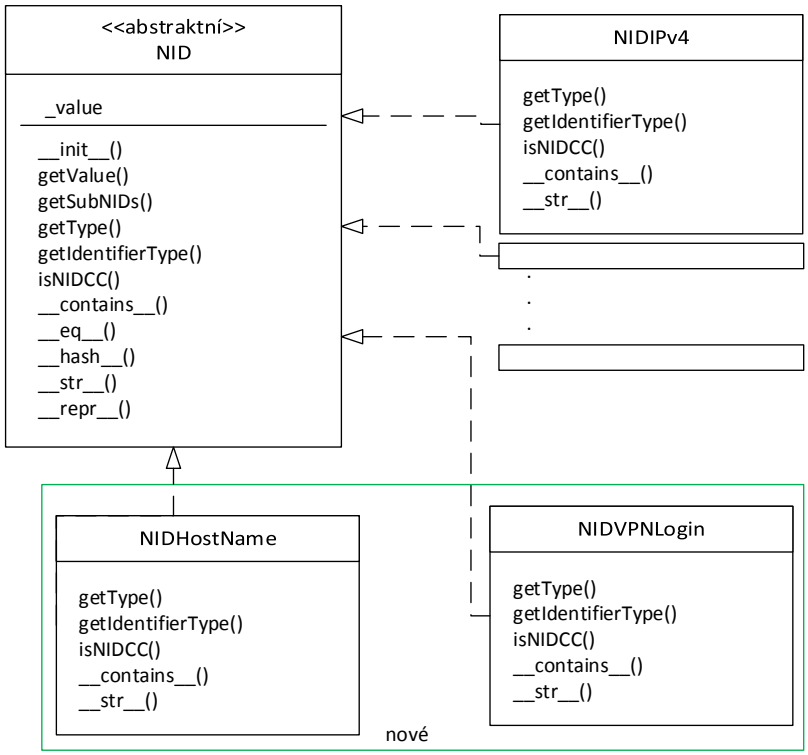
Z již dostupných NIDů jsou využívány tyto:

- IPv4,
- IPv6,
- MAC,
- E-mail address,
- TCP.

Všechny síťové identifikátory vychází z abstraktní třídy `NID`, která definuje rozhraní pro získání například typu NIDu a typu identifikátoru, viz obrázek 6.1.

Bylo také nutné upravit konfigurační soubor iri kolektoru a v něm zadefinovat nové moduly.

[1] SIMS - Sec6Net Identity Management System



Obrázek 6.1: Diagram tříd vybraných NID

6.1 Modul VPN

Modul VPN se stará o čtení zadaných logovacích souborů, kterých může být i více, a v nich hledá důležité identifikátory pro určení identity.

Názvy logovacích souborů se v závislosti na nastavení mohou měnit například podle dne, proto se po zadaní parametru doplní aktuální datum, např. *soubor-20150420.log*.

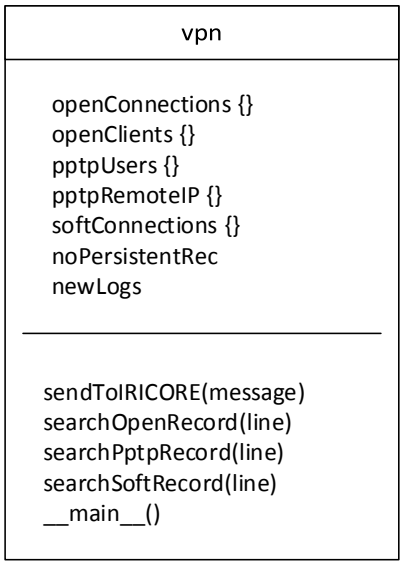
Modul nejprve zpracuje parametry, které mohou být následující:

- `-f,--files=` : seznam vstupních souborů, které obsahují informace z VPN serverů.
- `-r,--rotateLogs` : pokud je tento parametr zadán, řetězec `<DATE>` v seznamu souborů je nahrazen aktuálním datem.
- `-p` : informace o připojení klientů jsou perzistentní, po odpojení zůstane záznam v IRI-IIF. Lze použít nejlépe k demonstračním účelům, kteří klienti byli připojeni.
- `-n` : modul prověřuje pouze nově příchozí události, tedy při otevření souboru přeskočí nakonec. Pokud není tento parametr zadán logovací soubory se čtou celé.

Spuštění programu pak může vypadat takto:

```
./vpn.py --files="openvpn.log","/var/log/messages-<DATE>"--rotateLogs
```

Jak lze vidět na obrázku 6.2, třída vpn definuje několik globálních asociativních polí (slovníků) a proměnných. Proměnné se používají pro uložení vstupních parametrů. Proměnná `newLogs` uchovává hodnotu vstupního parametru `-n` a `noPersistentRec` hodnotu parametru `-p`, viz výše.



Obrázek 6.2: Diagram třídy vpn.py

Po otevření všech vstupních souborů se postupně načítají jednotlivé řádky a předávají se na zpracování. Hledání informací z různých VPN serverů zabezpečují odlišné funkce `search<DRUH>Record(line)`. Pokud některá metoda narazí na hledaný identifikátor generuje pro jádro IRI-IIF zprávu `Begin`. Mapování jednotlivých identifikátorů na NID lze vidět v tabulce 6.1. Každá metoda eviduje připojené klienty, ve většině případů pomocí slovníku s klíčem IP adresy. Například metoda na vyhledání OpenVPN připojení používá 2 slovníky. První pro uložení IP adresy na základě klíče, kterým je IP adresa přidělená uživateli ve VPN tunelu. Druhý pak pro uložení informací o připojených klientech. Po zpracování informace, že IP adresa se odpojila ze serveru vyhledá odpovídající záznam ve slovníku a jádru IRI-IIF pošle *END* zprávu.

V souborech se mohou nacházet tyto zaznamenávané údaje:

Identifikátor		**NID**
IP adresa klienta		IPv4
Přihlašovací jméno		VPN Login
IP přidělená klientovi ve VPN tunelu		IPv4,IPv6
MAC adresa		MAC
E-mail	pouze OpenVPN, z certifikátu	E-mail address
Host name	pouze softEther	HOST NAME

Tabulka 6.1: Mapování identifikátorů v logovacích souborech na NID

Pokud modul objeví v souboru hledaný identifikátor, pošle (`sendToIRICore()`) pomocí unix soketu zprávu zda se jedná o připojení nebo odpojení klienta.

U pptpd serveru v logovacích informacích objevíme pouze jaká IP adresa se připojila a jaká IP adresa byla klientovi přidělena. Jaký uživatel se připojil zaznamenává do souboru *wtmp*, proto je v metodě zpracovávající pptp záznamy použito volání funkce z příkazové

řádky `commands.getoutput('last')`. Po připojení klienta tedy modul zpracuje výstup programu `last` a hledá v něm IP adresu z pptpd logu. Pokud adresu nalezne na stejném řádku se nachází uživatelské jméno přihlašovaného.

Odpojení klientů se u většiny serverů zaznamenává, ale jelikož OpenVPN do logovacích souborů nezaznamenává jaký přesně klient se odpojil, vznikl problém podle čeho zrušit záznam v jádru IRI-IIF. Možnost aktivně se dotazovat serveru nebo klienta z podstaty programu, kdy by měl pouze pasivně kontrolovat identity uživatelů a zaznamenávat ji, nelze. Využití předem určené doby po které je uživatel odpojen a tak záznam zrušit je také problematické, jak dobu určit, i když čas neuplyne adresa může být přidělena již jinému uživateli. Jako nejlepší se jevil způsob ponechaní záznamu do té doby dokud VPN server nepřidělí přidělované adresy jinému uživateli. Pro uložení těchto adres slouží slovník `openConnections`. Pokud VPN server přidělí zaznamenanou adresu novému uživateli, je jasné že dříve připojený uživatel se musel odhlásit. V tento okamžik tedy zrušíme záznam o připojení a vytvoříme nový aktualizovaný.

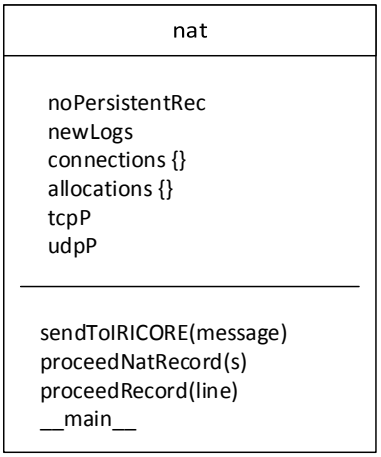
6.2 Modul NAT

Modul NAT podporuje podobné parametry jako modul VPN a přidává k nim další 2:

- `-f,--files=` : seznam vstupních souborů, které obsahují Syslog zprávy ze směrovače.
- `-r,--rotateLogs` : pokud je tento parametr zadán, řetězec `<DATE>` v seznamu souborů je nahrazen aktuálním datem.
- `-p` : informace o překladu adres jsou perzistentní, po expiraci zůstane záznam v IRI-IIF. Lze použít nejlépe k demonstračním účelům které adresy se překládaly a na jaké.
- `-n` : modul prověřuje pouze nově příchozí zprávy, tedy při otevření souboru přeskočí nakonec. Pokud není tento parametr zadán logovací soubory se čtou celé.
- `--IPconnect` : při zadání tohoto parametru se propojují identity na základě IP adres obsažených v tcp pětici. Jinak se propojují pouze přesné tcp pětice.
- `--PAT` : pokud je zadán tento parametr přeložené adresy se zaznamenávají s čísly portů, tedy překlad na síťové a transportní vrstvě. Jinak se zaznamenává překlad IP adres s číslem portu 0.

Za předpokladu, že na Cisco směrovači je nastaveno detailní vypisování ladících informací překladu adres, má proces překladu 3-4 fáze. V 1. fázi je alokován port překládané adresy, v modulu je tento port ukládán do slovníku `allocations` spolu s IP adresou a požadovaným portem. Pokud se jedná o tcp nebo udp spojení je v další fázi zaznamenán zdrojový a cílový port do pole `tcpP`, respektive `udpP`. Ve 3. fázi se objevuje informace o zdroji a cíli komunikace a jaká adresa se překládá. V této fázi se vytváří zpráva `Begin` pro jádro IRI-IIF. Ke zdroji a cíli se připojí adresy portů z fáze 2 a vytvoří se první tcp pětice. V závislosti na tom, zda se překládá zdrojová nebo cílová adresa, nahradí přeložená adresa a port z fáze 1 danou adresu a vytvoří se další tcp pětice. Obě tcp pětice se pak odešlou jádru IRI-IIF. Z důvodů aby se tcp pětice nespojovali s jinými identitami na základě pouze IP adresy, jsou ve 2.seznamu odesílané zprávy, viz seznamy NIDů v kapitole 2.2, uvedeny jednotlivé IP adresy. Pokud bychom z nějakého důvodu chtěli na základě IP adres tcp pětice spojovat, lze vynechat odesílání druhého seznamu parametrem `--IPconnect`. Poslední fází

je expirace překladu, kdy dojde k zaslání zprávy *End* jádru IRI-IIF. Na diagramu 6.3 lze vidět hlavní proměnné a metody třídy nat.



Obrázek 6.3: Diagram třídy nat.py

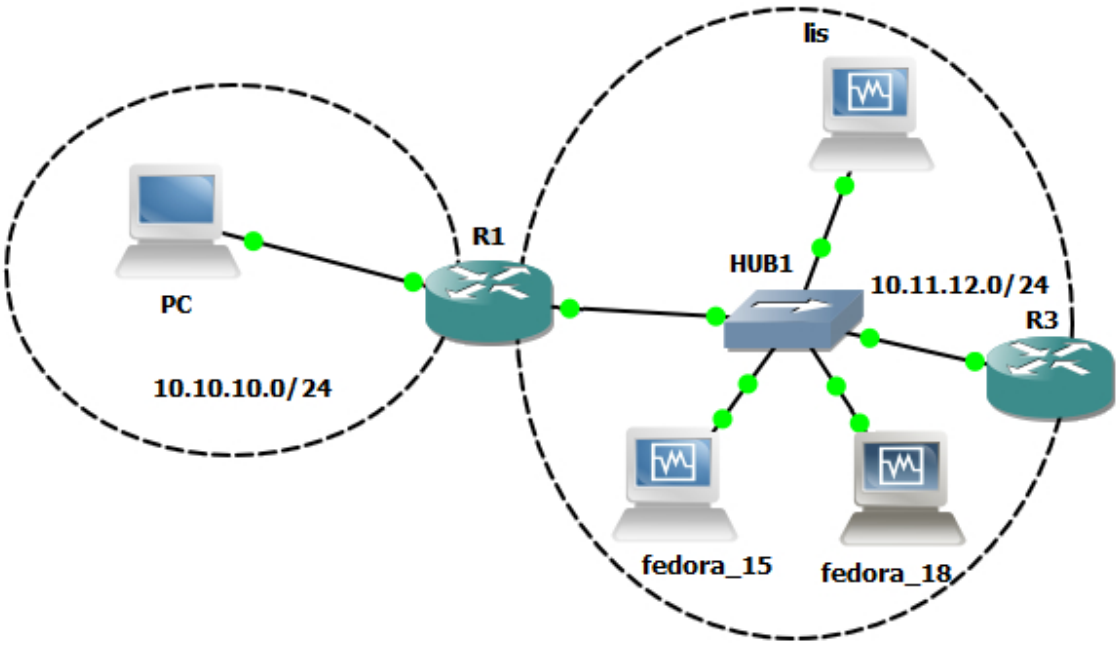
Kapitola 7

Testování

Základní testování probíhalo již při vlastní implementaci. Pro simulování jednoduché sítě byl použit nástroj GNS3 se 3 virtuálními počítači, 2 přepínači s Cisco IOS 3600, rozbočovačem a připojením k fyzickému rozhraní počítače, viz obrázek 7.1.

7.1 GNS3

Graphical Network Simulator-3 [5] je software umožňující kombinovat virtuální a reálná zařízení s možností libovolného propojení. GNS3 začalo jako nástroj zaměřený hlavně na Cisco. Nyní již podporuje emulaci mnoha jiných výrobců jako Arista, Juniper, HP, Alcatel a Extreme [4]. Lze si vytvořit v podstatě libovolnou síťovou topologii, do které se dají zakomponovat virtuální stroje VirtualBox, díky využití emulačního softwaru Dynamips lze simulovat Cisco IOS a celé prostředí pak napojit na reálnou síť.



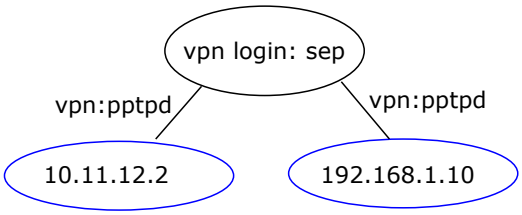
Obrázek 7.1: Prostředí GNS3

7.2 Testování modulů

Nejprve byly testovány jednotlivé protokoly VPN, následně všechny dohromady a posléze byl otestován modul NAT. Daný modul je vždy spouštěn s parametrem `-p`, aby i po odpojení klienta nebo expiraci překladu zůstali identity zaznamenány.

7.2.1 VPN - PPTP

PPTP server *pptpd* byl umístěn na virtuálním počítači fedora_15 a klient se připojoval z virtuálního počítače fedora_18 s IP adresou *10.11.12.2*. Cílem tohoto testu bylo prověřit propojení klientovi IP adresy s přidělenou IP adresou pomocí uživatelova loginu.

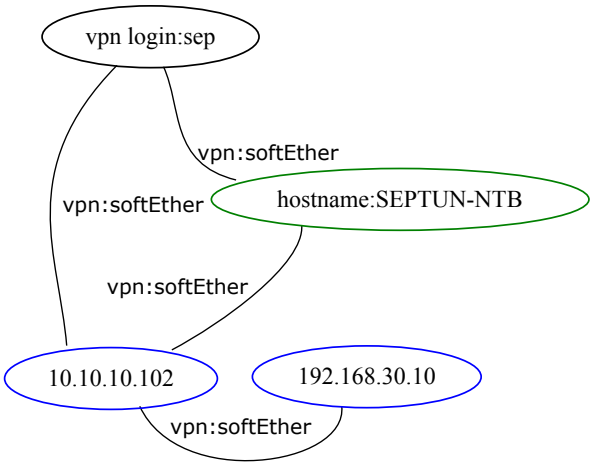


Obrázek 7.2: PPTP graf

Na obrázku 7.2 lze vidět připojení uživatele *sep* s IP adresou *10.11.12.2* a přidělenou adresou od PPTPD serveru *192.168.1.10*.

7.2.2 VPN - L2TP (SoftEther)

SoftEther VPN server u tohoto testu byl umístěn na virtuálním počítači fedora_18 a klient se připojoval z reálného notebooku.

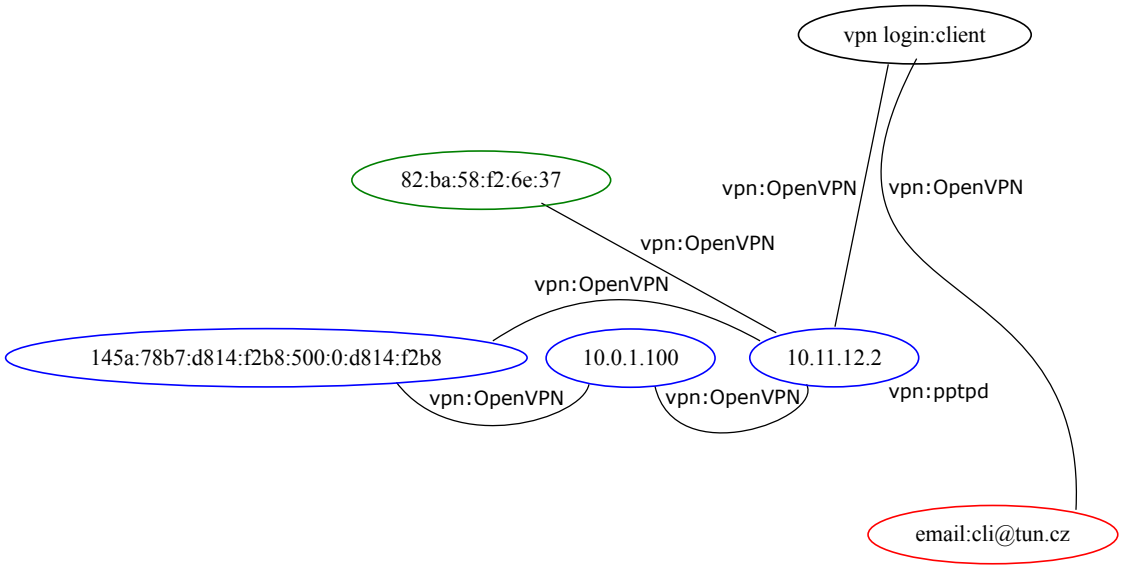


Obrázek 7.3: SoftEther graf

Na obrázku 7.3 lze vidět, že SoftEther poskytuje navíc informaci o jméně počítače ze kterého se uživatel přihlašuje, *SEPTUN-NTB*, jako ostatní servery pak zaznamenává IPv4 adresu, *10.10.10.102*, uživatelské jméno, *sep*, a přidělenou IPv4 adresu *192.168.30.10*.

7.2.3 VPN - OpenVPN

U tohoto testu byl OpenVPN server umístěn opět na virtuálním počítači fedora_15 a a klient se připojoval z virtuálního počítače fedora_18 s IP adresou *10.11.12.2*. Cílem bylo otestování správného propojení identit, MAC adresy s IP adresou virtuálního stroje, IP adres a informací o uživateli.



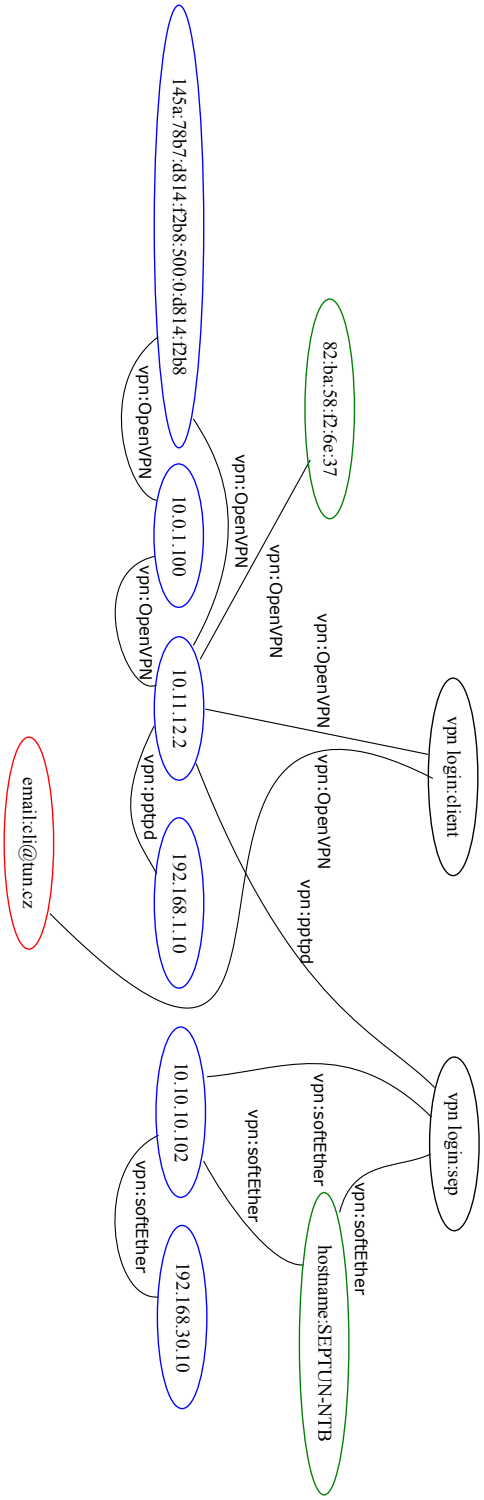
Obrázek 7.4: OpenVPN

Na obrázku 7.4 lze vidět, že OpenVPN, stejně jako SoftEther, poskytuje oproti pptpd identifikátory navíc. Tedy že uživatel se jménem *client* a e-mailem *cli@tun.cz* z certifikátu použitého pro přihlášení se přihlašoval z počítače s IPv4 adresou *10.11.12.2* a MAC adresou *82:ba:58:f2:6e:37*. Dále také vidíme, že OpenVPN server klientovi přidělil IPv4 adresu *10.0.1.100* a IPv6 adresu *145a:78b7:d814:f2b8:500:0:d814:f2b8*.

7.2.4 VPN - dohromady

Na obrázku 7.5 pak pěkně vidíme, když zpracujeme informace se všech serverů dohromady, jak například vpn login *sep* propojuje dva různé VPN servery a dvě různé IP adresy. Dá se tedy předpokládat, že uživatel se přihlašoval z odlišných počítačů nebo využívá notebook a ten se připojoval přes různé poskytovatele.

Dále také lze vidět, že uživatel s IP adresou *10.11.12.2* se připojoval k různým VPN serverům s odlišným uživatelským jménem.



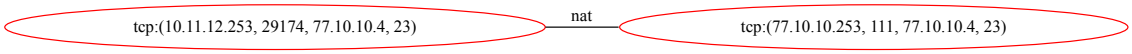
Obrázek 7.5: Graf VPN serverů dohromady

7.2.5 NAT - překlad portu

V tomto testu využijeme Cisco směrovač a překládání portů. Cílem je ověřit, že modul správně zpracuje záznamy poslané ze směrovače na Syslog server a korektně přiřadí čísla portů. Abychom mohli ověřit správnost je důležitý výpis ladících informací přímo z Cisco směrovače:

```
*Mar 1 00:03:41.847: NAT: [0] Allocated Port for 10.11.12.253 -> 77.10.10.253:
wanted 29174 got 111
*Mar 1 00:03:41.847: NAT: i: tcp (10.11.12.253, 29174) -> (77.10.10.4, 23) [0]
*Mar 1 00:03:41.851: NAT: s=10.11.12.253->77.10.10.253, d=77.10.10.4 [0]
```

Jedná se o komunikaci z lokální sítě do veřejné na IP adresu 77.10.10.4 port 23.



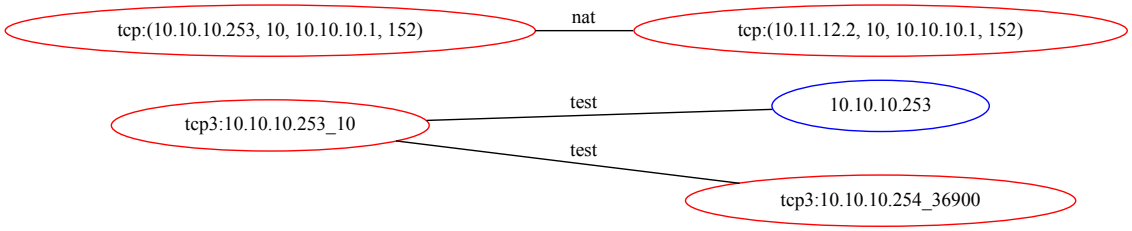
Obrázek 7.6: Graf překladů síťových adres s porty - PAT

Na obrázku 7.6 lze vidět překlad adresy na síťové i transportní vrstvě, tedy PAT. Adresa *10.11.12.253* je přeložena na *77.10.10.253* a v tomto případě je přeloženo i číslo portu z *29174* na *111*. Lze také vidět, že obrázek odpovídá výstupu ze směrovače, modul tedy zpracoval vše korektně.

7.2.6 NAT - propojení pomocí IP adresy

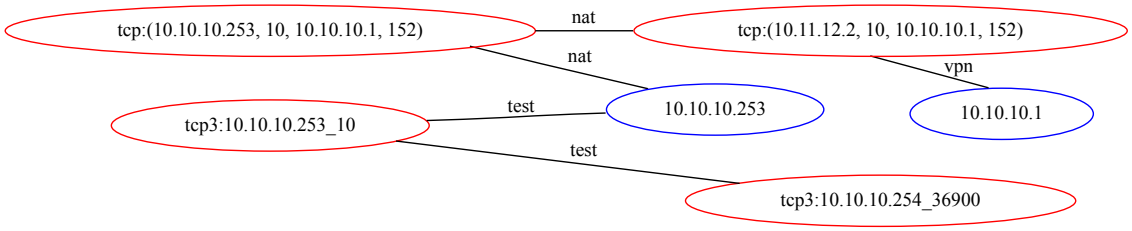
Nyní otestujeme funkčnost parametru `--IPconnect`, tedy zda se identity propojí pomocí IP adres obsažených v tcp pěticích.

Nejprve otestujeme případ bez parametru, identity se tedy propojit nemají.



Obrázek 7.7: TCP pětice se nepropojí přes IP adresu

Pro propojení identit spustíme modul s parametrem `--IPconnect`.



Obrázek 7.8: TCP pětice propojená přes IP adresu

Jak lze vidět na grafu 7.7, bez parametru se tcp pětice se stejnou IP adresou nespojí. Ale pokud zadáme parametr `--IPconnect`, tcp pětice se spojí s IP adresou, která je v tcp pětici obsažena.

Kapitola 8

Závěr

Cílem této práce bylo popsat systém pro zákonné odposlechy, principy tunelování síťového provozu, překlad síťových adres, navrhnout a implementovat rozšíření mechanismu pro monitorování identity v projektu Sec6Net a nakonec implementaci otestovat.

S architekturou, rozhraním, jednotlivými částmi a hlavně funkcí dynamické identity systému SLIS pro zákonné odposlechy jsme se seznámili v kapitole 2.

Některé z dnes používaných tunelovacích protokolů jsou uvedeny v kapitole 3. Je naznačen základní princip tunelování a následně popsány protokoly a programy využívané pro přenášení dat po internetu pomocí VPN. Také jsme se dozvěděli, jak lze zpřístupnit IPv6 na zařízeních s IPv4, a to pomocí přechodových mechanismů.

Překlad síťových adres, jeho varianty NAT 1:1, překlad síťových adres s překladem portů, byl popsán v kapitole 4. Byl zde také představen protokol Syslog pro posílání systémových zpráv na vzdálené zařízení a překlad adres na směrovačích Cisco.

V kapitole 5 bylo navrženo rozšíření mechanismu pro monitorování identity o dva moduly, které monitorují a analyzují logovací záznamy. Jednotlivé moduly se starají buď o identitu ve VPN sítích nebo o překlad adres. Nalezneme zde jakým způsobem moduly poznávají identity a kdy generují zprávy pro jádro IRI-IIF.

Parametry jednotlivých modulů a princip fungování je nastíněn v kapitole 6. Jsou zde popsány třídy reprezentující jednotlivé moduly a také problémy některých VPN programů a jejich řešení.

Testovací prostředí emulátoru sítí GNS3 a výsledky testů jsou popsány v kapitole 7. Jsou zde uvedeny výstupy testování modulů v podobě grafů, které zobrazují propojení identit vyskytujících se v počítačové síti.

V této práci jistě nebyly popsány všechny protokoly a metody tunelování, a tak ji lze dále rozšířit, například o další tunelovací mechanismy na aplikační úrovni pomocí SSL/TLS. Je také možné zaměřit se na bezpečností rizika tunelování, což může být například procházení paketů skrz IDS/IPS systémy nebo tunelování pomocí DNS protokolu, které lze využít pro obejití Captive portálu u veřejných sítí.

Literatura

[1] COUNCIL RESOLUTION of 17 January 1995 on the lawful interception of telecommunications (96/C 329/01). 1996.

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

[3] Farinacci, D.; Li, T.; Networks, P.; aj.: Generic Routing Encapsulation (GRE) [online]. `http://tools.ietf.org/html/rfc2784`, 2000-05 [cit. 2015-01-10].

[4] Fogarty, S.: GNS3 Network Simulator Raises Its Game [online]. `http://www.networkcomputing.com/networking/gns3-network-simulator-raises-its-game/d/d-id/1319279`, 2015-02-03 [cit. 2015-05-10].

[5] GNS3: Graphical Network Simulator 3 [online]. `http://www.gns3.com/`, [cit. 2015-05-08].

[6] Kent, S.; Corp, B.; Atkinson, R.: Security Architecture for the Internet Protocol [online]. `http://tools.ietf.org/html/rfc2401`, 1998-11 [cit. 2015-01-10].

[7] Microsoft TechNet: Protokoly tunelového propojení VPN [online]. `http://technet.microsoft.com/cs-cz/library/cc771298.aspx`, [cit. 2015-01-17].

[8] OpenVPN documentation: Security Overview - OpenVPN cryptographic layer [online]. `openvpn.net/index.php/open-source/documentation/security-overview.html`, 2008-11-01 [cit. 2015-01-14].

[9] Podermanski, T.; Grégr, M.; Švéda, M.: User Identification in IPv6 Network [online]. `http://6lab.cz/article/user-identificationq-in-ipv6-network`, 2012-01 [cit. 2015-01-17].

[10] 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?id=9829

[11] Polčák, L.; Martínek, T.; Hranický, R.; aj.: Zákonné odposlechy v moderních sítích - Shrnutí výsledků skupiny pro zákonné odposlechy projektu Moderní prostředky pro boj s kybernetickou kriminalitou na Internetu nové generace. Technická zpráva, 2014.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=10788

[12] Polčák, L.; Martínek, T.; Hranický, R.; aj.: Správa identity z projektu Sec6Net [online]. http://www.fit.vutbr.cz/ ipolcak/prods.php?id=399, 2014 [cit. 2015-05-02].

[13] Sbírka zákonu č. 127/2005: Zákon o elektronických komunikacích. 2005.

[14] Srisuresh, P.; Networks, J.; Egevang, K.; aj.: Traditional IP Network Address Translator (Traditional NAT) [online]. http://tools.ietf.org/html/rfc3022, 2001-01 [cit. 2015-01-12].

[15] University of Tsukuba Japan: SoftEther Project [online]. http://www.softether.org/, [cit. 2015-05-08].

[16] Veselý, V.; Grégr, M.: CCNP ROUTE - Module 7 - WAN Technologies [online]. https://netacad.fit.vutbr.cz/ccnp/route/ROUTE_M7_ENG.pdf, 2012-09-09 [cit. 2014-01-06].

Příloha A

Obsah CD/DVD

Přiložené optické médium obsahuje následující soubory a adresářovou strukturu.

Soubor/adresář	**Obsah**
technicka_zprava.pdf	tento dokument v elektronické podobě
tex/	zdrojové soubory LaTeXtohoto dokumentu
src/	zdrojové soubory systému pro určování identity
mysrc/	pouze nově vytvořené nebo upravené soubory vzniklé v této práci
VM/	VirtualBox virtuální stroj se systémem Ubuntu

A.1 Spuštění SIMS a vygenerování grafů na virtuálnímu stroji Ubuntu

Přiložené DVD obsahuje virtuální počítač se systémem Ubuntu. Systém obsahuje program SIMS a je zde nainstalován také server pptpd a SoftEther server.

Před spuštěním SIMS je důležité v souboru *iri-collector.ini* definovat jaké moduly se mají spustit a zadat vstupní soubory pro moduly NAT a VPN.

Pro spuštění zadáme příkaz: `./sims.sh start [debug|info]`. Příkazem se spustí program a moduly definované v *iri-collector.ini*.

Pro vygenerování a zobrazení grafu je vytvořen skript `./create-viewGraph.sh`. Skript vygeneruje obrázek ve formátu SVG a s defaultním názvem graph.svg, dá se zadat parametr určující název výsledného svg obrázku, např. `./create-viewGraph.sh graf.svg`.

Spuštění VPN serveru provedeme skriptem `start-pptpd.sh` nebo `start-SoftEther.sh`

Příloha B

Konfigurace VPN

B.1 pptpd server

```
> yum/apt-get install pptpd
> vim /etc/pptpd.conf
   localip 10.11.12.1
   remoteip 10.1.1.2-254
> vim /etc/ppp/options.pptpd
   name pptpd
   ms-dns 10.11.12.254
   ms-dns 8.8.8.8
   refuse-pap
   refuse-chap
   refuse-mschap
   require-mschap-v2
   require-mppe-128
   require-mppe
> vim /etc/ppp/chap-secrets
   septun pptpd heslo *
   michal pptpd heslo2 *
> vim /etc/sysctl.conf nebo /etc/sysctl.d/30-ipforward.conf
   net.ipv4.ip_forward = 1
> sysctl -p nebo > sysctl --system
> service pptpd start
```

B.2 OpenVPN server

```
> apt-get install openvpn openssl
> cd /etc/ssl/
> mkdir mojeCA mojeCA/certs mojeCA/crl mojeCA/newcerts mojeCA/private
> touch /etc/ssl/mojeCA/index.txt
> echo 01 > /etc/ssl/mojeCA/serial
> cd /etc/ssl/mojeCA
> openssl req -new -x509 -nodes -out cacert.pem -keyout cakey.pem -days 3650
> mv cacert.pem certs/ && mv cakey.pem private/
```

Nastavení cest k souborům v `/etc/ssl/openssl.cnf` v sekci `[ CA_default ]`

```
> mkdir server && cd server
> openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
> openssl ca -in request.pem -out cert.pem
> openssl dhparam -out /etc/ssl/mojeCA/dh1024.pem 1024
```

¿ vim /etc/openvpn/vpn_server.conf

```
   mode server
   tls-server
   dev tap0
   port 1194
   ifconfig 10.0.1.1 255.255.255.0
   ifconfig-pool 10.0.1.100 10.0.1.200 255.255.255.0
   duplicate-cn
   proto udp
   ca /etc/ssl/mojeCA/certs/cacert/cacert.pem
   cert /etc/openvpn/cert.pem
   key /etc/openvpn/key.pem
   dh /etc/ssl/mojeCA/dh1024.pem
   log-append /var/log/openvpn
   status /tmp/vpn.status 10
   user root
   group root
   comp-lzo
   verb 3
   keepalive 1 220
```

```
> /etc/init.d/openvpn start
> mkdir client && cd client
> openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
> openssl ca -in request.pem -out cert.pem
```

B.3 OpenVPN klient

¿ vim /etc/openvpn/vpn_client.conf

```
   remote 1.2.3.4 ### IP adresa serveru
   tls-client
   dev tap
   pull
   mute 10
   ca /etc/openvpn/cacert.pem
   cert /etc/openvpn/cert.pem
   key /etc/openvpn/key.pem
   comp-lzo
   verb 3
```

Do `/etc/openvpn/` umístíme klientský certifikát, klíč, certifikát certifikační autority. `> cd /etc/openvpn && openvpn --config ./vpn_client.conf`.

Příloha C

Konfigurace NAT a Syslog

C.1 NAT na Cisco směrovači

```
interface ethernet 0
   ip address 10.10.10.1 255.255.255.0
   ip nat inside
interface serial 0
   ip address 172.16.10.64 255.255.255.0
   ip nat outside
ip nat pool <NAZEV> 172.16.10.1 172.16.10.63 prefix 24
access-list 7 permit 10.10.10.0 0.0.0.31
access-list 7 permit 10.10.20.0 0.0.0.31
ip nat inside source list 7 pool <NAZEV> [overload]
ip nat inside source static tcp 172.16.10.64 80 10.10.10.254 80 extendable
```

C.2 Syslog na Cisco směrovači

```
debug ip nat detailed
logging trap debugging
logging facility syslog
logging 10.11.12.1
```

C.3 Syslog server

C.3.1 Syslog-ng

```
> vim /etc/syslog-ng/syslog-ng.conf
   source li_net { udp(ip(10.11.12.1) port(514) ); };
   destination ciscoLIS { file("/var/log/myLog.log"); };
   log { source ( li_net ); destination ( ciscoLIS ); };
```

C.3.2 Rsyslog

```
> vim /etc/rsyslog.conf
   $ModLoad imudp
```

```
$UDPServerRun 514
syslog.debug /var/log/myLog.log
```