VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE PROVOZU SKYPE POMOCÍ DAT NETFLOW

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. PATRIK ŠEBEŇ
AUTHOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE PROVOZU SKYPE POMOCÍ DAT NETFLOW

IDENTIFYING SKYPE TRAFFIC USING NETFLOW DATA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. PATRIK ŠEBEŇ
AUTHOR

VEDOUCÍ PRÁCE — Ing. MATĚJ GRÉGR
SUPERVISOR

BRNO 2012

Abstrakt

NetFlow je sieťový protokol bežne používaný k zberu informácií o IP tokoch. Avšak existuje možnosť, ako použiť tieto zachytené dáta k identifikácií klientov v Skype komunikácií. Táto práca pojednáva o identifikácií určitých vzorov v Skype protokole a ich detekcií v NetFlow dátach. Týmto spôsobom sme schopný identifikovať nódy a supernódy v sieti Skype.

Abstract

NetFlow is a network protocol commonly used for collectiong IP traffic information. But there is a way to use this collected data for indentifying clients in Skype communication. This paper describes identifiable patterns in Skype protocol and how to find them in NetFlow data. This way we can identify nodes and supernodes in Skype network.

Klíčová slova

NetFlow, Skype, supernóda

Keywords

NetFlow, Skype, supernode

Citace

Patrik Šebeň: Detekce provozu Skype pomocí dat NetFlow, diplomová práce, Brno, FIT VUT v Brně, 2012

Detekce provozu Skype pomocí dat NetFlow

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Matěje Grégra

. . . . . .
Patrik Šebeň
23. května 2012

Poděkování

Chcel by som sa poďakovať vedúcemu práce Ing. Matějovi Grégrovi za to, že my venoval svoj čas a odbornú pomoc pri vedení tejto práce

© Patrik Šebeň, 2012.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**
1.1 Cieľ diplomovej práce . . . 3
1.2 Členenie práce . . . 3

2 IP toky a ich monitorovanie **5**
2.1 IP tok . . . 5
2.1.1 Spôsob monitorovania . . . 5
2.1.2 Využitie monitorovania datových tokov . . . 6
2.2 NetFlow . . . 6
2.2.1 Verzie NetFlow . . . 6
2.2.2 Architektúra NetFlow . . . 7
2.3 sFlow . . . 9
2.4 IPFIX . . . 9

3 Skype **10**
3.1 História [12] . . . 10
3.2 Architektúra Skype siete . . . 10
3.3 Skype protokol [4] . . . 12
3.3.1 Spustenie aplikácie . . . 12
3.3.2 Login . . . 12
3.3.3 Supernódy . . . 13
3.3.4 Detekcia NATu a firewallu . . . 13
3.3.5 Nadviazanie spojenia . . . 13
3.3.6 Zabezpečenie . . . 14
3.3.7 Kodek . . . 14

4 Metódy detekcie účastníkov Skype siete **15**
4.1 Metodológia . . . 15
4.1.1 UDP próba . . . 15
4.1.2 TCP handshake . . . 16
4.1.3 TCP autentizácia . . . 16
4.1.4 Detekčný algoritmus . . . 17

5 Realizácia riešenia **19**
5.1 Nástroje . . . 19
5.1.1 Nfdump [1] . . . 19
5.1.2 Nmap [2] . . . 19
5.1.3 Wireshark [13] . . . 20

5.2 Implementácia 20

6 Testovanie **23**
6.1 Supernódy 23
6.2 Časová náročnosť 24

7 Zmena architektúry Skype siete **26**
7.1 MSnóda 26
7.2 Detekcia MSnód 28
7.3 Testovanie 28

8 Budúcnosť Skype siete **33**
8.1 Možné rozšírenia 34

9 Záver **35**

A Použitie supernode detektora **38**

B MSnóda **39**

C Zoznam Msnód **42**

D Skype detekčný skript **43**

Kapitola 1

Úvod

Rýchly rozvoj internetu a neustále sa zväčšujúca potreba komunikácie umožnila vznik mnohých aplikácii pre hlasové, prípadne video spojenie dvoch alebo viacerých uzlov. Tento spôsob sa teší svojej popularite aj vďaka nižším nákladom oproti klasickej komunikácií pomocou telefónnej siete, za predpokladu, že celkový prevolaný čas prekročil istú hranicu prípadne sa komunikovalo mimo hraníc daného štátu.

Skype môžeme dnes označiť ako jednu z najpopulárnejších aplikácií pre audio/video komunikáciu po internete. Je používaný nie len v bežnej domacnosti, ale hlavne vďaka možnosti konferenčných hovorov aj vo firemnej sfére. Medzi ďalšie výhody tejto aplikácie patrí aj možnosť nadviazať hovor so stanicou v klasickej telefónnej sieti (samozrejme táto služba je už spoplatnená) a aj dostupná bezpečnosť oproti odpočúvaniu hovorov, ktorú zaručuje samotný charakter siete a použité šifrovanie (vzhľadom na to, že dáta prechádzajú rôznymi uzlami internetu je to vlastnosť, ktorej treba venovať zvýšenú pozornosť).

1.1 Cieľ diplomovej práce

Napriek nevedomosti o presnom fungovaní aplikácie bolo možné pomocou metód reverzného inžinierstva získať pomerne dobrú predstavu na základe, ktorej sme schopní detekovať určité operácie, ktoré nám umožnujú monitorovať prípadnú komunikáciu, čo je vlastne hlavným cieľom tejto práce. Zo získaných NetFlow dát sa pokúsime identifikovať účastníkov Skype siete a ich vzájomnú interakciu. Ďalej overíme presnosť predchádzajúcich uskutočnených meraní, ktoré nepracovali s najaktuálnejšími dátami, čomu sa budeme snažiť vyhnúť.

1.2 Členenie práce

Práca je členená do niekoľkých kapitol, v ktorých rozoberiem podrobnejšie skúmanú problematiku. Kapitola 2 sa venuje monitorovaniu IP tokov, popisu protokolu NetFlow a jeho alternatív. Skype protokol a jeho podrobnejšie fungovanie je popísane v Kapitole 3, kde je opísaná aj architektúra siete a niektoré zo špecifických vlastností Skypu. Kapitola 4 je venovaná metódam detekcie Skype komunikácie v NetFlow dátach. Je opísaný priebeh jednotlivých operácií, ktoré nám umožnujú identifikovať účastníkov Skype komunikácie. Popis samotnej implementácie a použitých nástrojov je možné nájsť v kapitole 5. Kapitola 6 obsahuje výsledky z meraní na aktuálnych NetFlow záznamoch z siete VUT. Je zachytená štatistika výskytu klientov a supernód za testované obdobie a časová náročnosť detekcie. Vzhľadom na zmeny vykonané na sieti a protokole Skype počas písania práce som v kapi-

tole 7 zopakoval merania na NetFlow záznamoch staršieho dáta a popísal zistené rozdiely. Taktiež som sa snažil popísať tieto zmeny a ich vplyv na celú sieť. V kapitole 8 sa venujem ďalšiemu smerovaniu siete Skype po zmene jej architektúry a popisujem možnosti rozšírenia tejto práce.

Kapitola 2

IP toky a ich monitorovanie

2.1 IP tok

IP tok (IP flow) je definovaný podľa RFC 3917 nasledovne: Tok je množina IP paketov prechádzajúcich cez pozorovací bod v sieti počas určitého časového intervalu. Všetky pakety patriace do rovnakého toku majú sadu spoločných vlastností. Každá vlastnosť je definovaná ako výsledok funkcie aplikovanej na nasledujúce hodnoty:

- Jedna alebo viac položiek z hlavičky paketu (napr. cieľová, zdrojová IP adresa), z transportnej hlavičky paketu (napr. cieľový port) alebo z aplikačnej hlavičky.
- Jedna alebo viac charakteristík paketu samotného (napr. MPLS návestie).
- Jedna alebo viac položiek odvodených od spracovania paketu (napr. "next hop"adresa, výstupné rozhranie).

Paket patrí do príslušného toku, ak sú splnené všetky vlastnosti daného toku.

Obrázek 2.1: IP tok [11]

2.1.1 Spôsob monitorovania

Informácie o tokoch sú zaznamenávané sondami, ktoré sú vhodne umiestnené na určitých častiach infraštruktúry siete. Sonda môže byť samotný hardwarový prvok alebo funkciu sondy podporuje smerovač, prípadne prepínač. Informácie zachytené sondou sú odosielané a uchovávané najčastejšie v kolektoroch, odkiaľ možu byť následne dostupné k analýze.

Samotný proces vytvárania záznamu o toku pozostáva z informácii o časovom údaji jeho vzniku, dĺžke trvania, počte prenesených paketov (prípadne ich veľkostí) a ďalších informácií.

2.1.2 Využitie monitorovania datových tokov

Využitie nazhromaždených dát je skutočne všestranné:

- *Monitorovanie siete* - aktuálne získané dáta môžeme v takmer reálnom čase analyzovať a vďaka tomu monitorovať toky prechadzajúce jednotlivými smerovačmi. V prípade vzniku problému na sieti sme schopní včasne odstrániť odhalené poruchy.
- *Monitorovanie a analýza aplikácií* - Zo získaných informácií sme schopní zostaviť štatistiku využívania jednotlivých aplikácií v konkrétnych časových úsekoch a podľa toho upraviť topológiu siete k jej lepšiemu výkonu.
- *Monitorovanie a analýza užívateľov* - umožnuje získať informácie o aktivitách konkrétnych užívateľov na sieti a predchádzať tak potenciálnemu bezpečnostnému ohrozeniu. Taktiež máme lepší prehľad o rozložení zaťaženia siete a sme lepšie schopní predchádzať nedostupnosti niektorých služieb.
- *Účtovanie* - Charakter ukladaných informácií (zdrojová a cieľová adresa, počet prenesených dát, čas, porty, služby) nám umožnuje zostaviť podrobný prehľad komunikácie konkrétneho uzlu v sieti za určité časové obdobie. Internetoví poskytovatelia používajú tieto dáta na spoplatnenie svojich služieb (najčastejšie je to objem prenesených dát).
- *Plánovanie a analýza siete* - dáta sme schopní použiť k optimalizácií strategického plánovania siete (kto s kým komunikuje, ako často, ktorými prvkami komunikácia prechádza). Cieľom tohto procesu je minimalizovať cenu sieťových operácií a dosiahnuť maximálny výkon siete.
- *Ukladanie dát* - Získané informácie môžu byť uložené k neskoršej analýze a prípadnému generovaniu grafov alebo štatistík (napr. vyťaženosť liniek). Taktiež je vhodné mať informácie o tom kto, kedy, ako dlho a s kým komunikoval pri prípadnej bezpečnostnej analýze alebo optimalizácií.

2.2 NetFlow

NetFlow protokol je protokol vyvinutý firmou Cisco pre prenos záznamov o sieťových tokoch, pôvodne ako doplnková služba k smerovačom Cisco. Jeho podrobnú špecifikáciu je možné najsť v [5], vďaka čomu bola možná aj jeho implementácia na iných systémoch ako Cisco IOS (napr. Smerovače Juniper, FreeBSD, OpenBSD). Existuje niekoľko verzií protokolu, z ktorých sú dnes najpoužívanejšie verzie 5 a 9. Jednotlivé rozdiely vo verziách budú popísane ďalej. Posledná verzia protokolu (v.9) sa stala základom protokolu IPFIX, vytvoreného IETF, ktorý by mal nahradiť v blízkej dobe NetFlow ako štandard pre monitorovanie tokov.

2.2.1 Verzie NetFlow

Protokol NetFlow je aktuálne k dispozícií vo verzii 9, rozdiely medzi jednotlivými verziami sú nasledovné :

- *Verzia 1* - prvá uvedená verzia, ukladali sa informácie o zdrojovej/cieľovej IP, o zdrojovom/cielovom porte, o IP protokole, ToS, vstupné/výstupné rozhranie, počet paketov, čas a bitový súčet TCP príznakov. Nepoužívali sa poradové čísla.

- *Verzia 2 až 4* - interné verzie, ktoré nikdy neboli zverejnené.
- *Verzia 5* - pridané poradové čísla, kvôli detekcií stratených dát.
- *Verzia 6* - pridaná podpora pre tunelovanú prevádzku.
- *Verzia 7* - špecialna podpora pre prepínače.
- *Verzia 8* - podpora agregácie, jednotlivé agregačné schémy sú v Tabuľke 2.1.
- *Verzia 9* - zavedenie šablón ako kompromis medzi granularitou a agregáciou. Šablóna poskytuje flexibilnejší spôsob sledovania tokov, vďaka možnosti voľby množiny sledovaných polí.

8.1	vstupné/výstupné rozhranie, zdrojový/cilový AS
8.2	protokol, port
8.3	vstupné rozhranie, zdrojový prefix
8.4	výstupné rozhranie, cielový prefix
8.5	8.3 + 8.4
8.6	zdrojová IP, ToS
8.7	vstupné/výstupné rozhranie, zdrojová/cieľová IP, ToS
8.8	8.7 + vstupný/výstupný port
8.9	8.1 + ToS
8.10	8.2 + ToS
8.11	8.3 + ToS
8.12	8.4 + ToS
8.13	8.5 + ToS
8.14	8.3 + ToS + port

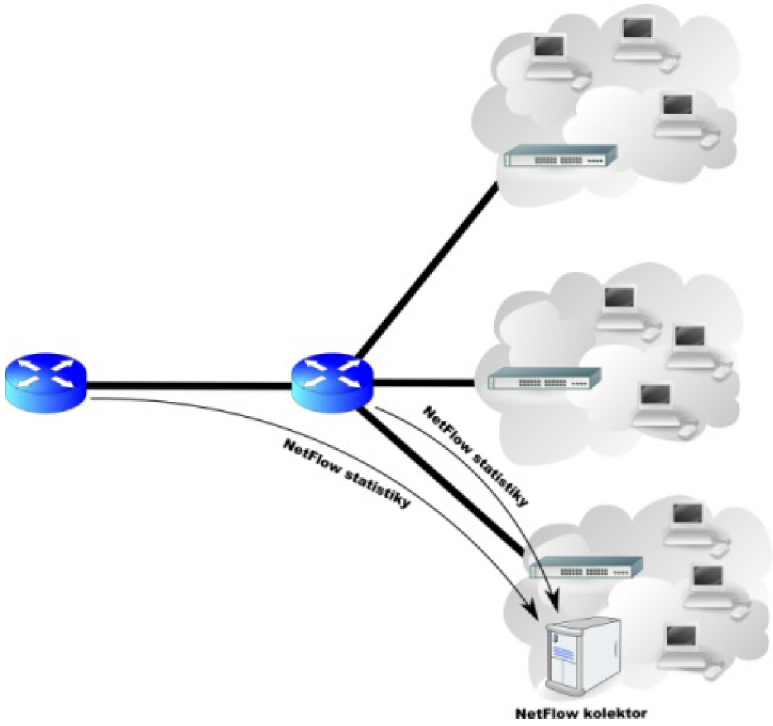
Tabulka 2.1: Agregačné schémy NetFlow v8

2.2.2 Architektúra NetFlow

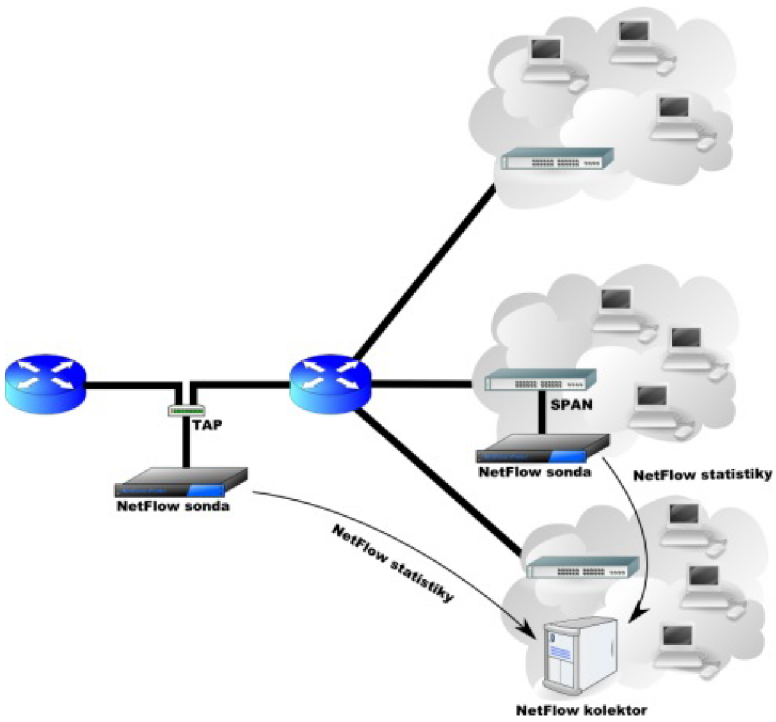
Architektúra NetFlow sa skladá z nasledujúcich prvkov:

- *NetFlow Exportér* - zariadenie, ktoré monitoruje prechádzajúce pakety a vytvára z nich IP tok. Z NetFlow záznamov je zostavený paket, ktorý je exportovaný do kolektora. Funkciu exportéra vykonáva buď smerovač, prepínač popri svojej primárnej činnosti alebo sú do siete vkladané sondy, ktoré výkonávajú túto funkciu primárne.
- *NetFlow Kolektor* - prijíma a spracováva pakety z jedného alebo zvyčajne viacerých exportérov a ukladá ich.

Do siete je bežne umiestnených niekoľko exportérov a kolektor. Podľa vyťaženia siete sa ako exportér použije buď smerovač (Obrázok 2.2) alebo pasívna sonda (Obrázok 2.3). Je taktiež vhodné využívať dedikované spojenia medzi sondami a kolektorom.



Obrázek 2.2: Architektúra využívajúca smerovače [11]



Obrázek 2.3: Architektúra využívajúca pasívne sondy [11]

2.3 sFlow

sFlow je alternatívny protokol k protokolu NetFlow popísaný v [8]. Je určený predovšetkým na použitie vo vysokorýchlostných sieťach vzhľadom na jeho možnosť vzorkovania (v NetFlow až od verzie 5). Používajú sa dva druhy vzorkovania, a to buď náhodné vzorkovanie paketov prípadne operácií aplikačnej vrstvy alebo sa v pravidelných časových intervaloch odosielajú "countre". Získané vzorky sú odosielané ako sFlow datagramy na centrálny server (sFlow collector), kde sa vykonáva výsledná analýza. Vzorky sú odosielané vo väčšom množstve ako UDP paket a prípadna strata paketu nie je ošetrená, keďže príliš neovplyvní výsledok. Datagram obsahuje informácie o verzií protokolu, IP zdroja, sekvenčné číslo, počet vzoriek a samotné vzorky.

2.4 IPFIX

IPFIX je v podstate ďalšia verzia protokolu NetFlow (ako základ bol použitý NetFlow verzie 9) avšak navrhnutá IETF ako nový štandard, vzhľadom na potrebu spoločného nezávislého formátu pre záznam IP tokov zo smerovačov, sond a iných sieťových zariadení. Definuje, ako budú informácie formátované a odosielané zo zberačov do kolektora. Pre prenos dát je preferovaný protokol SCTP, avšak podporované sú aj TCP a UDP. Zatiaľ nie je príliš rozšírený, avšak postupne sa naň prechádza. Popísaný je v [9].

Kapitola 3

Skype

Skype môžeme považovat za najpopulárnejšiu a najpoužívanejšiu VoIP aplikáciu v dnešnej dobe. Svojich užívateľov si získala predovšetkým príjemným vzhľadom, jednoduchým ovládaním, bezplatnými audio/video hovormi (pokiaľ sa nejedná o hovor do klasickej telefónnej siete) a podporou takmer každeho, či už mobilného (Android, iOS) alebo bežného (Windows, Mac OS X) operačného systému. V tejto kapitole by som sa ďalej rád venoval histórií, architektúre siete a fungovaniu Skype protokolu.

3.1 História [12]

V roku 2003 založil Janus Friis z Dánska a Niklas Zennström zo Švédska firmu Skype Limited s centrálou v Luxembursku. Na vývoji samotného produktu sa podieľali Ahti Heinla, Priit Kasesalu a Jaan Tallinn (všetci z Estónska), ktorí sa už predtým podieľali na vzniku peer-to-peer aplikácie Kazaa na zdieľanie multimediálneho obsahu.

Pôvodne mala byť aplikácia pomenovaná Sky peer-to-peer, čo bolo neskôr skrátené na Skyper, avšak vzhľadom na to, že doména s týmto názvom bola už obsadená vypustilo sa posledné r a tým sme sa dopracovali k dnešnému názvu Skype.

Vzhľadom na obrovský potenciál aplikácie bolo len otázkou času, kedy vzbudí záujem zo strany väčších investorov. V októbri roku 2005 ju odkúpil Ebay Inc. A následne v 2009 sa stal 65% vlastníkom Silver Lake Partners. Dnes je vlastníkom korporácia Microsoft, ktorá spravila zo Skypu jednu zo svojich divízii.

3.2 Architektúra Skype siete

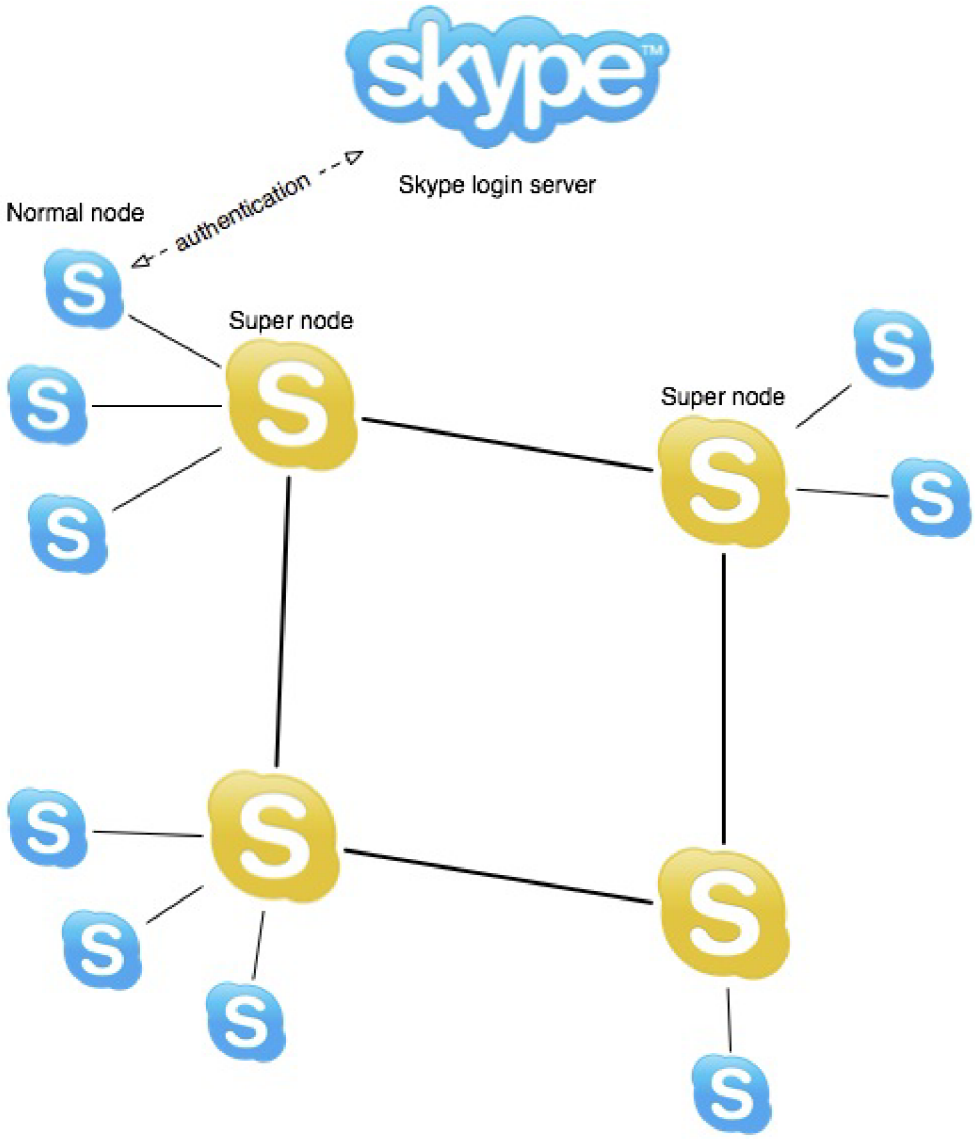
Hlavnou výhodou Skype siete je jej peer-to-peer architektúra, ktorá narozdiel od iných klient-server alternatív umožnuje minimalizovať náklady na jej fungovanie. Skype sieť pozostáva z nasledujúcich prvkov:

- *Skype nóda* - je každý bežný užívateľ, u ktorého je spustená aplikácia a je pripojený do siete.
- *Skype login server* - je jeden z mála centralizovaných prvkov Skype siete, ktorý má na starosti prihlasovanie užívateľov do siete.
- *Skype supernóda* - je obyčajná Skype nóda povýšená do stavu supernódy po splnení určitých predpokladov (dostupný výpočetný výkon, rýchlosť linky, dlhšia doba

bežiacej aplikácie bez prerušenia, verejná IP adresa). V tomto režime môže slúžiť užívateľom, ktorí nie sú schopní nadviazať navzájom spojenie (NAT, firewall) ako sprostredkovateľ komunikácie. Taktiež môže pracovať ako login server a uľahčiť tým prácu centralizovaným prvkom siete. Skype ma niekoľko vlastných supernód, ktoré sú využívané pri novej inštalácií klienta a registrácií užívateľa do siete.

- *Skype gateway* - je hraničný bod medzi skype Sieťou, ktorý smeruje hovory do klasickej telefónnej siete.

Tento typ siete umožňuje rýchly rast, keďže väčšinu práce vykonávajú supernódy. Na druhú stranu sa niektorým užívateľom a firmám nepáči myšlienka, že sú to zrovna oni, ktorých výkon je uberaný a linka je zaťažovaná. Z tohto dôvodu sa často snažia administrátori obmedziť komunikáciu cez Skype nastaveniami firewallu.



Obrázek 3.1: Architektúra Skype siete

3.3 Skype protokol [4]

Skype protokol je proprietárny, firmou Skype používaný protokol, postavený na peer-to-peer architektúre. Špecifikácia protokolu nie je verejne dostupná (closed-source), preto všetky dostupné informácie o jeho fungovaní sú získané vďaka reverznému inžinierstvu. Hlavným cieľom reverzného inžinierstva bolo získať čo najviac informácií o jeho fungovaní, zabezpečení a prípadnom použití alternatívnych klientov. Pri zvážení istých skutočností zistíme, že obavy administrátorov nemusia byť až také neopodstatnené. Skype má v súčastnej dobe (september 2011) cca. 663 miliónov registrovaných užívateľov, ktorí majú u seba pustenú aplikáciu, o ktorej vnútornej štruktúre nemáme zatiaľ všetky potrebné informácie. Politika uzavretosti na jednej strane zťažuje odhalenie slabých miest útočníkmi v programe, no na druhej strane súčasne sťažuje aj ich včasné odhalenie špecialistami na bezpečnosť. Potenciálne riziko vzniku botnetu alebo prípadná možnosť fungovania aplikácie ako rootkitu nie je preto zanedbateľná. Avšak to nie je tematika, ktorou sa budem v tejto práci zaoberať.

3.3.1 Spustenie aplikácie

Po spustení Skype klienta sa odošle HTTP 1.1 GET request Skype serveru (skype.com), ktorého obsah záleží od toho, či sa jedná o úplne prvé spustenie aplikácie po inštalácii alebo už o opakované spustenie. V jednotlivých prípadoch je rozdiel v prvom riadku dotazu nasledovný :

- Reťazec "installed"je odoslaný pri prvom spustení.
- Reťazec "getlatestversion"je odoslaný pri opätovnom spustení a jeho účelom je zistiť dostupnosť novšej verzie aplikácie.

Po úspešnej kontrole aktuálnej verzie nasleduje login.

3.3.2 Login

Login je pravdepodobne jedna z najkritickejšich operácíi aplikácie. Je to proces autentizácie klienta svojim užívatelským menom a heslom voči Skype login servru. Ak by bol prístup k login servrom z nejakého dôvodu blokovaný, snaží sa klient posielať správy login servru cez supernódu. Následne sú s prítomnosťou klienta oboznámení užívatelia v zozname kontaktov klienta a je zistená prítomnosť NATu prípadne firewallu, ktorý by mohol obmedzovať následnú komunikáciu.

Aby sa klient mohol pripojiť do siete musí nadviazať TCP spojenie s niektorou supernódou z lokálneho zoznamu supernód (host cache), ktorý je neustále aktualizovaný. V zozname sú uložené IP adresy supernód, spárované s príslušným portom. Proces nadviazania spojenia prebieha v niekoľkých krokoch. Klient odošle UDP paket na niektorú adresu a port zo svojho zoznamu. Ak nedostane odpoveď v nejakom rozumnom časovom intervale (cca. 5sec.), skúsi poslať TCP paket na totožnú adresu a port. V prípade neúspechu posielame TCP paket na port 80 (HTTP port) a následne 443 (HTTPS port) vzhľadom na to, že je menšia pravdepodobnosť blokovania týchto portov. Tento proces je niekoľkokrát zopakovaný a pokiaľ sme neuspeli ani u jedného záznamu, login končí neúspešne. Ak dostaneme v ktoromkoľvek bode odpoveď, prebieha následná TCP komunikácia, ktorá určí, či nás príslušná supernóda akceptuje alebo nie.

Až po úspešnom pripojení nasleduje autentizačná procedúra s login servrom, ktorý je jeden z mála centralizovaných prvkov v skype sieti a okrem potvrdenia totožnosti klienta má nastarosti aj zabezpečenie unikátnosti uživatelského mena v celej sieti.

3.3.3 Supernódy

Zoznam supernód je vždy po štarte aplikácie inicializovaný centrálnymi Skype supernódami (bootstrap supernodes) a následne rozširovaný o novoobjavené(geograficky bližšie), ktoré sú pri následnej komunikácií uprednostňované. Pri počiatočnom nadväzovaní spojenia sa bežne kontaktuje viacero supernód a v prípade viacerých kladných odpovedí vyberieme jednu, s ktorou sme spojení, až kým neukončíme klienta alebo nestratíme spojenie so supernódou. Klient pravidelne posiela supernóde, na ktorú je pripojený refresh (keep alive) správy v intervale cca. jednej minuty.

3.3.4 Detekcia NATu a firewallu

Predpokladá sa, že klient je schopný detekovať svoju prítomnosť za firewallom prípadne NATom a následne podľa toho upraví spôsob komunikácie. Na detekciu je pravdepodobne použitá varianta protokolu STUN (Simple Traversal of UDP through NATs) a detekcia prebieha pri nasledujúcich udalostiach:

- pri výmene informácií s pripojenou supernódou,
- pri počiatočnom výbere supernódy počas komunikácie s väčším počtom supernód.

Tieto informácie sú pravidelne aktualizované a uchovávané na strane klienta.

3.3.5 Nadviazanie spojenia

Skype klient po spustení otvorí TCP a UDP porty, na ktorých očakáva prichádzajúcu komunikáciu. Čísla portov sú náhodne zvolené pri inštalácii klienta (prípadne sú použité porty 80 a 443 pre TCP komunikáciu). Pre signálovanie je použitá výhradne TCP komunikácia, na rozdiel od prenosu multimediálnych dát, kde je logicky zväčša použitý UDP protokol. Prenos dát prebieha na rozdielnom porte ako signálovanie.

Spôsob nadviazania spojenia medzi dvoma klientami ovplyvňuje viacero faktorov. V prípade, že sa užívatelia nemajú navzájom v zozname kontaktov, je nutné zahájiť za spolupráce supernódy vyhľadávanie. Ak sú klienti navzájom oboznámení s adresou volaného, môže nastať niekoľko prípadov:

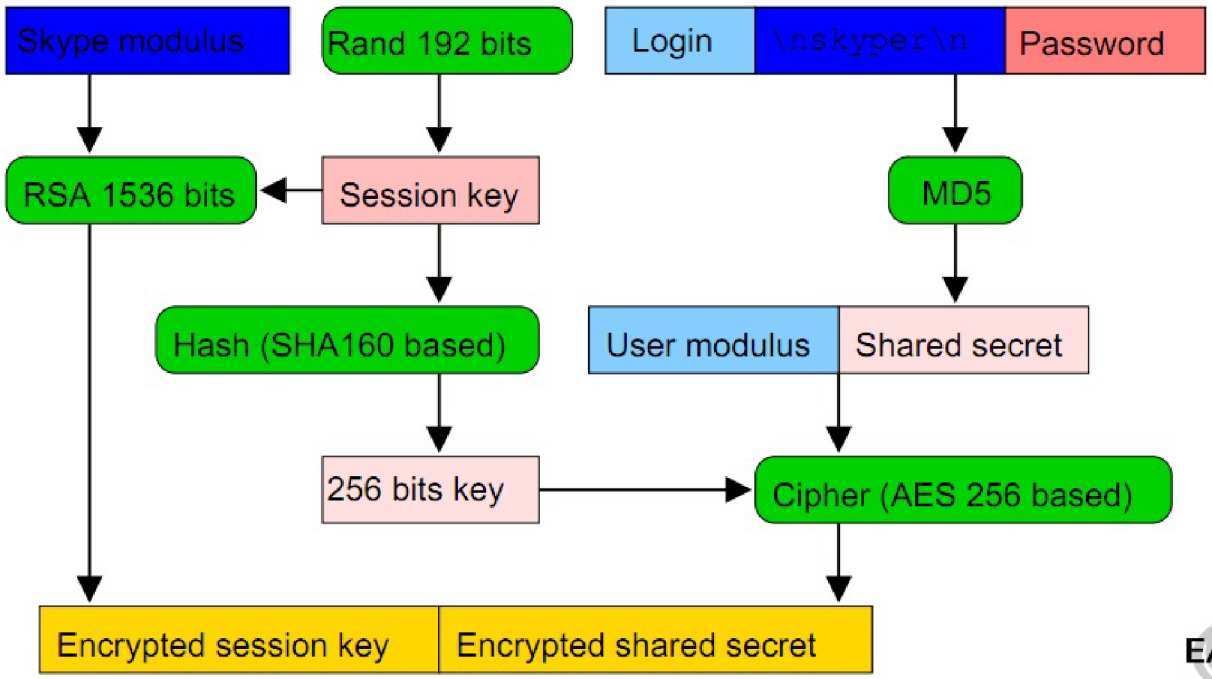
- *Obidvaja klienti majú verejnú adresu a nie su ničím obmedzení* - nadviaže sa TCP spojenie štýlom "chalenge-response"a po určitom počte správ je hovor nadviazaný.
- *Jeden klient je za NATom* - je kontaktovaná supernóda neprístupného klienta, ktorá pomáha nadviazať spojenie a sú ňou prenášané aj multimediálne dáta.
- *Obaja klienti sú za NATom* - spojenie je nadviazané za pomoci supernód klientov a multimediálny obsah je prenášaný jednou dohodnutou supernódou.

Tento spôsob komunikácie cez jednu supernódu nielenže umožňuje klientom za NATom, prípadne firewallom nadviazať spojenie, ale navyše je možné na danú nódu pripojiť aj ďalších klientov do konferenčného hovoru. Nevýhodou v tomto prípade je zvýšená záťaž stanice užívateľa figurujúceho ako supernóda. Počas nadväzovania spojenia a aj prenosu hlasu/videa dochádza ku komunikácií klientov aj s inými supernódami než so sprostredkovateľskou.

3.3.6 Zabezpečenie

Komunikácia je v Skype protokole šifrovaná pomocou AES s 256 bitovým kľúčom, ktorý zabezpečuje súkromie, aj keď dáta prechádzajú niektorou supernódou. Pri logine klienta Skype vygeneruje dve 512 bitové prvočísla, z ktorých je vytvorený 1024 bitový RSA privátny a verejný kľúč. Klient si vygeneruje svoj symetrický kľúč K pre aktuálne spojenie. Výmena kľúčov prebehne nasledovne. Klient vytvorí MD5 hash z reťazca, pozostávajúceho z loginu hesla a slova skyper. Následne zašifruje svoj verejný modulus so získaným hašhom svojim kľúčom K. Potom zašifruje svoj kľúč K a odošle ho s predchádzajúcimi dátami na login server.

V prípade komunikácie sú klientom odoslané ich verejné kľúče podpísane Skype autoritou pre ich vzájomnú autentifikáciu. Následne sa klienti dohodnú na vzájomnom kľúči pre nasledovnú komunikáciu.



Obrázek 3.2: Zabezpečenie skype protokolu [6]

3.3.7 Kodek

Skype vystriedal niekoľko metód kodóvania audio a video obsahu. Pre audio to bol napríklad G.729 alebo SVOPC, aktuálne sa používa nimi navrhnutý SILK. Video obsah je aktuálne kodóvaný do VP8 formátu (predtým VP7).

Kapitola 4

Metódy detekcie účastníkov Skype siete

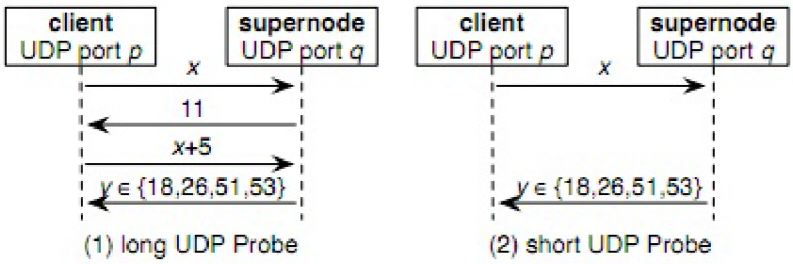
Pomocou metód reverzného inžinierstva bolo získaných dostatok informácii o fungovaní Skype protokolu. Detekcia Skype dát však vzhľadom na architektúru siete (decentralizovaná peer-to-peer) nie je triviálna, keďže klienti komunikujú medzi sebou navzájom bez použitia nejakého všeobecne známeho servra. Jednou z možností detekcie by teda bolo analyzovať obsah jednotlivých paketov. Toto riešenie je reálne aplikovateľné iba na veľmi malej sieti, pretože uloženie celého dátoveho toku siete by prakticky vyčerpalo cenovo dostupné dátové úložište v priebehu niekoľkých minút a "realtime" spracovanie prechádzajúcich dát prevyšuje aktuálne dostupný výpočetný výkon. Ostala nám teda možnosť analyzovať NetFlow dáta, ktorých uskladnenie a prípadné "realtime" spracovanie je realizovatelné.

4.1 Metodológia

Sledovaním NetFlow dát bolo objavených niekoľko podobností vo veľkosti prenášaných signalizačných paketov, z ktorých je možné detekovať jednotlivých učastníkov Skype komunikácie. Zaujímajú nás hlavne dva druhy komunikácie, čo je UDP probe a TCP handshake.

4.1.1 UDP próba

Jedná sa o UDP komunikáciu medzi klientom a supernódou (Obrázok 4.1) po spustení aplikácie pri prihlasovaní do siete.



Obrázek 4.1: Varianty UDP próby [10]

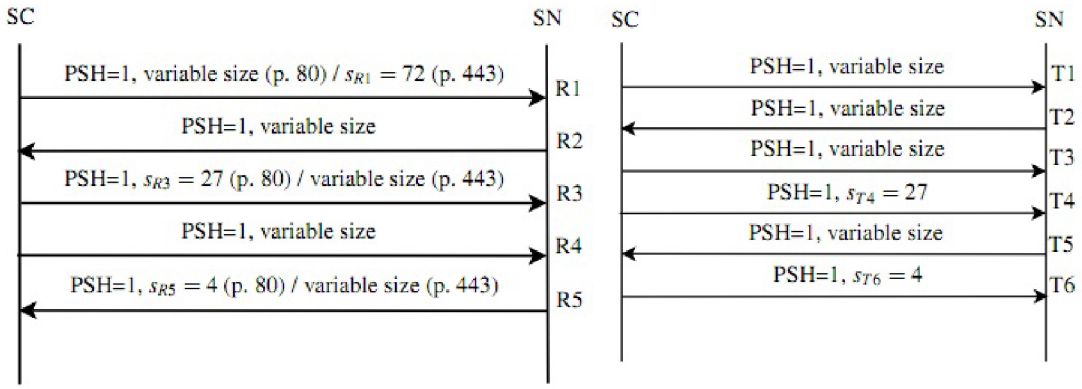
Rozlišujeme dve varianty UDP probe, a to krátku a dlhú. Pri prihlasovaní klienta je

kontaktovaných niekoľko supernód zo zoznamu klienta jednou z variánt (nie je nám zrejmé, podľa čoho sa varianta vyberá). Pokiaľ je posledná správa od supernódy o veľkosti 18 bajtov znamená to, že nás daná nóda akceptuje a môžeme pristúpiť k nadviazaniu TCP spojenia. V prípade, že je veľkost 26, 51 alebo 53 bajtov, chápeme túto odpoveď ako odmietnutie a daná nóda už nebude ďalej kontaktovaná. Krátka UDP próba je odosielaná aj po nadviazaní TCP spojenia, aby sa klient uistil, že má stále dostupné supernódy v prípade zlyhania aktuálnej. Vzhľadom na počet a charakter správ je táto komunikácia ideálna na detekciu klientov a supernód.

4.1.2 TCP handshake

Po výbere supernódy môžeme pristúpiť k nadviazaniu TCP spojenia na celú reláciu alebo do zlyhania supernódy. V prípade neúspechu komunikácie na porte, ktorý máme uložený v lokálnom zázname, skúša klient komunikovať na porte 80 prípadne 443. Pokiaľ sa nepodarí nadviazať spojenie ani na týchto portoch, je zahájená komunikácia s inou supernódou, ktorá nám kladne reagovalana UDP próbu. Pri tejto komunikácií už nemáme tak presnú informáciu od veľkosti paketov ako v predchádzajúcom prípade.

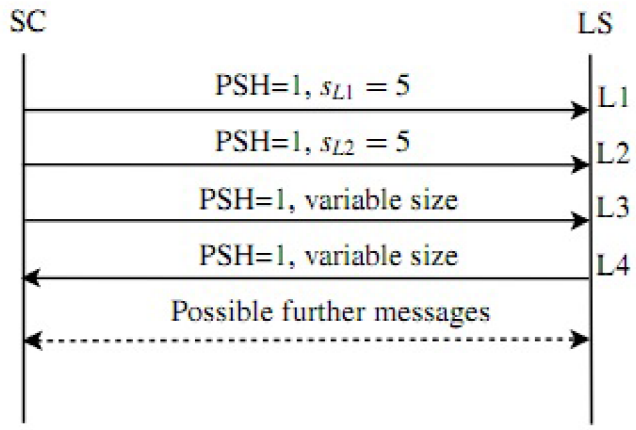
Priebeh komunikácie vyzerá nasledovne (Obrázok 4.2). Vidíme, že pri komunikácií (SC - Skype client, SN - Skype supernode) na rozdielnych portoch sú isté odlišnosti medzi veľkosťou a počtom správ. Avšak všetky pakety majú príznak TCP PSH (PSH=1) a posledné tri správy majú zhodnú veľkosť ($s_r = 27; 4$) nezávisle na použitom porte.



Obrázek 4.2: TCP handshake [3]

4.1.3 TCP autentizácia

Po nadviazaní TCP spojenia z supernódou je zahájená autentizácia (Obrázok 4.3) klienta (SC) s login servrom (SL), čo je jeden z mála centralizovaných prvkov siete, a preto je táto komunikácia tiež použiteľná k detekcií klientov. Počet správ nie je pri tejto komunikacií konštantný, vieme iba, že správy majú príznak TCP PSH a veľkosť prvých dvoch správ od klienta je 5. Avšak vzhľadom na to, že Skype umožnuje ukladať užívateľom niektoré autentizačné údaje, nie je táto komunikácia pri každom štarte aplikácie nutná.



Obrázek 4.3: TCP autentizácia [3]

4.1.4 Detekčný algoritmus

Vzhľadom na získané informácie bol zostavený detekčný algoritmus identifikujúci klientov a supernódy [10]:

```
acklist ← 0, supernodes ← 0, clients ← 0
for all f ∈ flowstream do
  if protocol(f) = UDP and portdest ≥ 1024 then
    if <packets(f), bytes(f)> ∈ {<1,46>,<2,85>} then
      sn ← addresssource(f), q ← portsource(f), cl ← addressdest(f)
      acklist ← acklist ∪ <sn,{q,80,443},cl>
    end if
  else
    if  protocol(f) = TCP and packets(f) ≥ 3 and PSH ∈ flags(f) then
      sn ← addressdest(f), q ← portdest(f), cl ← addressource(f)
      if  <sn,sp,cl> ∈ acklist then
        supernodes ← supernodes ∪ sn
        clients ← clients ∪ cl
      end if
    end if
  end if
end for
```

Algoritmus 4.1: skype nodes detection

Algoritmus funguje nasledovne:

- vyberie záznam f z NetFlow dát (flowstream).
- Ak je to UDP komunikácia, snaží sa detekovať short a long UDP probe (hodnoty 46,85 boli získané na základe veľkosti IP a UDP hlavičiek a samotnej próby).
  - V prípade úspechu sú IP adresy (sn - supernóda, cl - klient) a port (q, 80, 443) uložene do zoznamu potenciálnych klientov Skype siete (acklist).

- Ak je to TCP komunikácia, snaží sa detekovať TCP handshake na základe počtu práv v toku a prítomného PSH príznaku.
  - v prípade nájdenia odpovedajúceho toku je porovnaná adresa zdroja, cieľa a port so záznamami z acklistu a pri zhode sme detekovali klienta a supernódu

Analýzou algoritmu vidíme, že sme pri UDP komunikácií vylúčili porty nižšie ako 1024, aby nedošlo k nesprávnemu ohodnoteniu klienta ako užívateľa Skype siete, keďže na nižších portoch komunikuje napríklad DNS protokol, ktorého dotaz by mohol byť zle vyhodnotený. Následne nesmieme údaje z acklistu nechať starnúť a údaj, ktorý je v zozname dlhšie ako 3 sekundy odstrániť. Algoritmus má však aj svoje obmedzenia. Flow data nesmú byť vzorkované ani pozostávať zo vzorkovaných dát, keďže k úspešnej detekcií potrebujeme presné poradie, počet a veľkosť. Ďalším obmedzením je nemožnosť detekovať klientov, komunikujúcich zo sietí so zakázanou UDP komunikáciou, keďže nemáme presné informácie o priebehu spojenia so supernódou v tomto type siete.

Kapitola 5

Realizácia riešenia

Táto kapitola je venovaná popisu jednotlivých nástrojov, ktoré som pri práci používal alebo sú nevyhnutné k behu detekčného skriptu. Taktiež budú podrobne popísané jednotlivé kľúčové časti implementovaného skriptu.

5.1 Nástroje

Pri práci som využíval prevažne nástroje `nfdump`, k extrakcií dát z požadovaných NetFlow záznamov a nmap k overovaniu nájdených supernód.

5.1.1 Nfdump [1]

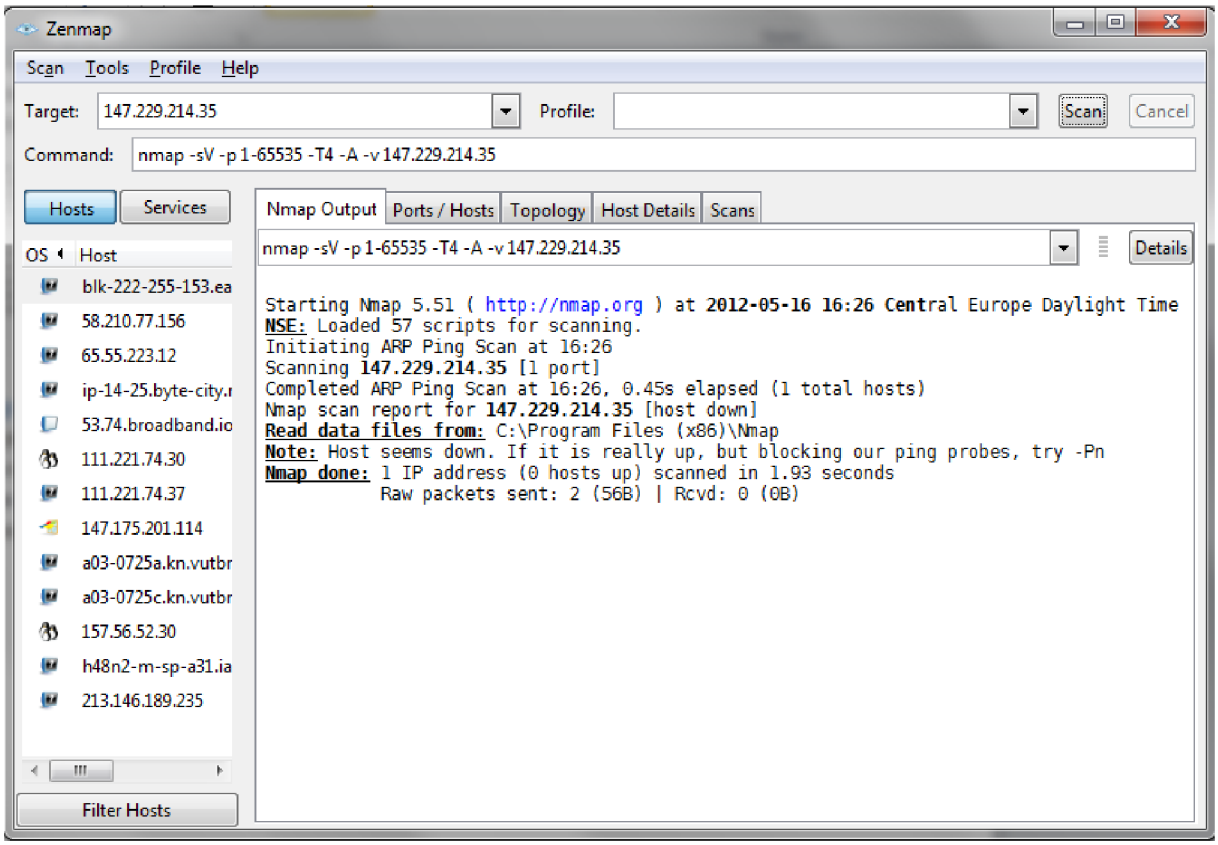
Nfdump je nástroj, ktorý umožňuje predovšetkým extrahovať NetFlow dáta v čitateľnej podobe z nfcapd súborov. Ďalšou podstatnou funkcionalitou je možnosť vytvárať rôzne štatistiky a filtrovať záznamy na základe širokej palety parametrov. Medzi jeho zásadné výhody patrí hlavne rýchlosť, ktorú zaisťuje implementácia v jazyku C. Ovládanie je štandardné, pomocou príkazovej riadky a množstva prepínačov. Aktuálne podporuje NetFlow záznamy verzie 5, 7 a aj 9.

```
Date flow start         Duration Proto    Src IP Addr:Port         Dst IP Addr:Port   Flags Tos  Packets    Bytes  Flows
2005-08-30 06:50:11.218  700.352 TCP     126.52.54.27:47303 ->    42.90.25.218:435   ......   0    1.4 M    2.0 G      1
2005-08-30 06:47:06.504  904.128 TCP   198.100.18.123:54945 ->    126.52.57.13:119   ......   0   567732  795.1 M      1
2005-08-30 06:47:06.310  904.384 TCP     126.52.57.13:45633 ->  91.127.227.206:119   ......   0   321148  456.5 M      1
2005-08-30 06:47:14.315  904.448 TCP     126.52.57.13:45598 ->  91.127.227.206:119   ......   0   320710  455.9 M      1
2005-08-30 06:47:14.316  904.448 TCP     126.52.57.13:45629 ->  91.127.227.206:119   ......   0   317764  451.5 M      1
2005-08-30 06:47:14.315  904.448 TCP     126.52.57.13:45634 ->  91.127.227.206:119   ......   0   317611  451.2 M      1
2005-08-30 06:47:06.313  904.384 TCP     126.52.57.13:45675 ->  91.127.227.206:119   ......   0   317319  451.0 M      1
2005-08-30 06:47:06.313  904.384 TCP     126.52.57.13:45619 ->  91.127.227.206:119   ......   0   314199  446.5 M      1
2005-08-30 06:47:06.321  790.976 TCP     126.52.54.35:59898 ->   132.94.115.59:2466  ......   0   254717  362.4 M      1
2005-08-30 06:47:14.316  904.384 TCP     126.52.54.35:59773 ->  55.107.224.187:11709 ......   0   272710  348.5 M      1
```

Obrázek 5.1: vzorový výstup nfdump

5.1.2 Nmap [2]

Nmap je bezpečnostný port scanner . Za jeho pomoci sme schopní odhaliť topológiu siete, stav portov na skenovaných zariadenia, prípadne používaný operačný systém. Je šírený pod licenciou GNU GPL na takmer všetky operačné systémy. Okrem tradičného ovládania z príkazovej riadky, bolo vytvorených aj niekoľko verzií s GUI (napríklad Zenmap obrázok 5.2). V tejto práci je využitá hlavne jeho vlastnosť identifikovať služby, bežiace na daných portoch, za použitia identifikačných skriptov napísaných prevažne v jazyku `LUA`.



Obrázek 5.2: Zenmap

5.1.3 Wireshark [13]

Wireshark je pod GNU GPL licenciou šíriteľný open-source paket analyzér. Jeho implementácia v jazyku C mu umožnuje beh pod väčšinou operačných systémov. Na rozdiel od linuxového `tcpdump` nástroja má grafický frontend (obrázok 5.3) a viac možností filtrovania. Na zachytávanie paketov používa predovšetkým knižnicu libpcap.

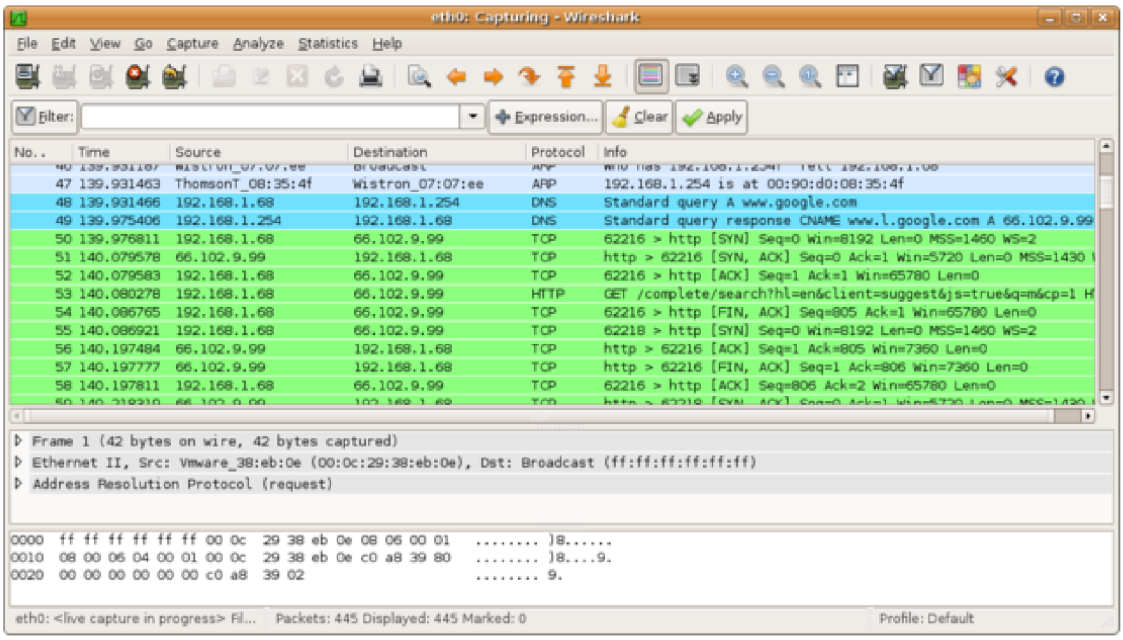
5.2 Implementácia

Vzhľadom na charakter spracovávaných dát som sa rozhodol implementovať supernode detektor v sriptovacom jazyku python.

Prvým krokom je získanie vstupu (jednotlivé NetFlow záznamy) v nami požadovanom tvare. To zabezpečíme použitím nasledujúcich parametrov nmapu:

```
-m -o "fmt: %ts %pr %sap %dap %flg %pkt %byt"
```

Prepínač `-m` nám zoradí všetky záznamy podľa času začiatku toku. Vzhľadom na to, že záznamy sú v súboroch iba relatívne zoradené, vznika pri odstraňovaní starých potencionálnych supernód z acklistu problém, pretože kvôli novo-načítanému toku so začiatkom, ktorého rozdiel s nastaršími položkami v ackliste je väčší ako nastavená delta, môže dôjsť k strate supernódy. Prepínač `-o` nám podľa zadaného reťazca vracia vo výstupe programu



Obrázek 5.3: Wireshark

`nfdump` len požadované polia, čo je v našom prípade štart toku, cieľová a zdrojová IP adresa a port, počet paketov v toku a ich veľkosť. `Nfdump` výstup v požadovanom formáte spracujeme za použitia príslušného regulárneho výrazu a prechádzame k druhému kroku, čo je samotná detekcia.

Najskôr zisťujeme, či sa jedná o UDP alebo TCP protokol a podľa toho detekujeme buď `UDP probe` alebo `TCP handshake`. V prípade, že sme našli pravdepodobne UDP probe komunikaciu, ukladáme do zoznamu potencionálnych supernód (acklist) informáciu o čase, `s_ip` a `s_port`, čo je zdrojová IP adresa a port daného toku (pravdepodobne naša supernóda) a `d_ip` čiže cieľová IP adresa toku (náš Skype klient). Porty nižšie ako 1024 nás nemusia zaujímať, pretože aplikácia ich bežne nepoužíva. Ak sa jedná o TCP komunikáciu (handshake iniciovaný od klienta), požadujeme aspoň 3 pakety a prítomný PSH príznak, ktorý zabezpečuje, že pakety budu prijímané a odosielané ihneď. Tu sa dostávame k časovo najnáročnejšej časti algoritmu, a to k prehľadávaniu acklistu. Snažíme sa nájsť zhodu s cieľovou adresou a portom (supernóda) a zdrojovou adresou (klient) nášho toku s niektorým zo záznamov v ackliste a potvrdiť tak, že sa skutočne jedná o supernódu. Z acklistu treba pravidelne odstraňovať staré záznamy, aby nedošlo k chybnej detekcií, ale hlavne, aby sme zbytočne nevykonávali operácie, ktoré zvyšujú časovú náročnosť. Časový rozdiel začiatku aktuálneho toku a záznamu v ackliste potrebný pre odstránenie záznamu bude podrobnejšie popísaný v nasledujúcej kapitole. Pri detekcií nesmieme zabudnúť na fakt, že klient sa môže snažiť o TCP handshake so supernódou aj na portoch 80 a 443, okrem štandardného, náhodne zvoleného portu pri inštalácií aplikácie.

Požitie detekčného skriptu a jeho vstupy a formát výstupu sú bližšie popísane v prílohe.

```
if(protocol == "UDP" and d_port >= 1024):
        if((packets == 1 and bytes == 46) or
           (packets == 2 and bytes == 85)):
                acklist.append((s_ip,s_port,d_ip,date))
```

Algoritmus 5.1: UDP probe detection

```
if(protocol == "TCP" and packets >= 3 and flag == 'P'):
         for ack in acklist:
                delta = date - ack[3]

        if(delta.seconds > 3):
                dumpacks.append(ack)
                continue

        if(ack[0] == d_ip and
          (ack[1] == d_port or ack[1] == 80 or ack[1] == 443) and
           ack[2] == s_ip):
                supernodes.append((ack[0]))
                clients.append((ack[2]))

for old in dumpacks:
        acklist.remove(old)
dumpacks = []
```

Algoritmus 5.2: TCP handshake detection

Kapitola 6

Testovanie

Pre testovacie účely som mal k dispozícií aktuálne NetFlow záznamy z internátnej siete VUT, ktorú obýva niekoľko tisíc študentov. Záznamy sú zachytávané v 5 minútových intervaloch, čiže k testovaniu supernód dochádzalo s maximálne 5 minútovým oneskorením.

6.1 Supernódy

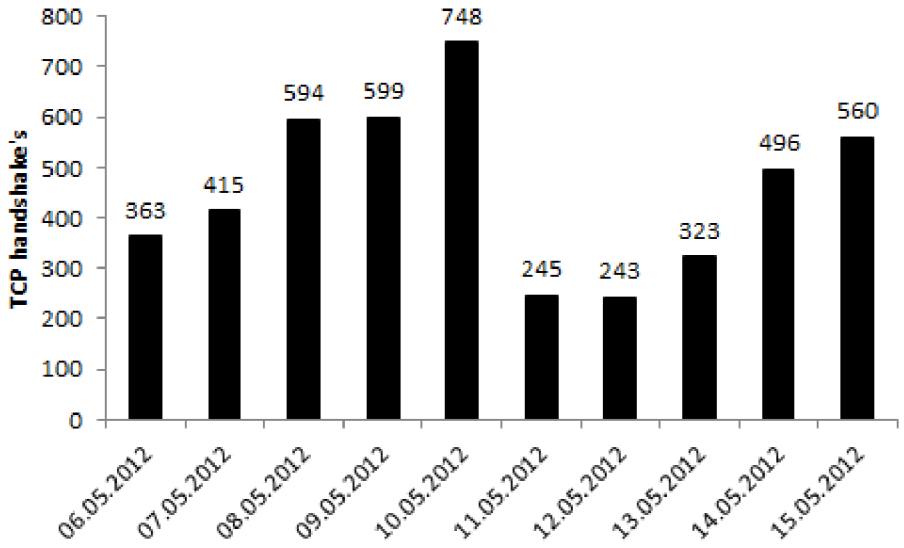
Detekčný skript hľadal aktívne supernódy v období od 6. mája do 15. mája. Skript bol aplikovaný každých 5 minút na novo uložené dáta z NetFlow kolektora. Pri objavení novej supernódy bol pre potvrdenie správnosti detekcie použitý port skener `nmap`. Využitý bol predovšetkým jeho prepínač `-sV`, ktorý sa snaží detekovať službu bežiacu na danom porte alebo portoch. Pre overenie správnosti stačilo nechať nmap otestovať len skript detekujúci skype (priloha D), avšak potom by sme v prípade neúspechu prišli o možnosť zistiť, ktorá služba generuje podobný data flow.

Za toto 10 dňové obdobie sme detekovali 4586 nadviazaných spojení (TCP handshake), z čoho bolo 1576 unikátnych supernód. Dôvod tohto nízkeho počtu supernód bude bližšie popísaný v nasledujúcej kapitole. Čo sa týka úspešnosti detekcie, tak sa blížime 100%, keďže iba jeden výsledok z 4586 nebola Skype supernóda, ale aplikácia `Shareaza`, čo je p2p klient siete `gnutella` prípadne `eDonkey`. Počet spojení v priebehu týchto 10 dní je možné vidieť na grafe 6.1. Na grafe je viditeľný pokles výskytu spojení, ktorý je spôsobený pravidelným odchodom vačšieho počtu študentov v priebehu piatka až nedele.

Graf na obrázku 6.2 zachytáva štatistiku celkového počtu klientov pripojených na supernódu v pozorovanom 10-dňovom intervale. Zaujímavý je fakt, že približne tretina supernód mala za obdobie 10-dní iba jedného pripojeného klienta.

Čo sa týka životnosti supernód, tak je treba brať výsledky s istou rezervou, pretože nevieme presne odhadnúť, od kedy bola supernóda skutočne aktívna. Vzhľadom na to, že v priebehu 10-dní boli supernódy detekované postupne, nie je možne sa snažiť porovnávať ich dobu aktivity v hodinách, preto som ako je vidno na obrázku 6.3 vyjadril ich aktivitu v percentách od momentu detekcie, až do konca testovania. Supernódy boli testované v 10-minútových intervaloch na otvorenosť portu, na ktorom boli prvý krát spolu zdetekované. Vzhľadom na to, že užívatelia na tejto sieti majú počas celej doby ubytovania priradenú jednu, nemeniacu sa verejnú IP adresu a port, ktorý supernóda používa, sa bežne nemení, je aj po dlhšej dobe neaktivinosti možné opätovné testovanie na živosť. Zistené hodnoty sú avšak trochu prekvapivé, pretože takmer 73% supernód bolo aktívnych počas celej doby tj. od detekcie až do konca testu a ďalších 10% bolo aktívnych viac ako 90% tohto času.

Vzhľadom na charakter siete VUT, kde sú ideálne podmienky pre vznik supernód, bola väčšina supernód z adresového priestoru VUT a väčšina klientov z vonka. Našlo sa však aj pár prípadov, kde tomu bolo opačne.



Obrázek 6.1: Počet pripojení klientov v jednotliné dni

6.2 Časová náročnosť

Testovanie prebiehalo na zostave obsahujúcej procesor Intel Xeon X5650 (2,66 GHz) a 64Gb pamäte. Časovú náročnosť ovplyvňovali predovšetkým nasledujúce parametre:

- doba, kým `nfdump` zoradil nfcapd súbor,
- doba zotrvania záznamov v ackliste,
- a predovšetkým počet supernód v danom súbore.

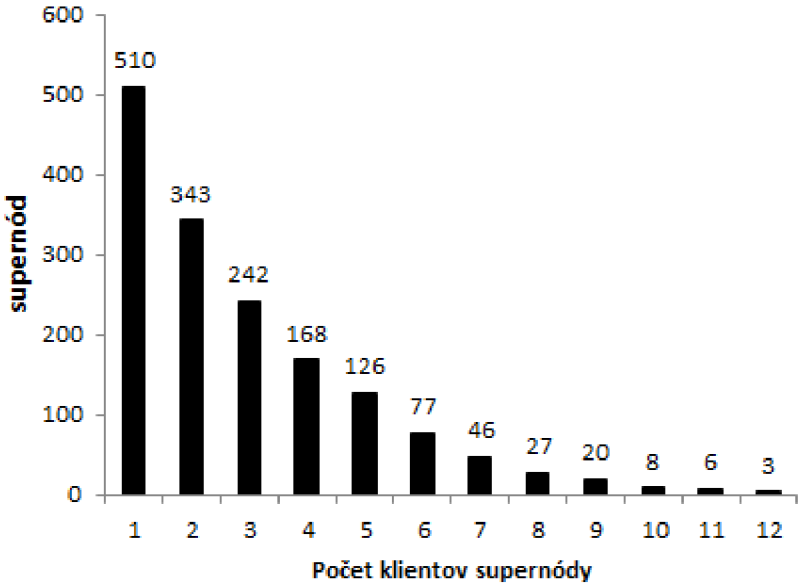
Doba, ktorú potrebuje `nfdump` na zoradenie záznamov, je pri porovnaní s ostatnými položkami zanedbateľná. Na zoradenie NetFlow záznamov z jedného pracovného dňa je potrebná doba približne 3 a pol minúty.

doba v ackliste	doba detekcie	počet supernód
1s	6m42s	52 supernód
2s	7m30s	52 supernód
3s	8m11s	54 supernód

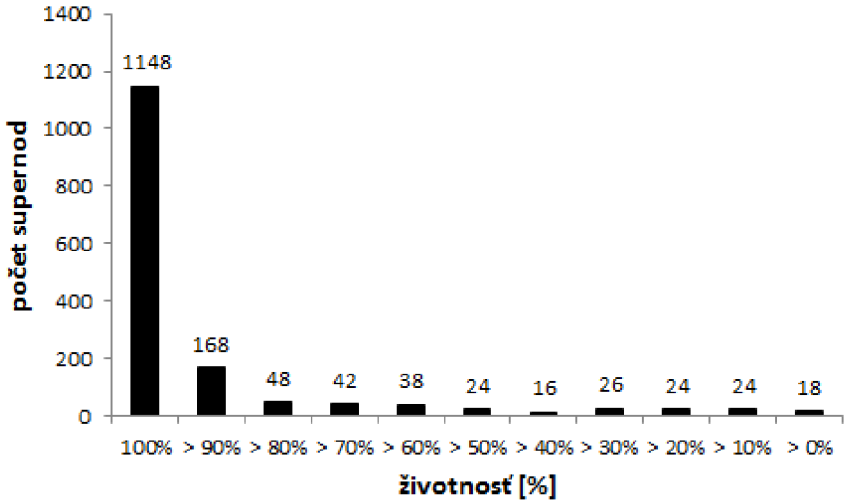
Tabulka 6.1: Vplyv doby zotrvania záznamov v ackliste na dĺžku detekcie a počet zdetekovaných supernód

Ako zobrazuje tabuľka 6.1, tak doba zotrvania záznamov v ackliste výrazne ovplyvňuje dĺžku detekcie. Uvedené časové údaje zobrazujú dobu trvania detekcie v hodinovom zázname uprostred pracovného dňa.

Tabuľka 6.1 ďalej ukazuje, že doba zotrvania záznamov v ackliste ovplyvňuje nie len časovú náročnosť ale aj počet zdetekovaných supernód. Preto treba pri nastavovaní tohto parametra zvážiť, či za cenu skracovania času detekcie neprichádzame o dáta.



Obrázek 6.2: Celkový počet klientov obslúžených supernódou za 10 dní



Obrázek 6.3: Približná životnosť supernód počas testovania

Kapitola 7

Zmena architektúry Skype siete

Pozorovaním klesajúceho počtu supernód za obdobie posledných pár mesiacov sa začalo uvažovať, či Microsoft ako nový vlastník Skype, nezačal vykonávať zásahy do jeho originálneho princípu fungovania. Kedže Microsoft neavizoval žiadne zmeny s predstihom, muselo sa opäť sledovaním nového správania aplikácie vyvodiť, o aké konkrétne úpravy sa jedná.

Zásah do siete Skype môžeme označiť ako dosť podstatný, pretože Microsoft začal nasadzovať vlastné dedikované supernódy (v texte ich budem ďalej označovať ako MSnódy), čím sieť stráca svoju tradičnú decentralizovanú p2p architektúru. Tieto zistenia sa začali potvrdzovať koncom apríla tohto roku a predpokladá sa, že k nasadeniu MSnód došlo niekedy v období medzi febuárom a marcom 2012. Po otázkach, či sa skutočne jedná o nasadenie dedikovaných supernód vydal Microsoft v máji tohto roku nasledujúce vyhlásenie[7]:

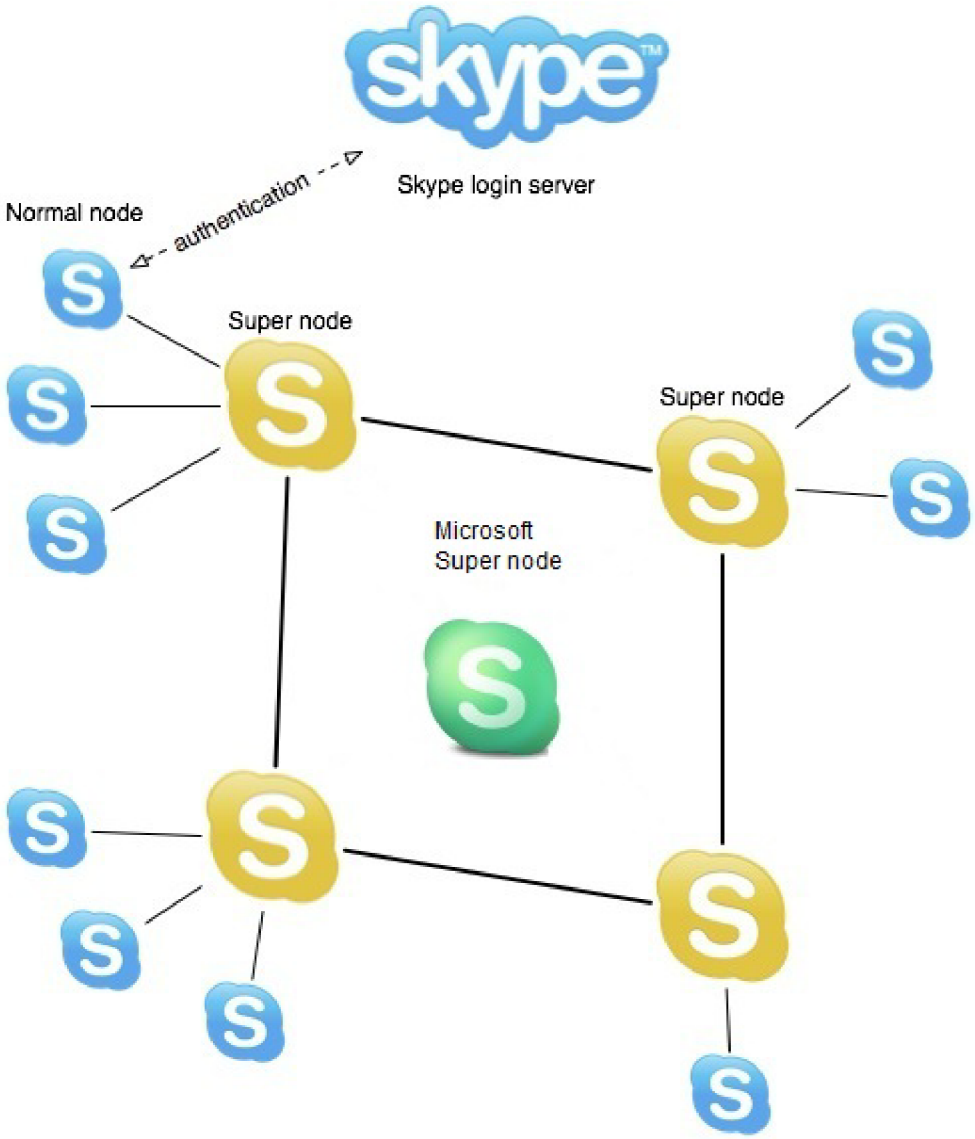
> As part of our ongoing commitment to continually improve the Skype user experience, we developed supernodes which can be located on dedicated servers within secure datacentres. This has not changed the underlying nature of Skype's peer-to-peer (P2P) architecture, in which supernodes simply allow users to find one another (calls do not pass through supernodes). We believe this approach has immediate performance, scalability and availability benefits for the hundreds of millions of users that make up the Skype community.

Podľa vyššie uvedeného textu sa potvrdilo predpokladané a dedikované supernódy by skutočne mali prevziať úlohu ich predchodcov. Tento fakt by mohol potešiť užívateľov, pretože vo výsledku by to znamenalo, že už nebude nutné riešiť problém s nadmerným vyťažením internetového pripojenia, či ubúdajúceho výkonu daného zariadenia a ako sa ďalej píše v správe, tak aj kvalita hovoru by mala stúpnuť. Problém s hľadaním novej supernódy, by v podstate prestal existovať, pretože je málo pravdepodobné, že by došlo k odstaveniu MSnódy. Správa ďalej uvádza informáciu o ponechaní aktuálnej architektúry, avšak terajšie supernódy by mali vykonávať už len funkciu vyhľadávania užívateľov navzájom a funkciu tunelovania komunikácie preberú kompletne MSnódy.

7.1 MSnóda

Bohužiaľ sa zatiaľ neobjavila nikde informácia bližšie popisujúca novo-nasadené MSnódy, takže sa pokúsim v tejto sekcií zhrnúť doteraz zistené údaje.

Msnódy sú stanice pracujúce pod bližšie nešpecifikovanom Linuxovom operačnom systéme. Pri návrhu bol kladený dôraz predovšetkým na bezpečnosť a výkonnosť. Okrem Skypu



Obrázek 7.1: Nová topológia Skype siete

by na MSnódach nemala bežať žiadna iná zvonka prístupná, a tým pádom aj napadnuteľná služba. Ďalším zabezpečením je použitie `grsecurity`, čím by sa malo predchádzať aj "zero day's" útokom.

`Grsecurity` je sada záplat pre Linuxové systémy s zameraním predovšetkým na bezpečnosť. Prichádza s RBAC (Role-Based Access Control) modelom, ktorý významne rozširuje pôvodný MAC systém ACL práv pre prednotlivé procesy. Asi najzaujímavejšie rozšírenie je záplata `PaX`, ktorá pridáva na x86 architektúré podporu nespustiteľných stránok a umožnuje tak predchádzať hrozbám zneužívajúcim napríklad "buffer overflow".

Čo sa výkonnosti týka, MSnóda aktuálne spravuje približne 4000 klientov a údajne má byť schopná spravovať až 100 000 klientov. Pre porovnanie, originálna supernóda mala na starosti približne 800 užívateľov. Počet Msnód je odhadovaný približne na 10 000.

MSnóda má tak isto ako stará supernóda spustenú Skype službu na porte 80 a 443, avšak na rozdiel od jej predchodcu nemá zvolený len jeden náhodný port v rozsahu nad

1024, ale zvyčajne 47 portov v intervale 40 001 - 40 047. Pomerne často sa objavuje aj port 33033, ktorý bol používaný u originálnych Skype bootstrap supernódach.

Detekcia operačného systému MSnód nástrojom nmap nám vracala rozdielne výsledky. Vo väčšine prípadov to bol Linux verzie 2.6.17 - 2.6.35 (detekcia nie je úplne presná) a Linux 2.6.28 upravený pre Nokie N900. Predpokladám, že tá druhá varianta je len nepresný odhad, ale je tu aj možnosť, že Microsoft ako partner firmy Nokia prebral nejake ich Linuxové riešenia využívané v ich mobilných telefónoch. Konkrétne sa jedná o systém Maemo, ktorý beží na ARM platforme.

7.2 Detekcia MSnód

Bohužiaľ zavedením Msnód sa očividne zmenili aj incializačné postupy, ktoré sme boli schopní detekovať. Detekčný algoritmus nie je schopný zachytit komunikáciu klienta s MSnódou, keďže dvojica procedúr UDP probe a TCP handshake sa tu už nevyskytuje alebo je upravená. Vzhľadom na to, že IP adresy MSnód sa pravdepodobne nebudú dynamicky meniť, bude po vytvorení kompletného zoznamu možné jednoducho detekovať Skype komunikáciu, keďže každý flow z alebo na dané adresy bude s najväčšou pravdepodobnosťou Skype komunikácia.

Za použitia nástroja wireshark som sa snažil sledovať zmeny v tejto komunikácií. Po spustení sa zachovala operácia zisťovania najnovšej verzie aplikácie a očividne aj UDP probe. Zaujímavé je, že v dobe testovania sa najskôr odoslala jedna krátka UDP probe na adresu zrejme zo zoznamu v registroch klienta, pretože sa tam občas vyskytla aj stará supernóda a nie MSnóda. Veľkosť správy smerom od klienta sa štandardne mení ako pri originálnej UDP probe, avšak odpoveď od supernódy mala pri testovaní vždy veľkosť 26 bajtov. To by pri starej variante znamenalo odmietnutie avšak v tomto prípade som si nie som istý, keďže som dostával vždy len odpoveď tejto veľkosti. Po prvej UDP probe sa kontaktuje zase UDP komunikáciou väčšie množstvo MSnód a supernód cca. 15 a aj po odpovediach kontaktovanie nových MS/supernód pokračovalo. Veľkosť odpovedí sa bohužiaľ na rozdiel od starej varianty UDP probe nepohybuje v množine 4 veľkostí, ale výskyt niektorých hodnôt je pomerne častý (17, 20, 21, 24, 98).

TCP komunikácia sa v pozorovaných dátach tiež vyskytuje a počtom správ a PSH príznakom sa dosť podobá TCP handshake procedúre. Zarážajúce je, že oproti starej variante ju nezačína klient, ale druhá strana, s ktorou sa do toho momentu nekomunikovalo. Pri skenovaní týchto adries nám väčšina neodpovedala a tak predpokladám, že sa nejednalo o MSnódy.

Bohužiaľ vzhľadom na neskoré objavenie MSnód som nemal viac času na lepšie skúmanie nových operácií.

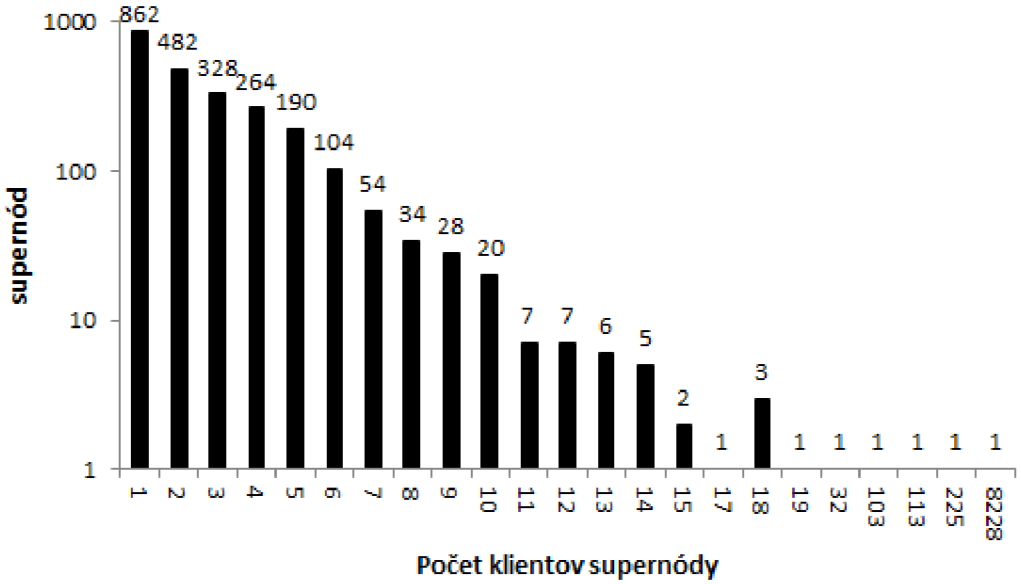
7.3 Testovanie

Vzhľadom na novo-zistené informácie a objasnenenie nízkeho počtu zdetekovaných supernód v predchádzajúcej kapitole, som sa rozhodol zopakovať testovanie aj na starších NetFlow záznamoch. Vzhľadom na to, že nemám možnosť novo-získané výsledky otestovať hneď po detekcií nástrojom nmap na správnosť, budeme predpokladať, že algoritmus funguje s presnosťou uvedenou pri testovaní v kapitole 6. Zameriam sa predovšetkým na počet detekovaných supernód a počty klientov, ktorí budu za testované obdobie spravované jednotlivými supernódami.

Graf na obrázku 7.4 znázorňuje počet detekovaných TCP handshake procedúr opäť v intervale 10 dní v mesiacoch január, marec a máj. Ak porovnáme počty za jednotlivé dni v marci a máji, môžeme hovoriť o dosť totožných výsledkoch. Treba však spomenúť, že od 10 hodiny doobeda 8.3.2012 do 9 hodiny doobeda 9.3.2012 neboli detekované žiadne supernódy. Keďže sa s näjvačšou pravdepodobnosťou nejednalo o problém s konektivitou na sieti CESNETu je pravdepodobné, že sa jednalo o odstávku Skype login servrov. Ak porovnáme počty pripojení na supernódy pred a po tomto období, vidíme značný nepomer, ktorý môže by pripísaný pomalým štartom siete, ale pravdepodobnejšie je, že sa v tejto dobe už nasadili MSnódy. Hodnoty sa od tohto dňa pohybujú v intervale, ktorý pretrváva do aktuálneho dňa. Jemné rozdiely, ktoré na grafe vidíme, sú spôsobené rozdielnym počtom prítomných študentov, kedže na dané dátumy pripadajú rozdielne dni v týždni.

Čísla za január nám však zreteľne indikujú, že tu ešte boli staré supernódy v plnej prevádzke a počet zdetekovaných pripojení je niekoľkonásobne väčší.

Počet jednotlivých pripojení na supernódy počas celého dňa zachytáva graf na obrázku 7.6. Je viditeľné, že sieť Skype je v špičke v intevale približne od 17 do 22 hodiny. Táto skutočnosť potvrdzuje fakt, že sa Skype snaží klientom poskytovať supernódy geograficky čo najbližšie. Čo sa týka počtu klientov obslúžených jednou supernódov, tak sú výsledky za



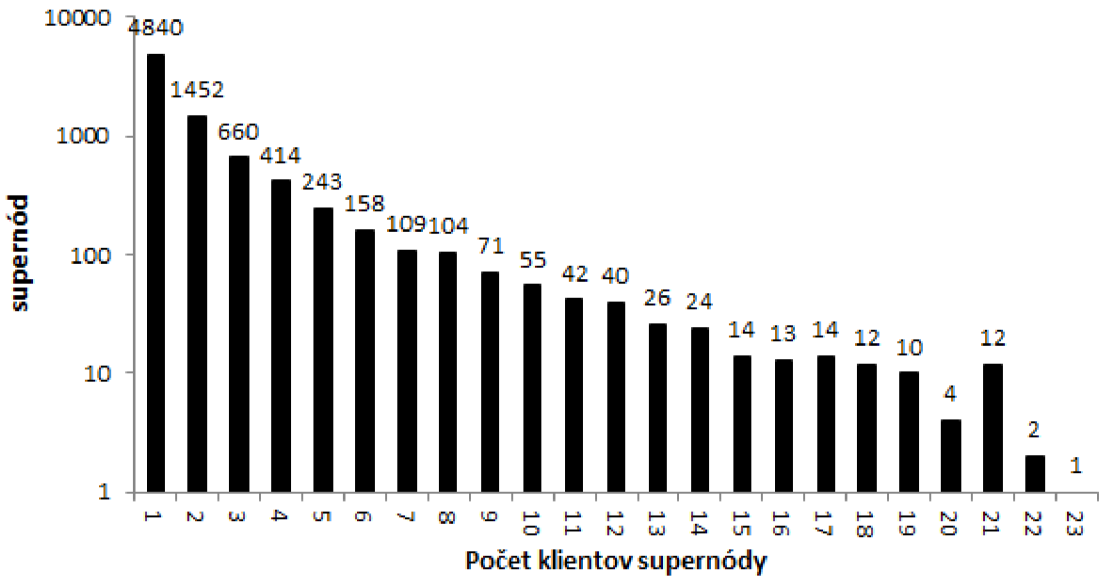
,

Obrázek 7.2: Celkový počet klientov obslúžených supernódou za 10 dní (marec 2012)

máj a marec (obrázok 7.2) pomerne podobné. Väčšina supernód nemala za 10 dní viac ako 5 klientov. Zaujímavý je jeden konkrétny užívateľ, ktorý musí mať veľmi vysoké percento doby online, keďže mal ako supernóda až 8228 pripojených klientov. Na grafe (obrázok 7.3) vidíme, že aj v januári mala väčšina z 8858 zachytených supernód len malé množstvo klientov. Graf nezachytáva niekoľko supernód, ktoré mali za toto obdobie nadštandardne viac klientov, avšak tých tam bolo len niekoľko. Rekordérom sa opäť stala vyššie spomínaná supernóda s 33617 klientami.

Vzhľadom na väčšie množstvo detekovaných dát som sa rozhodol zopakovať aj meranie dĺžky detekčného algoritmu a vplyvu doby zotrvania záznamov v ackliste na počet zdetekovaných supernód na hodinovom zázname z januára uprostred pracovného dňa. Tabuľka

,



Obrázek 7.3: Celkový počet klientov obslúžených supernódou za 10 dní (január 2012)

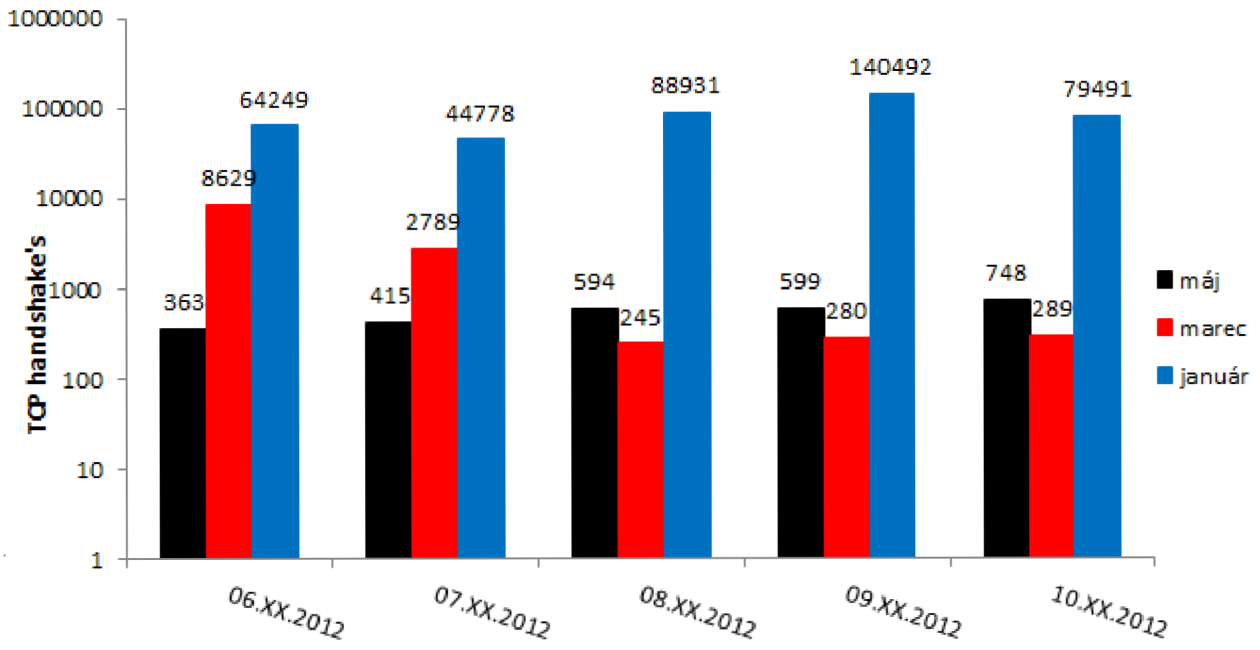
7.1 nám ukazuje, že niekoľkonásobne väčší počet Skype procedúr v NetFlow datach nám dosť podstatne predĺžil dobu detekcie.

Ako zobrazuje tabuľka 7.1, tak aj doba zotrvania záznamov v ackliste na tejto vzorke dát výraznejšie ovlyvňuje počet nájdených supernód.

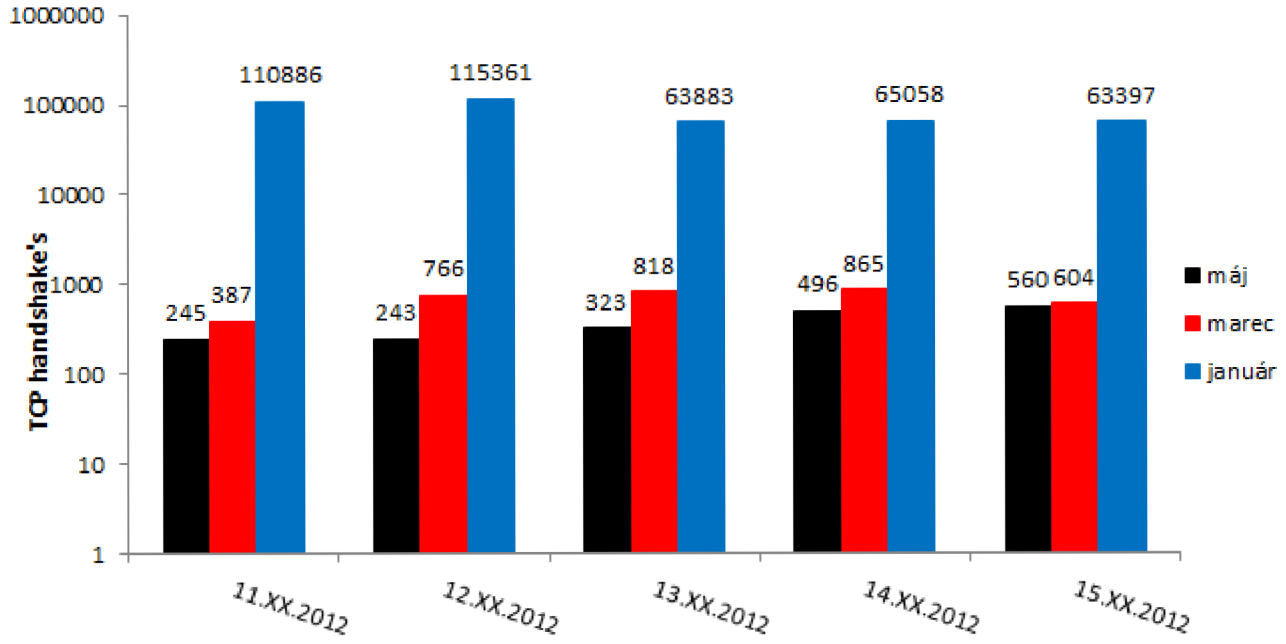
doba v ackliste	doba detekcie	počet supernód
1s	13m7s	2525 supernód
2s	17m24s	2723 supernód
3s	21m40s	2874 supernód
4s	25m51s	2972 supernód

Tabulka 7.1: Vplyv doby zotrvania záznamov v ackliste na dĺžku detekcie a počet zdetekovaných supernód

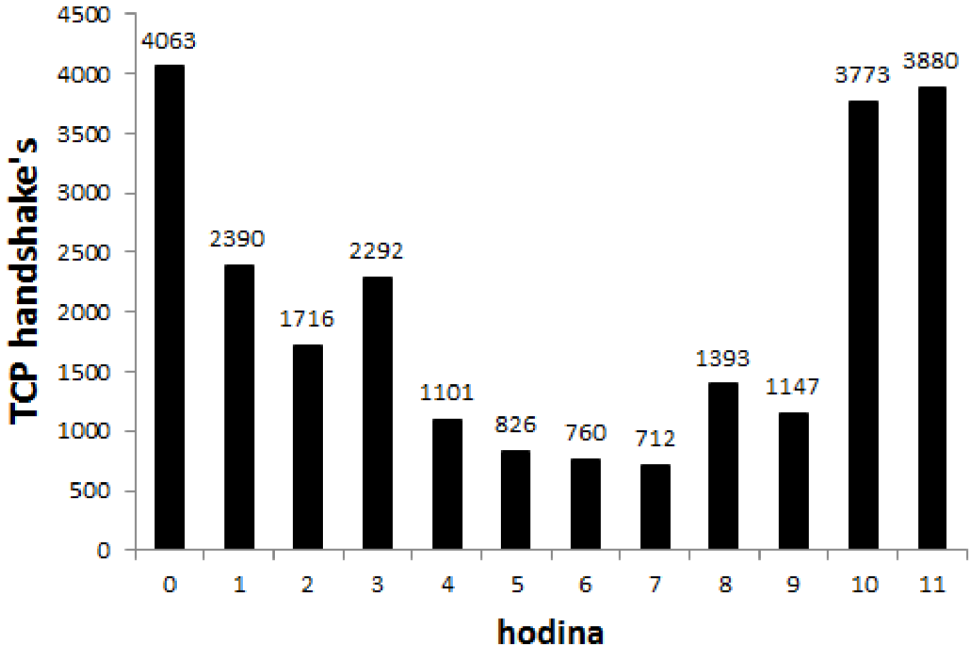
Zaujímavý je poznatok, že pri pokusoch o tunelovanie komunikácie cez MSnódy sa mi pomerne často stalo, že supernóda, ktorá mi bola priradená nebola MSnóda.



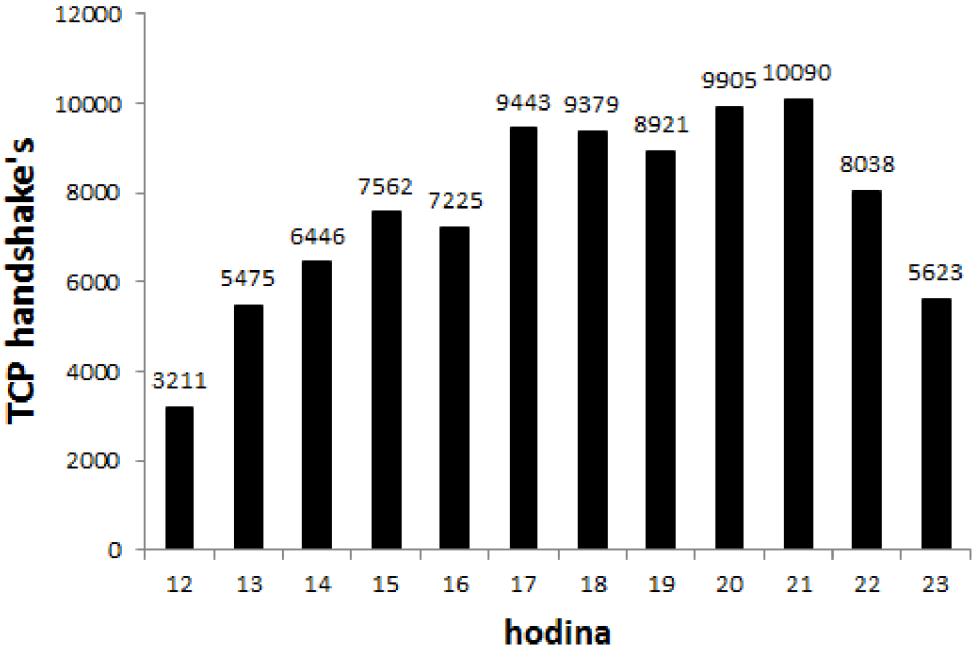
Obrázek 7.4: Štatistika detekovaných pripojení za január, marec a apríl



Obrázek 7.5: Štatistika detekovaných pripojení za január, marec a apríl (pokračovanie)



Obrázek 7.6: Počet detekovaných pripojení v priebehu jedného pracovného dňa (01.12.2012)



Obrázek 7.7: Počet detekovaných pripojení v priebehu jedného pracovného dňa (01.12.2012 pokračovanie)

Kapitola 8

Budúcnosť Skype siete

Originálna architektúra Skype siete umožňovala s rastom počtu užívateľov aj automatický rast počtu supernód. Nová architektúra je nútená pri neustále zvyšujúcom sa počte klientov pridávať nové MSnódy (aj keď podľa predpokladaného maximalneho počtu pripojení na jednu supernódu to ešte nie je v dohľadnej dobe problém).

Aj napriek veľkému dôrazu na bezpečnosť pri návrhu nových supernód, je zavedenie centrálneho prvku do decentralizovanej siete jemné riziko, keďže teraz je už možné identifikovať ciele pre možné distribuované útoky, čo pri starých supernódach nebolo príliš možné, keďže neustále dynamicky vznikali a zanikali. Na druhej strane je to plus, keďže naša komunikácia je spracovávaná zariadeniami, výhradne preto určenými, na rozdiel od staníc obyčajných užívateľov, kde bežne beží obrovské množstvo napadnuteľných služieb.

Čo sa kvality hovoru týka, je to určite veľké plus, pretože je nepravdepodobné, že nová supernóda počas hovoru zanikne a je treba nájsť a nadviazať komunikáciu s novou. Taktiež je príjemné, že už to nie je náš procesor a internetová linka, ktorá je vyťažovaná, aj ked množstvo dát prenesených supernódou nie je nejako extrémne veľké (cca. 1MB oboma smermi pri 3 minútovom hovore dvoch strán).

Na druhej strane je otázne, či sme neprišli o súkromie pri hovore, ktoré nám zabezpečovala nedeterministická voľba supernódy, čím sa hovor stalo v podstate nemožné odpočúvať (samozrejme by bolo treba najskôr prelomiť použité šifrovanie). Aktuálne vznikajú domnienky ohľadom funkcie nových supernód, pretože týmto sa stala sieť Skype ovládateľnejšou. Ak vieme, že hovor určite pôjde cez dané supernódy, je možné celú komunikáciu ukladať a samozrejme je možné aj upraviť spôsob šifrovania tak, aby bolo možné následné dešifrovanie. Toto sú však iba domnienky založené na doterajšej snahe niektorých vládnych organizácií dostať sa k možnosti odpočúvania tejto komunikácie.

Podľa vyhlásenia hovorcov Microsoftu by teda žiadna stará supernóda nemala už tunelovať komunikáciu klientov. Avšak počas môjho testovania sa mi podarilo komunikovať skrz staré supernódy. Je teda otázne, či to bolo spôsobené pretrvávajúcim záznamom tejto supernódy v mojich registroch alebo sa sieť Skype ešte úplne nepretransformovala na nový model alebo sa Microsoft rozhodol predsa len nechať isté množstvo starých supernód kompletne funkčných.

Aktuálne nám neostáva nič iné ako pozorne sledovať nové zmeny a bližšie analyzovať tie už doteraz zistené.

8.1 Možné rozšírenia

Práca poskytuje priestor pre mnohé rozšírenia, ako napríklad zmenu flow protokolu. Keďže NetFlow protokol je prevažne len na Cisco zriadeniach, je možné upraviť skript pre IPFIX či sFlow záznamy.

Nedávne zásahy do architektúry Skype siete poskytujú široký priestor pre ďalší výskum. Bolo by vhodné odsledovať, ako sa bude sieť zavedením MSnód postupne meniť v najbližšom období, zistiť, či skutočne staré supernódy prestali tunelovať hovory klientov a fungujú už len na vyhľadávanie klientov navzájom. Vzhľadom na nedostatok informácií o MSnódach a ich parametroch, je určite vhodné sledovať ich správanie, zistiť hardwarovú špecifikáciu, počty klientov na nich pripojených a aj skompletizovať ich zoznam.

Pretože došlo aj k zmene protokolu, hlavne čo sa týká komunikácie s MSnódami, je vhodné zamerať sa na podrobnejšie skúmanie obsahu, veľkosti a počtu paketov. Snažiť sa priradiť význam jednotlivým dotazom a odpovediam a zistiť, či je možné túto novú komunikáciu opätovne detekovať napríklad aj v NetFlow záznamoch.

Kapitola 9

Záver

Úvod práce bol venovaný predovšetkým teoretickej časti. Bol popísaný protokol NetFlow a jeho alternatívy a princíp fungovania Skype siete. Skype protokol som sa na základe doterajších informácií snažil popísať čo najlepšie a predstaviť tak operácie, ktoré sú vhodné k detekcií, a to predovšetkým UDP probe a TCP handshake.

V praktickej časti práce som sa venoval samotnej implementácií a testovaniu detekčného algoritmu. Ako implementačný jazyk som zvolil python. Samotná detekcia prebiehala na aktuálnych NetFlow dátach z pateřní siete VUT a dosiahnuté výsledky sú bližšie popísané v kapitole 6. Počas písania tejto práce došlo k dosť podstatným zásahom do architektúry siete Skype, hlavne čo sa týka supernód. Microsoft v rámci zlepšovania kvality služieb navrhol a nasadil vlastne supernódy (MSnódy), ktoré prebrali väčšinu povinnosíi ich predchodcov. Vzhľadom na to, že sa rozhodli ponechať originálne supernódy, tak detekčný algoritmus je stále použiteľný, avšak množstvo zachytených výsledkov je podstatne menšie. V kapitole 7 som sa snažil bližsie popísať vlastnosti nových supernód a s tým vyplývajúce zmeny v komunikačnom protokole. Bohužiaľ hlavne kvôli nedostatku času a vzhľadom na neskoré zistenie tejto informácie som nebol schopný dlhšie skúmať zmeny v protokole, a tak ho bližšie popísať.

Avšak samotná zmena celkom podstatne uľahčila detekciu Skype komunikácie, keďže nové supernódy sa pri skenovaní nástrojom nmap dajú pomerne jednoducho identifikovať a predpoklad, že ich IP adresy nebudú dynamicky obmieňané, je možné vytvoriť zoznam supernód. Možeme teda predpokladať, že každá komunikácia s niektorou z adries zo zoznamu je komunikácia Skype klienta a supernódy.

Verím, že zistené výsledky a vytvorený skript nájdu uplatnenie pri detekcií Skype komunikácie, či už v aktuálnych alebo starších záznamoch. Práca vytvára základ pre ďalšie pokračovanie, ktoré som podrobnejšie popísal v predchádzajúcej kapitole.

Literatura

[1] Nfdump. 2012.
URL http://nfdump.sourceforge.net/

[2] Nmap. 2012.
URL http://www.nmap.org/

[3] Adami, D.; Callegari, C.; Giordano, S.; aj.: A Real-Time Algorithm for Skype Traffic Detection and Classification. In *Smart Spaces and Next Generation Wired/Wireless Networking*, *Lecture Notes in Computer Science*, ročník 5764, editace S. Balandin; D. Moltchanov; Y. Koucheryavy, Springer Berlin / Heidelberg, 2009, ISBN 978-3-642-04188-4, s. 168–179, 10.1007/978-3-642-04190-7_16.
URL http://dx.doi.org/10.1007/978-3-642-04190-7_16

[4] Baset, S. A.; Schulzrinne, H. G.: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, ISSN 0743-166X, s. 1 –11, doi:10.1109/INFOCOM.2006.312.

[5] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), Říjen 2004.
URL http://www.ietf.org/rfc/rfc3954.txt

[6] Desclaux, F.; Kortchinsky, K.: Vanilla Skype part 2. 2006, [Online; accessed 22-December-2011].
URL http://www1.cs.columbia.edu/~salman/skype/skype2.pdf

[7] Goodin, D.: Skype replaces P2P supernodes with Linux boxes hosted by Microsoft @ONLINE. Květen 2012.
URL http://arstechnica.com/business/2012/05/skype-replaces-p2p-supernodes-with-linux-boxes-hosted-by-microsoft/

[8] Phaal, P.; Panchen, S.; McKee, N.: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. RFC 3176 (Informational), Září 2001.
URL http://www.ietf.org/rfc/rfc3176.txt

[9] Quittek, J.; Zseby, T.; Claise, B.; aj.: Requirements for IP Flow Information Export (IPFIX). RFC 3917 (Informational), Říjen 2004.
URL http://www.ietf.org/rfc/rfc3917.txt

[10] Trammell, B.; Boschi, E.; Procissi, G.; aj.: Identifying Skype Traffic in a Large-Scale Flow Data Repository. In *TMA*, *Lecture Notes in Computer Science*, ročník 6613,

editace J. Domingo-Pascual; Y. Shavitt; S. Uhlig, Springer, 2011, ISBN 978-3-642-20304-6, s. 72–85.

[11] Wikipedia: Netflow. 2008, [Online; accessed 22-December-2011].
URL http://cs.wikipedia.org/wiki/Netflow#Co_je_to_IP_tok

[12] Wikipedia: Skype Technologies. 2011, [Online; accessed 22-December-2011].
URL http://en.wikipedia.org/wiki/Skype_Limited#History

[13] Wireshark: Wireshark. 2012.
URL http://www.wireshark.org/

Příloha A

Použitie supernode detektora

Jedná a o skript v pythone, ktorý má na vstupe jeden povinny parameter, ktorý udáva cestu k vstupným netflow záznamom. Vrámci zachovania možnosti filtrovania záznamov je nutné zadať vstup vo tvare, aký prijíma nástroj nmap. V prípade použitia parametrov `-R` a `-M` je vhodné použiť parameter `-m` pre zoradenie jednotlivých tokov podľa ich začiatku. Skript je možné nájsť na priloženom DVD.

Príklad vzorového použitia:

```
python detect.py "-R /data/netflow_archive/LIST/2012-01-09/01 -m 'not host 0.0.0.0' "

2012-01-07 23:59:21.000940  supernode  147.229.192.14 : 36182 client  151.24.205.141 : 53472
2012-01-07 23:59:27.000241  supernode  147.229.197.3 : 46884 client  212.35.68.218 : 40209
2012-01-07 23:59:28.000732  supernode  147.229.210.220 : 52080 client  83.21.132.233 : 4893
2012-01-07 23:59:29.000946  supernode  147.229.212.153 : 7925 client  89.175.139.150 : 20524
2012-01-07 23:59:30.000081  supernode  147.229.220.134 : 17482 client  151.81.212.214 : 51297
2012-01-07 23:59:31.000273  supernode  147.229.207.17 : 4300 client  213.226.237.95 : 12055
2012-01-07 23:59:32.000168  supernode  147.229.192.76 : 25927 client  67.242.40.248 : 58217
2012-01-07 23:59:33.000968  supernode  147.229.146.146 : 18124 client  147.229.200.247 : 56269
2012-01-07 23:59:36.000395  supernode  147.229.212.170 : 25925 client  174.252.116.104 : 3556
2012-01-07 23:59:38.000808  supernode  147.229.200.143 : 18445 client  96.229.59.89 : 60368
2012-01-07 23:59:41.000969  supernode  147.229.207.17 : 4300 client  94.108.153.131 : 63529
2012-01-07 23:59:43.000011  supernode  147.229.210.220 : 52080 client  71.22.44.19 : 53441
```

Příloha B

MSnóda

Výsledok skenovania MSnódy nástrojom nmap

```
Starting Nmap 5.51 ( http://nmap.org ) at 2012-05-20 17:07 Central Europe Daylight Time
NSE: Loaded 57 scripts for scanning.
Initiating Ping Scan at 17:07
Scanning 111.221.77.158 [4 ports]
Completed Ping Scan at 17:07, 0.38s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:07
Completed Parallel DNS resolution of 1 host. at 17:07, 0.11s elapsed
Initiating SYN Stealth Scan at 17:07
Scanning 111.221.77.158 [65535 ports]

Completed SYN Stealth Scan at 17:14, 395.53s elapsed (65535 total ports)
Initiating Service scan at 17:14
Scanning 50 services on 111.221.77.158
Completed Service scan at 17:17, 205.61s elapsed (50 services on 1 host)
Initiating OS detection (try #1) against 111.221.77.158
Retrying OS detection (try #2) against 111.221.77.158
Initiating Traceroute at 17:18
Completed Traceroute at 17:18, 3.03s elapsed
Initiating Parallel DNS resolution of 18 hosts. at 17:18
Completed Parallel DNS resolution of 18 hosts. at 17:18, 0.76s elapsed
NSE: Script scanning 111.221.77.158.
Initiating NSE at 17:18
Completed NSE at 17:18, 23.05s elapsed
Nmap scan report for 111.221.77.158
Host is up (0.34s latency).
Not shown: 64531 filtered ports, 954 closed ports
PORT      STATE SERVICE     VERSION
80/tcp    open  http?
443/tcp   open  skype2      Skype
33033/tcp open  skype2      Skype
40001/tcp open  skype2      Skype
40002/tcp open  skype2      Skype
40003/tcp open  skype2      Skype
```

```
40004/tcp open  skype2      Skype
40005/tcp open  skype2      Skype
40006/tcp open  skype2      Skype
40007/tcp open  skype2      Skype
40008/tcp open  skype2      Skype
40009/tcp open  skype2      Skype
40010/tcp open  skype2      Skype
40011/tcp open  skype2      Skype
40012/tcp open  skype2      Skype
40013/tcp open  skype2      Skype
40014/tcp open  skype2      Skype
40015/tcp open  skype2      Skype
40016/tcp open  skype2      Skype
40017/tcp open  skype2      Skype
40018/tcp open  skype2      Skype
40019/tcp open  skype2      Skype
40020/tcp open  skype2      Skype
40021/tcp open  netbios-ssn
40022/tcp open  skype2      Skype
40023/tcp open  skype2      Skype
40024/tcp open  skype2      Skype
40025/tcp open  skype2      Skype
40026/tcp open  skype2      Skype
40027/tcp open  skype2      Skype
40028/tcp open  skype2      Skype
40029/tcp open  skype2      Skype
40030/tcp open  skype2      Skype
40031/tcp open  skype2      Skype
40032/tcp open  skype2      Skype
40033/tcp open  skype2      Skype
40034/tcp open  skype2      Skype
40035/tcp open  skype2      Skype
40036/tcp open  skype2      Skype
40037/tcp open  skype2      Skype
40038/tcp open  skype2      Skype
40039/tcp open  skype2      Skype
40040/tcp open  skype2      Skype
40041/tcp open  skype2      Skype
40042/tcp open  skype2      Skype
40043/tcp open  skype2      Skype
40044/tcp open  skype2      Skype
40045/tcp open  skype2      Skype
40046/tcp open  skype2      Skype
40047/tcp open  skype2      Skype

Device type: general purpose|WAP|webcam|broadband router|proxy server
Running (JUST GUESSING): Linux 2.6.X|2.4.X (98%), Gemtek embedded (89%),
Aggressive OS guesses: Linux 2.6.17 - 2.6.35 (98%), Linux 2.6.24 - 2.6.35 (94%)
```

```
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 43.518 days (since Sat Apr 07 04:52:04 2012)
Network Distance: 19 hops
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 40844/tcp)
HOP RTT       ADDRESS
1   2.00 ms   a05-sm.kn.vutbr.cz (147.229.212.1)
2   2.00 ms   gw-list.net.vutbr.cz (147.229.253.81)
3   16.00 ms  hp-kou.net.vutbr.cz (147.229.252.33)
4   4.00 ms   pe-ant.net.vutbr.cz (147.229.252.193)
5   14.00 ms  gw-ant.net.vutbr.cz (147.229.253.234)
6   ...
7   20.00 ms  20gigabitethernet1-3.core1.ams1.he.net (195.69.145.150)
8   39.00 ms  10gigabitethernet6-1.core1.lon1.he.net (184.105.213.245)
9   100.00 ms 10gigabitethernet7-4.core1.nyc4.he.net (72.52.92.241)
10  165.00 ms 10gigabitethernet5-3.core1.lax1.he.net (72.52.92.226)
11  180.00 ms pacnet.10gigabitethernet2-3.core1.lax1.he.net (216.218.223.206)
12  273.00 ms gi6-0-0.cr1.nrt1.asianetcom.net (202.147.61.166)
13  274.00 ms gi5-0-0.gw1.nrt2.asianetcom.net (202.147.0.214)
14  274.00 ms MSC-0043.gw1.nrt2.asianetcom.net (203.192.149.238)
15  275.00 ms xe-0-3-0-0.tyb-96cbe-1a.ntwk.msn.net (207.46.41.29)
16  334.00 ms ge-1-0-0-0.hkg-64cbe-1b.ntwk.msn.net (207.46.41.73)
17  335.00 ms 207.46.41.47
18  336.00 ms xe-1-0-0-0.hkn-96cbe-1a.ntwk.msn.net (207.46.41.3)
19  337.00 ms 111.221.77.158
```

Příloha C

Zoznam Msnód

Zoznam IP adries doteraz zistených MSnód je možné nájsť v súbore `supernodes.txt` na priloženom DVD.

Příloha D

Skype detekčný skript

```
1  -- @output
2  -- PORT   STATE SERVICE VERSION
3  -- 80/tcp open  skype2  Skype
4
5  require "comm"
6  require "shortport"
7
8  portrule = function(host, port)
9          return (port.number == 80 or port.number == 443 or
10                 port.service == nil or port.service == "" or
11                 port.service == "unknown")
12                and port.protocol == "tcp" and port.state == "open"
13                and port.service ~= "http" and port.service ~= "ssl/http"
14                and not(shortport.port_is_excluded(port.number,port.protocol))
15 end
16
17 action = function(host, port)
18         local status, result = comm.exchange(host, port,
19              "GET_/_HTTP/1.0\r\n\r\n", {bytes=26, proto=port.protocol})
20         if (not status) then
21                 return
22         end
23         if (result ~= "HTTP/1.0_404_Not_Found\r\n\r\n") then
24                 return
25         end
26         -- So far so good, now see if we get random data for another request
27         status, result = comm.exchange(host, port,
28                 "random_data\r\n\r\n", {bytes=15, proto=port.protocol})
29
30         if (not status) then
31                 return
32         end
33         if string.match(result, "[^%s!-~].*[^%s!-~].*[^%s!-~]") then
34                 -- Detected
35                 port.version.name = "skype2"
36                 port.version.product = "Skype"
37                 nmap.set_port_version(host, port, "hardmatched")
38                 return
39         end
40         return
41 end
```