VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE TĚŽENÍ KRYPTOMĚN POMOCÍ ANALÝZY DAT O IP TOCÍCH

DETECTION OF CRYPTOCURRENCY MINERS BASED ON IP FLOW ANALYSIS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. ERIK ŠABÍK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MARTIN ŽÁDNÍK, Ph.D.**
SUPERVISOR

BRNO 2017

Zadání diplomové práce/19931/2016/xsabik02

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Šabík Erik, Bc.**
Obor: Bezpečnost informačních technologií
Téma: **Detekce těžení kryptoměn pomocí analýzy dat o IP tocích**
Detection of Cryptocurrency Miners Based on IP Flow Analysis
Kategorie: Počítačové sítě

Pokyny:

1. Nastudujte problematiku analýzy síťového provozu a zaměřte se na problematiku kryptoměn.
2. Seznamte se se systémem pro analýzu síťových dat Nemea, který je vyvíjen organizací CESNET.
3. Navrhněte způsob detekce síťové komunikace používané při tzv. těžení kryptoměn. Detekci provádějte s využitím dat o síťových IP tocích, s využitím dostupných seznamů těžebních serverů i s využitím aktivního monitoringu.
4. Na základě navrženého přístupu vytvořte modul pro systém Nemea provádějící detekci IP adres, na kterých je prováděno těžení kryptoměn.
5. Ověřte jeho funkčnost a parametry v laboratorním prostředí ale i v praxi.
6. Diskutujte dosažené výsledky a možná rozšíření.

Literatura:

- Dle pokynů vedoucího práce.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1, 2 a 3 zadání.

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: **Žádník Martin, Ing., Ph.D.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 24. května 2017

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

prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Táto diplomová práca popisuje obecné informácie o kryptomenách, aké princípy sa využívajú pri tvorbe nových mincí a prečo môže byť ich ťaženie nežiadúce. Ďalej pojednáva o tom, čo je to IP tok a ako funguje monitorovanie sietí pomocou sledovania sieťovej komunikácie na úrovni IP tokov. Popisuje framework Nemea, ktorý slúži na vytváranie komplexných systémov pre detekciu nežiadúcej prevádzky. Vysvetľuje akým spôsobom boli získané dáta zachytávajúce komunikáciu ťaženia kryptomien a následne popisuje analýzu týchto dát. Na základe tejto analýzy je vytvorený návrh pre metódu schopnú detegovať ťaženie kryptomien pomocou záznamov o IP tokoch. Nakoniec táto správa obsahuje vyhodnotenie detekovaných udalostí v rámci rôznych sietí.

Abstract

This master's thesis describes the general information about cryptocurrencies, what principles are used in the process of creation of new coins and why mining cryptocurrencies can be malicious. Further, it discusses what is an IP flow, and how to monitor networks by monitoring network traffic using IP flows. It describes the Nemea framework that is used to build comprehensive system for detecting malicious traffic. It explains how the network data with communications of the cryptocurrencies mining process were obtained and then provides an analysis of this data. Based on this analysis a proposal is created for methods capable of detecting mining cryptocurrencies by using IP flows records. Finally, proposed detection method was evaluated on various networks and the results are further described.

Kľúčové slová

Nemea,IDS,NetFlow,IPFIX,Kryptomeny

Keywords

Nemea,IDS,NetFlow,IPFIX,Cryptocurrency

Citácia

ŠABÍK, Erik. *Detekce těžení kryptoměn pomocí analýzy dat o IP tocích*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Žádník Martin.

Detekce těžení kryptoměn pomocí analýzy dat o IP tocích

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána Ing. Martina Žádníka, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Erik Šabík
24. mája 2017

Poďakovanie

Rád by som poďakoval svojmu vedúcemu Ing. Martinovi Žádníkovi, Ph.D., za čas venovaný konzultáciam k tejto práci a za prístup k testovacím dátam. Ďalej by som tiež rád poďakoval svojej rodine za ich podporu pri štúdiu.

Obsah

1 Úvod **3**

2 Kryptomeny **4**
2.1 Obecný pohľad . . . 4
2.2 Metódy overovania správ . . . 5
2.2.1 Proof-of-work . . . 5
2.2.2 Proof-of-stake . . . 5
2.3 Pool mining . . . 6
2.4 Najpoužívanejšie kryptomeny . . . 6
2.5 Motivácia pre detekciu . . . 7

3 Monitorovanie sietí **8**
3.1 IP tok . . . 8
3.2 Protokoly pracujúce s IP tokmi . . . 9
3.2.1 NetFlow . . . 9
3.2.2 IPFIX . . . 10
3.3 Monitorovacia architektúra a jej prvky . . . 10
3.3.1 Exportér . . . 10
3.3.2 Kolektor . . . 11
3.3.3 Typy architektúr . . . 12

4 Nemea **14**
4.1 Moduly . . . 14
4.2 Rozhrania . . . 15
4.3 TRAP . . . 15
4.4 UniRec . . . 16
4.5 Nemea moduly . . . 17

5 Návrh detekčnej metódy **19**
5.1 Analýza dát sieťovej prevádzky . . . 19
5.1.1 Získanie dát . . . 19
5.1.2 Analýza dát a výber spoločných vlastností . . . 20
5.2 Pasívny spôsob detekcie . . . 20
5.3 Aktívny spôsob detekcie . . . 22
5.4 Vylepšenie detekčnej metódy . . . 24

6 Implementácia **25**
6.1 Spôsoby ukladania dát . . . 25
6.2 Štruktúra modulu . . . 27
6.3 Vstupy a výstupy modulu . . . 28
6.4 Konfigurácia . . . 29
6.5 Detekčné metódy . . . 31

7 Vyhodnotenie **33**
7.1 Rozhodovacie stromy . . . 33
7.2 Detekované udalosti . . . 34
7.3 Zhodnotenie výsledkov . . . 36

8 Záver **38**

Literatúra **40**

A Obsah CD **43**

Kapitola 1

Úvod

Kryptomeny sú dnes neoddeliteľnou súčasťou digitálneho sveta. Prvé decentralizované kryptomeny sú tu už od roku 2009 [26] a dnes s nimi môžeme nakupovať na vybraných webových stránkach [23]. Spôsoby akými vznikajú nové mince niektorých kryptomien sú ale veľmi náročné na výpočetné zdroje. V bežných prípadoch nám tento jav nevadí. Môžu sa ale nájsť ľudia, ktorý si chcú privyrobiť na úkor niekoho iného. Ako príklad môžem uviesť firemné prostredie, kde zamestnanci zneužívajú firemné prostriedky na nedovolenú ťažbu kryptomien. Ďalším príkladom môže byť prostredie školy alebo iného výskumného ústavu, v ktorom študenti využívajú prostriedky školy pre nedovolenú ťažbu kryptomien. V takýchto prípadoch sa zneužívajú prostriedky organizácií, ktoré musia následne platiť za zvýšené opotrebovanie hardware prostriedkov a tiež za zvýšený odber elektrickej energie. Je teda v záujme týchto organizácií detegovať ťaženie kryptomien vo svojom prostredí.

Nasledujúca kapitola 2 obsahuje popis čo sú to kryptomeny, aké princípy sa využívajú pri tvorbe nových mincí v kryptomenách a prečo môžu byť v niektorých prípadoch nežiadúce. V kapitole 3 popíšem čo je to IP tok. Ďalej v tejto kapitole popíšem systémy a princípy, ktoré využívajú IP toky na sledovanie sieťovej prevádzky a nakoniec stručne opíšem monitorovacie architektúry, ktoré sa dnes využívajú. V kapitole 4 priblížim čo je to framework Nemea, ako tento framework funguje a tiež popíšem niektoré detekčné moduly obsiahnuté vo frameworku Nemea, ktoré detegujú nežiadúcu sieťovú prevádzku na základe vzorov. Následne v kapitole 5 navrhnem spôsob detekcie IP adries, na ktorých prebieha ťaženie kryptomien. Tento spôsob detekcie bude schopný na základe informácií z IP tokov určiť, ktorá sieťová komunikácia je podozrivá v rámci ťaženia kryptomien. V návrhu tiež popíšem spôsob aktívneho dotazovania, ktorý bude schopný s istotou určiť či sa jedná o server určený na ťaženie kryptomien. Následujúca kapitola 6 obsahuje podrobný popis implementácie navrhnutých detekčných metód vo forme detekčného modulu pre framework Nemea implementovaného v jazyku C++. Okrem popisu samotných detekčných metód, táto kapitola popisuje aj spôsoby uchovávania dát, vstupy a výstupy detekčného modulu a tiež konfiguráciu tohto modulu. Predposledná kapitola 7 popisuje detekované udalosti z rôznych sietí, v ktorých bol detekčný modul nasadený a tiež obsahuje zhodnotenie týchto výsledkov. V poslednej kapitole 8 som zhodnotil celú prácu, popísal som v nej prínos detekčného modulu a nakoniec som opísal možné vylepšenia do budúcnosti.

Táto práca nadväzovala na prácu zo semestrálneho projektu. Konkrétne som zo semestrálneho projektu prebral a mierne upravil kapitolu 2, kapitolu 3, kapitolu 4 a kapitolu 5.

Kapitola 2

Kryptomeny

V tejto kapitole popíšem čo sú kryptomeny, v čom sa všeobecne od seba jednotlivé kryptomeny líšia, popíšem tiež princíp ich fungovania a na záver kapitoly vymenujem najpoužívanejšie z nich.

2.1 Obecný pohľad

Digitálne meny nahrádzajú klasické štátom podporované meny digitálnou verziou, ktorú je ťažšie sfalšovať, ktorá ľahko preklene medzinárodné hranice, je ju možné mať uloženú na vlastnom pevnom disku namiesto v banke a možno pre väčšinu užívateľov najdôležitejšie - nie je subjektom inflácie pri tlačení nových peňazí. [28]

Viac technickejšia definícia je, že digitálne meny sú peniaze vyjadrené ako reťazec bitov poslaný ako správa sieťou, ktorá overí autenticitu správy rôznymi mechanizmami, ako napr. proof-of-work (PoW) alebo proof-of-stake (PoS), oba mechanizmy budú vysvetlené nižšie. Väčšina digitálnych mien vystavuje verejne viditeľnú, distribuovanú účtovnú knihu, ktorá je zdielaná naprieč celou sieťou. To čo odlišuje jednotlivé digitálne meny je spôsob akým jej užívatelia sú dohodnutý na zmenách účtovnej knihy (inými slovami, ktoré transakcie sa budú akceptovať ako platné) a mechanizmus podľa ktorého sa bude odmeňovať validačný proces transakcií. [31]

Informácie o jednotlivých správach (transakciách) a tiež o tom, ktorý užívateľ má koľko mincí sú uložené vo verejne viditeľnej, distribuovanej účtovnej knihe nazývanej angl. blockchain. Ako vyplýva z názvu ide o reťaz blokov, pričom každý blok má v sebe niekoľko transakcií. Jednotlivé bloky sa za seba reťazia tak, že nový blok v sebe obsahuje výsledok hash funkcie, teda kryptografický otlačok predošlého bloku. Transakcie v blockchain sa berú ako prevedené a platné. Vytváranie nových blokov je teda nutným procesom v sieti. Aby sa ale zachovala integrita siete je nutné aby vytváranie blokov nebolo také jednoduché a vyžadovalo nejaký čas. Metódy vytvárania nových blokov a overovanie transakcií som popísal v nasledujúcej sekcii.

Kryptomeny sú teda digitálne meny využívajúce kryptografické funkcie pre zaručenie integrity a autenticity. Využíva sa napr. asymetrická kryptografia, ktorá slúži na podpisovanie transakcií. Klient ktorý vykonáva nejakú transakciu ju podpíše svojím privátnym kľúčom. Zvyšok siete následne transakciu overí za pomoci verejného kľúča autora transakcie. [20] Z tohto vyplýva aj isté riziko, pri strate privátneho kľúča užívateľ stratí aj svoje peniaze.

2.2 Metódy overovania správ

V distribuovanej sieti kryptomien existujú dva typy užívateľov:

- Klasický užívateľ - V sieti vykonáva iba transakcie resp. prevody medzi sebou a inými užívateľmi. Teda buď môže prijať nejaké peniaze alebo niekomu peniaze poslať
- Miner - Tento typ užívateľa overuje transakcie, hromadí ich a následne súperí s ostatnými miner užívateľmi o to, komu sa ako prvému podarí vytvoriť nový blok s týmito transakciami. Vytvorenie nového bloku je výpočetne náročná úloha. Za vytvorenie nového bloku je ale víťaznému miner užívateľovi pridelená odmena v podobne nejakej čiastky mincí. Miner užívatelia teda jednak overujú transakcie ale zároveň aj vytvárajú nové peniaze.

2.2.1 Proof-of-work

Pojem Proof-of-work (PoW) bol formalizovaný v technickej správe [29]. Zavedený bol ale už predtým, pričom sa jedná o mechanizmus alebo funkciu, ktorým je možné zabrániť rôznym typom útokov ako DoS[1] alebo spam[2] útokom takým spôsobom, že žiadateľ o službu musí vykonať nejakú prácu (proces). Poskytovateľ služby naopak overuje či žiadateľ túto prácu vykonal. Hlavným znakom tohoto mechanizmu je, že vykonanie práce je výpočetne náročné ale overenie či práca bola vykonaná je výpočetne jednoduché. [27]

PoW sa v niektorých kryptomenách využíva pri vytváraní nových blokov. PoW vyžaduje aby miner klient našiel také číslo nazývané *nonce*, že ak obsah nového bloku a nonce dáme na vstup hash funkcie, tak jej výstup bude číselne menší než nejaké iné verejne známe číslo, nazývané tiež zložitosť siete. Miner klienti teda súťažia o to, kto nájde ako prvý číslo nonce. Ten klient ktorý takéto číslo nájde ho rozošle do siete na overenie. Po overení si ostatný klienti siete pridajú novo vytvorený blok do blockchain a víťazný klient získa odmenu za vytvorenie bloku. [20]

2.2.2 Proof-of-stake

Proof-of-stake (PoS) je metóda, ktorou sa niektoré kryptomeny snažia zaviesť do vytvárania nových blokov distribuovanosť medzi užívateľmi. Narozdiel od PoW metódy kde záleží na výpočetnom výkone klienta, v prípade PoS metódy záleží na niečom čo klient vlastní, napr. počet mincí v sieti. Ak by ale záležalo iba na výške konta klienta, nové bloky by vytvárali iba klienti s najväčším objemom mincí. Kryptomena Peercoin [30] využíva princíp PoS, a zavádza koncept *coin age*. Pravdepodobnosť že nejaký klient vytvorí nový blok záleží na hodnote, ktorá je odvodená od súčinu počtu jeho mincí na konte a počtu dní koľko tento počet mincí na konte bol. Minimálny počet dní, ktoré je treba držať peniaze na konte je 30 a až potom je možné súperiť o vytváranie nového bloku. Čím viac mincí ma užívateľ na svojom konte a čím dlhšie ich tam má, tým ma väčšiu pravdepodobnosť vytvorenia nového bloku. Táto pravdepodobnosť stúpa až po dobu 90 dní, kedy dosiahne svoje maximum. V prípade že užívateľ výhra tvorbu nového bloku sa mu hodnota coin age vynuluje a začína odznova.

Výhoda PoS oproti PoW je tá, že nie je vyžadovaný príliš vysoký výpočetný výkon, z toho vyplýva že nie je potrebný vysokovýkonný hardware a tiež menšie nároky na elektrickú energiu.

[1] Zamietnutie služby - denial of service
[2] Nevyžiadaná pošta

2.3 Pool mining

Klasický spôsob ťaženia mincí nazývané tiež *solo mining*, kde jednotlivý klienti medzi sebou súperia o to kto ako prvý vyrieši nejakú výpočetne náročnú úlohu, pre bežného človeka bez špeciálneho hardware sa dnes finančne nevyplatí. Existujú tiež kalkulačky, ktoré nám vypočítajú či sa nám oplatí ťažiť na našom hardware. [22]

Pretože zložitosť pre vytváranie nových blokov už je tak vysoká, že sa to jednotlivcom nevyplatí, začali sa užívatelia spájať do skupín a vzniká systém nazývaný *pool minig*. Tento systém funguje tak, že užívatelia v skupine zdieľajú svoj výpočetný výkon a na vytváraní nového bloku spolupracujú. Ak sa podarí jednému užívateľovi zo skupiny vytvoriť nový blok, odmena za tento blok sa rozdelí v skupine medzi všetkých užívateľov. Odmena sa delí medzi užívateľov v pomere, v akom prispeli svojím výpočetným výkonom do celkového výpočetného výkonu skupiny. Takýto systém má oproti solo mining metóde, kde užívateľ ma veľmi malú šancu vyhrať veľkú odmenu, tú výhodu, že každý zapojený užívateľ s dostatočným výpočetným výkonom v skupine má garantovanú nejakú odmenu. [32]

Spôsob akým pool minig systém funguje, je že existuje jeden server nazývaný *pool server*, ktorý koriguje všetkých zapojených užívateľov v skupine. Jedná sa teda o klient-server architektúru. Klienti teda už nekomunikujú s ostatnými pomocou P2P[3] siete, ale komunikujú výhradne s pool serverom.

2.4 Najpoužívanejšie kryptomeny

Momentálne existuje viac ako 1000 kryptomien [8]. Väčšina z nich ale má na trhu malú hodnotu [7] a preto je pravdepodobné, že užívateľská základná týchto kryptomien bude tiež malá. Toto môže byť spôsobené rôznymi faktormi ako napr. možnosť prevodu kryptomeny na klasické peniaze.

Obrázok 2.1 zobrazuje TOP 5 kryptomien podľa ich hodnoty na trhu. Je teda vhodné sa zamerať pri návrhu detekčnej metódy hlavne na tieto kryptomeny.

#	Name	Market Cap	Price	Available Supply
1	Bitcoin	$15,290,486,708	$951.23	16,074,437 BTC
2	Ethereum	$696,822,826	$7.97	87,444,762 ETH
3	Ripple	$232,787,636	$0.006406	36,337,298,649 XRP *
4	Litecoin	$211,434,004	$4.30	49,122,604 LTC
5	Monero	$178,797,246	$13.09	13,657,715 XMR

Obr. 2.1: Ukážka trhovej kapitalizácie kryptomien ku dňu 31. December 2016 [7]

[3] Peer-to-peer

2.5 Motivácia pre detekciu

Samotný koncept kryptomien nie je škodlivý. Škodlivým alebo nechceným sa ale môže stať v prípade, že proces ťaženia bude prebiehať na zariadeniach, bez súhlasu ich majiteľa. Software pre ťaženie rôznych kryptomien môže byť distribuovaný ako malware, pričom na hostiteľských staniciach bude zaberať výpočetné zdroje, opotrebovávať hardware a spôsobovať vysoké účty za elektrickú energiu. Z tohoto dôvodu je vhodné vykonávať detekciu rôznych ťažiarov kryptomien.

Detekcia môže prebiehať priamo na hostiteľskej stanici alebo monitorovaním sieťovej prevádzky. Pri detekcii na hostiteľskej stanici si všímame vysoké vyťaženie výpočetných jednotiek, či už CPU alebo GPU. V niektorých prípadoch ale takáto detekcia nie je možná - nemáme prístup na hostiteľskú stanicu. Preto je nutné byť schopný detegovať proces ťaženia iba na základe sieťovej prevádzky.

Detekcia sa tiež nemusí zameriavať na všetky kryptomeny. Ako som popísal vyššie, kryptomeny využívajúce koncept proof-of-stake nevyťažujú výpočetné zdroje a preto ich nie je nutné detegovať. Taktiež detekcia solo mining metódy ťaženia nie je perspektívna, pretože v nej figuruje málo užívateľov z dôvodu aký som opísal vyššie. Čo ale má zmysel detegovať a na čo sa aj ďalej budem zameriavať je práve proof-of-work koncept a pool mining, pretože práve táto kombinácia je rozšírená medzi užívateľmi a tiež využíva veľké výpočetné zdroje.

Kapitola 3

Monitorovanie sietí

Monitorovanie sietí je proces, ktorý vykonávame za účelom dozvedieť sa informácie o množstve a type sieťovej prevádzky v monitorovanej sieti. V tejto kapitole popíšem čo je to IP tok, stručne popíšem protokoly ktoré ho využívajú a vysvetlím akú rolu hrá v monitorovaní sietí.

Informácie získané monitorovaním sietí sa môžu využiť následovne:

- Optimalizácia sieťovej topológie alebo smerovacích pravidiel.
- Analýza sieťových aplikácií alebo užívateľov.
- Na základe počtu prenesených dát môže prebiehať účtovanie, napr. účtovanie zákazníka poskytovateľom internetového pripojenia.
- Ukladanie záznamov o aktivitách na sieti poskytovateľom internetových služieb pre prípadné neskoršie dohľadanie incidentov.
- Bezpečnostná analýza sieťovej prevádzky v reálnom čase, prípadne niekedy v budúcnosti pomocou uložených dát.

3.1 IP tok

IP tok je podľa [24] definovaný ako jednosmerná sekvencia paketov so spoločnými vlastnosťami, ktorá prejde cez monitorovací bod za určitý časový interval. Keďže sa jedná o jednosmernú sekvenciu tak pre každé spojenie dvoch staníc budú existovať dva IP toky, pre každý smer jeden IP tok. Spoločné vlastnosti IP toku na základe ktorých prebieha agregácia:

- Zdrojová a cieľová IP adresa
- Zdrojový a cieľový port
- Protokol sieťovej vrstvy ISO/OSI modelu

IP tok okrem vyššie spomenutých vlastností obsahuje aj ďalšie dodatočné informácie. Týmito informáciami môžu byť napr. počet paketov, ktoré boli prenesené v rámci IP toku, počet bajtov, časové značky kedy IP tok vznikol a kedy skončil a ďalšie. Niektoré protokoly pre monitorovanie sietí dovoľujú pridávať užívateľom definované položky. Pri použití takéhoto protokolu môže IP tok obsahovať aj napr. informácie z aplikačnej vrstvy ISO/OSI

modelu. V prípade ale, že protokol ktorý na monitorovanie sietí používame nepodporuje pridávanie užívateľom definované položky, sme obmedzení iba na protokolom definovanú množinu položiek.

3.2 Protokoly pracujúce s IP tokmi

Na pracú s IP tokmi môžeme využívať rôzne protokoly, v tejto sekcii popíšem dva najpoužívanejšie z nich.

3.2.1 NetFlow

NetFlow [2] je protokol pre prácu s IP tokmi, vyvinutý spoločnosťou Cisco Systems. NetFlow protokol má viacero verzií (v dobe písania tejto práce to boli verzie 1 až 9). Najpoužívanejšie sú verzia 5 a verzia 9 a preto sa v tejto sekcii budem venovať iba týmto verziám.

Verzia 5

NetFlow verzia 5 využíva na odosielanie informácií o IP tokoch protokol UDP. Datagram tohoto protokolu obsahuje hlavičku a záznam o jednom, poprípade viac IP tokoch. V hlavičke datagramu sa nachádza verzia protokolu, počet záznamov v datagrame a ďalšie informácie. Táto verzia nedovoľuje definovať užívateľovi vlastné položky. To znamená že položky nesúce informácie sú pevne dané a ich formát nie je možné meniť. Veľkým nedostatkom verzie 5 je tiež to, že chýba podpora IPv6 tokov. Pre úplnosť uvediem všetky položky ktoré táto verzia obsahuje:

- Zdrojová IP adresa
- Cieľová IP adresa
- Next hop IP adresa - IP adresa ďalšie routra
- Číslo vstupného rozhrania
- Číslo výstupného rozhrania
- Počet bajtov v IP toku
- Počet paketov v IP toku
- Čas príchodu prvého paketu IP toku
- Čas príchodu posledného paketu IP toku
- TCP/UDP zdrojový port
- TCP/UDP cieľový port
- Zjednotenie všetkých TCP príznakov pomocou bitového súčtu (operácia OR)
- Číslo protokolu 3. vrstvy modelu ISO/OSI
- Typ služby
- Číslo zdrojového autonómneho systému

- Číslo cieľového autonómneho systému
- Zdrojová maska podsiete
- Cieľová maska podsiete

Verzia 9

NetFlow verzia 9 [24] má oproti verzii 5 veľkú výhodu v tom, že formát správ záznamov o IP tokoch je flexibilný. Tejto flexibility sa dosiahlo využitím šablón. Šablóna popisuje typy položiek, ktoré sa nachádzajú v zázname o IP toku. Datagram NetFlow verzie 9 rovnako ako verzia 5 obsahuje hlavičku, záznam o jednom poprípade viac IP tokoch a naviac obsahuje jednu, poprípade viac šablón. Vďaka využitiu šablón je možné upravovať formát správ, teda používať rôzne položky podľa aktuálnej potreby. V tejto verzii tiež pribudla podpora IPv6 tokov a tiež podpora pre iné položky ktoré verzia 5 nepodporovala. Verzia 9 dovoľuje exportovať záznamy o IP tokoch nielen pomocou protokolu UDP (ako tomu bolo vo verzii 5), ale aj pomocou iných protokolov ako napr. TCP, SCTP a iných.

3.2.2 IPFIX

IPFIX [25] vznikol ako rozšírenie protokolu NetFlow. Jedná sa teda o NetFlow verziu 10, pričom táto verzia bola prehlásená za Štandard IETF. IPFIX oproti NetFlow ma výhodu v tom že dovoľuje užívateľovi špecifikovať vlastné položky, ktoré sa budú exportovať. IPFIX je v tomto oveľa flexibilnejší ako NetFlow protokol. IPFIX protokol každú položku identifikuje unikátnym identifikačným číslom, na základe ktorého jednoznačne určuje o akú položku sa jedná. Okrem tohoto čísla, používa IPFIX protokol naviac ešte ďalšie tzv. číslo spoločnosti. Toto číslo sa využíva pri identifikácií položiek, ktoré boli špecifikované užívateľom alebo nejakou spoločnosťou. Základné položky ako napr. číslo portu, TCP príznaky, počet bajtov a pod. majú číslo spoločnosti rovné 0. Jednotlivé položky [1] ako aj prideľovanie čísla spoločnosti [5] spravuje organizácia IANA (Internet Assigned Numbers Authority). IPFIX je tiež rovnako ako NetFlow verzie 9 nezávislý na transportnom protokole.

3.3 Monitorovacia architektúra a jej prvky

V tejto kapitole popíšem čo je to monitorovacia architektúra, exportér, kolektor a aké monitorovacie architektúry sa dnes používajú. Monitorovacia architektúra sa typicky skladá z niekoľkých exportérov a jedného kolektoru. Vzhľadom nato že architektúra pre NetFlow a pre IPFIX je podobná, vysvetlím jej princíp iba pre NetFlow, pričom pre IPFIX je princíp analogický.

3.3.1 Exportér

Exportér je zariadenie, ktoré je pripojené k monitorovanej sieti a ktoré zachytáva pakety prechádzajúce touto sieťou. Na základe informácií z týchto paketov vytvára v pamäti záznamy o IP tokoch. Tieto záznamy sú v pamäti uložené tak dlho dokedy neexpirujú. Po expirácii budú tieto záznamy o IP tokoch odoslané kolektoru pomocou príslušného protokolu (NetFlow alebo IPFIX).

Udalosti vedúce k expirovaniu záznamu o IP toku v pamäti:

- Prekročenie časovej hranice u aktívneho IP toku. To znamená že po určitom čase (typicky 5 minút), prebehne expirácia záznamu aj v prípade, že pre tento záznam stále prichádzajú nové pakety.
- Prekročenie časovej hranice u neaktívneho IP toku. To znamená že po určitom čase (typicky 30 sekúnd), prebehne expirácia záznamu v prípade, že pre daný záznam nepríde žiadny ďalší paket.
- V prípade že sa jedná o TCP spojenie a bol detekovaný paket obsahujúci FIN (koniec spojenia) alebo RST (reset spojenia) príznak.
- V prípade že zaplnenie pamäte pre ukladanie záznamov o IP tokoch je nad určitou hodnotou.

Príkladom exportéru môže byť smerovač podporujúci zbieranie a exportovanie štatistík o IP tokoch alebo samostatne stojaca sonda ako napr. FlowMon sonda [11] od firmy Flowmon Networks. Flowmon sonda je pasívna autonómna sonda (samostatne stojaca sonda, neupravujúca monitorované dáta), ktorá monitoruje sieťovú prevádzku v sieti a vytvára štatistiky o tejto prevádzke vo formáte NetFlow verzie 5, NetFlow verzie 9 alebo IPFIX.

3.3.2 Kolektor

Kolektor je zariadenie, ktoré prijíma dáta od exportéru. Tieto dáta následne ukladá do databáze alebo na disk. V špeciálnych prípadoch môže kolektor prijaté dáta preposlať na ďalší kolektor. Formát v ktorom budú dáta uložené závisí od toho, aké položky chceme sledovať alebo aké nástroje na vizualizáciu dát chceme použiť. Typicky sa používa formát Nfdump [4] pre ukladanie NetFlow záznamov o IP tokoch. Tento formát, rovnako ako NetFlow protokol, neumožňuje ukladať užívateľom definované položky. Preto ak je potreba ukladať aj užívateľom definované položky, (napr. rôzne údaje z aplikačných protokolov) je nutné použiť iný formát pre ukladanie dát. Jednou z možností je použiť FastBit [10] databázu, ktorá toto umožňuje. Kolektor môže taktiež dáta preposlať ďalej na hlbšiu analýzu, napr. do systému Nemea, viď kapitola 4.

Príklad kolektoru môže byť aplikácia vyvíjaná združením CESNET [14] s názvom IPFIXcol [12]. Jedná sa o open-source implementáciu IPFIX kolektoru v jazyku C/C++ podla špecifikácie v RFC7011 [25]. Hlavnou výhodou tohoto riešenia je schopnosť jednoducho pridávať užívateľom implementované zásuvné moduly (angl. *plugin*). Použité zásuvné moduly v aplikácii IPFIXcol patria do jednej z následujúcich troch hlavných skupín:

- Input: Do tejto skupiny patria zásuvné moduly, ktorých úlohou je počúvať na špecifikovanom sieťovom rozhraní a prijímať z neho správy, ktoré odosiela exportér pomocou podporovaného transportného protokolu. Tieto správy ďalej musia spracovávať podľa toho o aký monitorovací protokol sa jedná (NetFlow alebo IPFIX) a následne pomocou funkcií aplikácie IPFIXcol poslať tieto dáta ďalším zásuvným modulom. Príkladom takýchto zásuvných modulov sú UDP input plugin alebo TCP input plugin.
- Intermediate: Zásuvné moduly v tejto skupine prijímajú dáta z input zásuvných modulov a podľa potreby v nich upravujú niektoré položky. Ako príklad môže slúžiť zásuvný modul anonymization, ktorý anonymizuje IP adresy pomocou knižnice CryptoPAn [6]. Po úprave sa dáta následne posielajú output/storage zásuvným modulom.

- Output/Storage: Jedná sa o zásuvné moduly, ktoré špecifikujú výstupný formát záznamov o IP tokoch a taktiež spôsob výstupu. Môže sa jednať o zápis do súboru na disku, zápis do databáze, preposlanie dát inej instancii aplikácie IPFIXcol, preposlanie dát do systému Nemea a pod. Príkladom môže byť fastbit storage plugin, ktorý umožňuje ukladať prijaté správy do FastBit databáze.

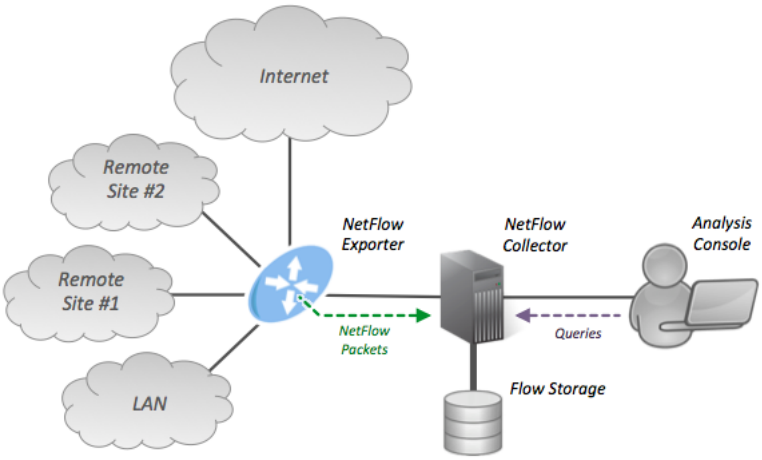
Pomocou týchto zásuvných modulov je možné docieliť úpravu položiek záznamu o IP toku alebo tiež spôsob ich ukladania. Využitie aplikácie IPFIXcol je výhodné aj z toho dôvodu, že obsahuje zásuvný modul pre preposielanie záznamov o IP tokoch do systému Nemea a tým dovoľuje využívať systém Nemea na analýzu dát v reálnom čase.

3.3.3 Typy architektúr

V tejto sekcii sa budem venovať dvom hlavným typom architektúr, s ktorými sa môžeme v dnešnej dobe stretnúť. Časť informácií v tejto kapitole som čerpal z online zdroja [3].

Architektúra využívajúca routre

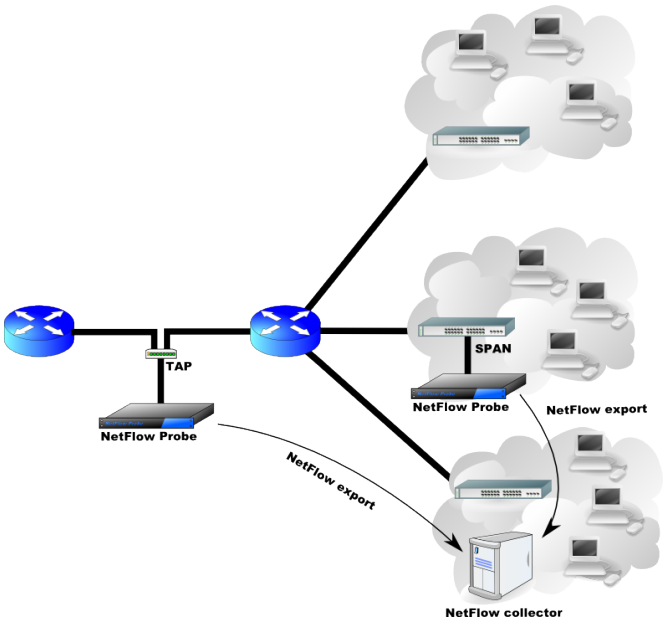
Táto architektúra využíva smerovače, ktoré sa nachádzajú na hraničných bodoch siete, ako exportéri. Tieto smerovače musia dokázať analyzovať pakety a vytvárať z nich záznamy o IP tokoch. Toto riešenie má dve hlavné nevýhody. Prvou nevýhodou je cena zariadenia (smerovača), z čoho vyplýva že toto riešenie nemusí byť vhodné pre malé siete. Samotná analýza paketov a spracovanie štatistiky o IP tokoch taktiež obmedzuje celkový výkon zariadenia, čo u menej výkonných zariadeniach môže viesť až ku vzorkovaniu paketov z ktorých sa vytvárajú IP toky. Druhou nevýhodou môže byť malá množina podporovaných protokolov pre export záznamov o IP tokoch. Ako príklad môžem uviesť Cisco smerovače podporujúce výhradne protokol NetFlow. Z toho vyplýva že s použitím daného zariadenia nieje možné exportovať užívateľom definované položky, pretože ako bolo popísane vyššie, protokol NetFlow to neumožňuje.



Obr. 3.1: Ukážka architektúry využívajúcej smerovač [3]

Architektúra využívajúca sondy

V tejto architektúre sa využívajú pasívne sondy ako exportéri. Jedná sa o špeciálne zariadenie určené výhradne na monitorovanie siete a export štatistík o IP tokoch. Táto architektúra prináša hneď niekoľko výhod. Prvou z nich je cena zariadenia, ktorá oproti smerovaču poskytujúcemu rovnakú funkcionalitu môže byť výrazne nižšia. Ďalšou výhodou je, že toto zariadenie je možné pripojiť transparentne do ľubovolného bodu v sieti pomocou TAP rozhrania. Je treba si ale uvedomiť že v tomto prípade bude monitorovanie prebiehať iba v tomto bode siete. Je tiež možné zapojiť sondu do siete pomocou zrkadlenia portov (angl. *port mirroring*) na smerovači a teda monitorovať sieťovú prevádzku prechádzajúcu daným smerovačom. Štatistiky o IP tokoch môže sonda odosielať do kolektoru po odlišnej sieťovej linke akú monitoruje, a tým nezaťažovať monitorovanú sieť. Výhodou použitia sondy namiesto smerovača taktiež môže byť podpora viacerých monitorovacích protokolov alebo možnosť implementácie a nasadenie vlastného algoritmu pre analýzu sieťovej prevádzky.



Obr. 3.2: Ukážka architektúry využívajúcej sondu [3]

Kapitola 4

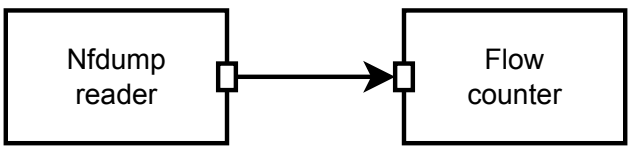
Nemea

Nemea (Network Measurements and Analysis) je framework, ktorý umožňuje tvorbu systému pre automatickú analýzu záznamov o IP tokoch v reálnom čase. Tento systém sa skladá z oddeliteľných blokov nazývaných moduly. Jednotlivé moduly sú medzi sebou prepojené rozhraniami a ako celok môžu spracovávať, analyzovať a následne vytvárať správy o rôznych sieťových incidentoch. Väčšinu informácií popísaných v tejto kapitole som čerpal z príslušnej technickej správy [21].

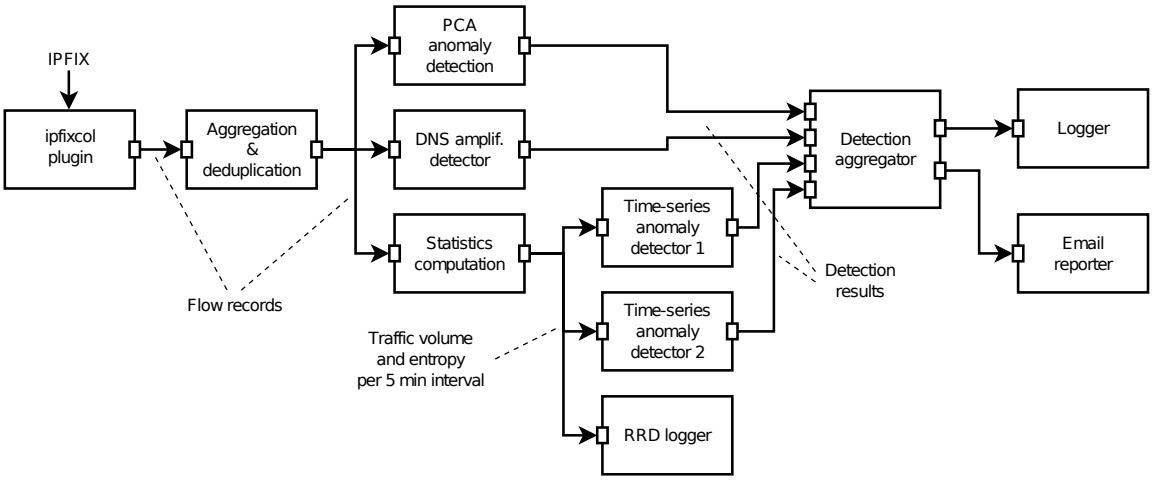
4.1 Moduly

Každý modul spustený v systéme je samostatne bežiaci program, ktorý nejakým spôsobom pracuje so záznamom o IP toku. Väčšina modulov funguje ako filter, teda príjmu dáta na vstupnom rozhraní, tieto dáta spracujú a odošlú pomocou výstupného rozhrania. Moduly môžu počítať rôzne štatistiky z prichádzajúcich záznamov o IP tokoch alebo hľadať typické vzory útokov a následne odosielať výsledok tohoto procesu pomocou výstupného rozhrania. Ďalší modul môže tieto výsledky spracovávať, agregovať alebo korelovať s ostatnými výsledkami iných modulov. Týmto spôsobom je možné vytvoriť komplexný systém pre analýzu sieťovej prevádzky v reálnom čase. Moduly je možné implementovať v jazyku C, C++ alebo Python.

Na obrázku 4.1 môžeme vidieť minimálne zapojenie, ktoré pozostáva z dvoch modulov. Prvý modul v tomto zapojení načítava záznamy o IP tokoch zo súboru a odosiela ich cez jeho výstupné rozhranie. Druhý modul dáta prijme a počíta z nich celkový počet IP tokov, paketov a bajtov. Na obrázku 4.2 môžeme vidieť komplexnejšiu konfiguráciu pozostávajúcu z viacerých modulov, v ktorej IP toky sú získavané zo siete v reálnom čase pomocou zásuvného modulu pre IPFIX kolektor. IP toky sú analyzované niekoľkými algoritmami, výsledky z tejto analýzy sú agregované a následne z týchto agregovaných výsledkov je vytvorená správa, ktorá sa odosiela externému systému. Takýmto systémom pre zber nahlásených udalostí môže byť napr. systém Warden [18], ktorý je vyvíjaný združením CESNET.



Obr. 4.1: Minimálna konfigurácia systému Nemea [21]



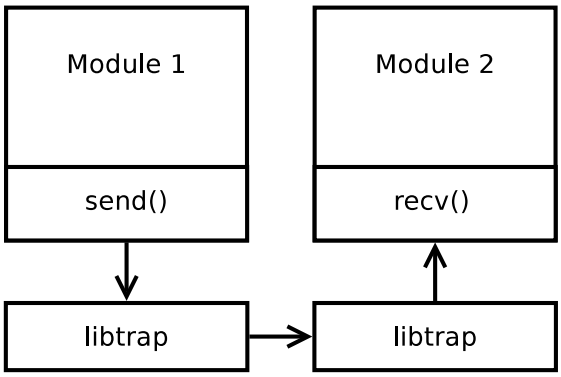
Obr. 4.2: Komplexná konfigurácia systému Nemea [21]

4.2 Rozhrania

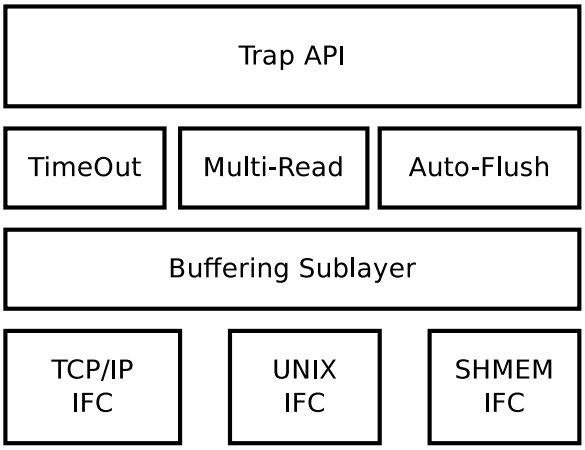
Rozhrania slúžia pre jednosmernú komunikáciu modulov. Dáta ktoré odosielajú sú vo forme záznamov. Všetky záznamy posielané cez jedno rozhranie musia mať rovnaký formát. To znamená že formát výstupného rozhrania modulu ktorý dáta odosiela a vstupného rozhrania modulu ktorý dáta prijíma musí byť rovnaký. Formát špecifikuje ktoré položky sa nachádzajú danom zázname. Formát pre nejaké rozhranie je špecifikovaný dynamicky v momente kedy sa modul pripája do systému. Dynamická špecifikácia formátu dovoľuje napr. pridať novú položku do tohto formátu bez potreby meniť kód modulov, ktoré pracujú s týmto formátom. Protokol ktorý špecifikuje ako definovať tieto formáty, ako ich vytvárať a používať sa nazýva UniRec, viď sekcia 4.4.

4.3 TRAP

TRAP (Traffic Analysis Platform) je knižnica, ktorá efektívne implementuje rozhrania používané Nemea modulmi. Táto knižnica je linkovaná ku každému Nemea modulu. Obrázok 4.3 zobrazuje koncept komunikácie medzi dvoma modulmi. TRAP abstrahuje modul od aktuálneho rozhrania a jeho špecifík, čím uľahčuje prácu tvorcom modulov. Odosielací modul odosiela dáta ihneď ako sú dostupné. Operácia odoslania dát môže byť podľa konfigurácie neblokujúca alebo blokujúca. Prijímací modul číta dáta zo vstupu, pričom táto operácia môže byť taktiež neblokujúca alebo blokujúca. TRAP sa taktiež stará o bufferovanie dát a o ich zahadzovanie podľa konfigurácie. Na obrázku 4.4 je zobrazená architektúra knižnice TRAP.



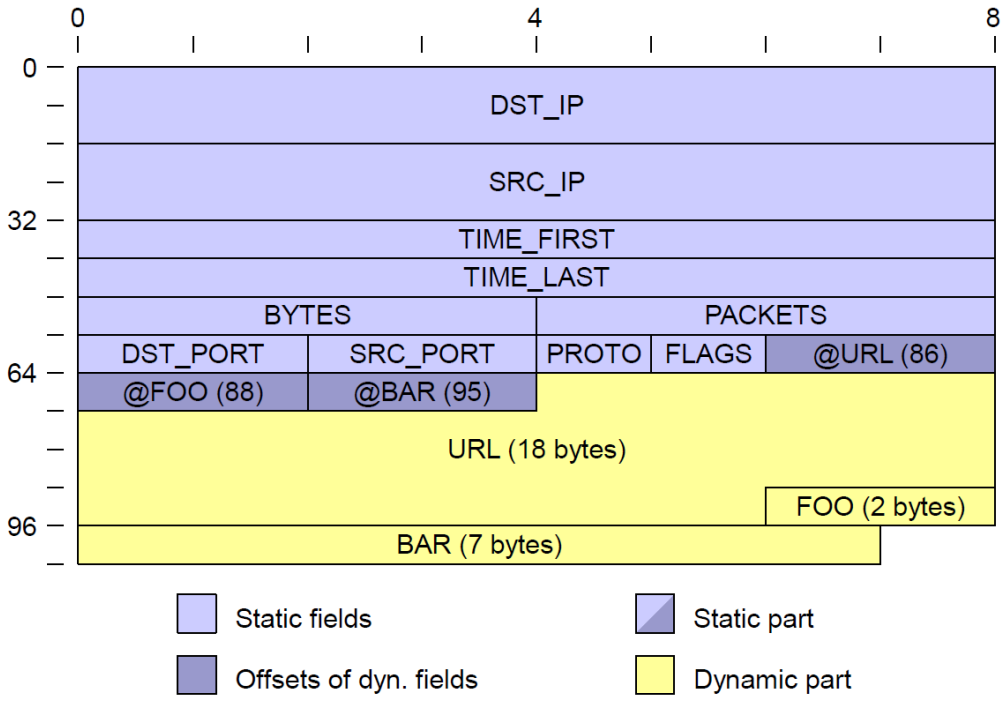
Obr. 4.3: Typická komunikácia medzi dvoma modulmi v systéme Nemea [21]



Obr. 4.4: Architektúra libtrap knižnice [21]

4.4 UniRec

UniRec (Unified Record) je špecifický formát správ (záznam), ktoré sa posielajú pomocou TRAP rozhraní. Pozostáva z niekoľkých položiek, pričom každá položka ma svoje meno a dátový typ. Zoznam položiek v zázname sa nazýva šablóna (angl. *template*). Šablóna je špecifikovaná vymenovaním všetkých jej položiek. Pretože TRAP rozhranie používa práve jednu šablónu, tak všetky správy poslané cez toto rozhranie majú rovnaký formát. Akékoľvek dva moduly spojené pomocou TRAP rozhrania musia používať rovnakú šablónu na tomto rozhraní. Šablóna sa pre odosielajúci aj pre prijímací modul špecifikuje pri inicializácii modulu. UniRec záznam obsahuje statickú časť, pre položky ktorých veľkosť sa v čase nemení, a dynamickú časť, pre položky ktorých veľkosť sa môže s časom meniť. Obrázok 4.5 zobrazuje príklad ako môže vyzerať UniRec záznam.



Obr. 4.5: Príklad UniRec záznamu reprezentujúceho základný IP tok rozšírený o niekoľko dynamických položiek [21]

Na poradí položiek v špecifikácii UniRec šablóny nezáleží. Poradie položiek v samotnom UniRec zázname je ale presne definované podľa nasledujúcich pravidiel:

- Ako prvé sú uložené statické položky, za nimi nasledujú offsety dynamických položiek a na konci sú uložené samotné dynamické položky.
- Položky v statickej časti sú zoradené zostupne podľa ich veľkosti.
- Položky s rovnakou veľkosťou sú zoradené abecedne podľa ich názvu.

4.5 Nemea moduly

V dobe písania tejto práce obsahoval framework Nemea viac ako 20 modulov. Nemea moduly obvykle implementujú základné spracovanie alebo analýzu IP tokov. V tejto sekcii popíšem tie detekčné moduly, ktoré sa zameriavajú na detekciu škodlivej komunikácie pomocou dopredu známeho vzoru, pričom tieto komunikácie sú typu je klient-server.

- HostStatsNemea - Detekčný modul, ktorý je schopný detegovať rôzne typy sieťových útokov. Modul počas celého svojho behu počíta štatistiky zo záznamov o IP tokoch pre každú IP adresu, ktorú v týchto záznamoch nájde. Paralelne s tým modul v určitých časových intervaloch vyhľadáva v týchto štatistikách typické vzory útokov. Tieto vzory môžu byť napr. horizontálne skenovanie siete, útoky hrubou silou alebo DoS útoky. Akonáhle je nejaká udalosť detegovaná modul odošle správu o detegovanej udalosti pomocou výstupného rozhrania.

- BruteForceDetector - Detekčný modul, ktorý na základe známych vzorov útokov dokáže detegovať útoky hrubou silou na služby SSH, Telnet, RDP. Hlavnou výhodou modulu oproti modulu HostStatsNemea je schopnosť detegovať aj veľmi pomalé útoky. Princíp metódy spočíva v tom, že si modul ukladá posledných N záznamov o IP tokoch a agreguje ich podľa zdrojovej IP. Tieto záznamy má uložené po určitú dobu, pričom pri každom ďalšom príchodzom zázname pre danú zdrojovú IP adresu sa tento agregát porovnáva voči preddefinovanému vzoru. Tento vzor určuje napr. aké TCP príznaky musia byť vyplnené, počet paketov v rámci IP toku alebo počet bajtov v rámci IP toku. V prípade že kontrolovaný záznam odpovedá vzoru, tak sa nahlási udalosť ako útok hrubou silou na konkrétny protokol.

- SIP Brute-Force Detector - Detekčný modul, ktorý na základe známych vzorov deteguje útoky hrubou silou na službu SIP. Analyzuje SIP odpovede a je schopný detegovať distribuované útoky. Detekcia spočíva v tom, že modul kontroluje stavový kód odpovedí SIP serverov a počíta počet "401 Unauthorizedödpovedí. Akonáhle počet týchto udalostí prekročí zadanú hranicu pre konkrétne užívateľské meno, tak modul nahlási túto udalosť ako útok hrubou silou.

- Vportscan detector - Detekčný modul, ktorý je špeciálne zameraný na detekciu vertikálneho skenovania portov. Detekčný algoritmus je založený na analýze počtu rôznych cieľových portov pre každý pár zdrojovej a cieľovej IP adresy. Vstupom sú všetky prichádzajúce záznamy o IP tokoch, ktoré spĺňajú podmienky ako napr. počet paketov v IP toku je menší ako 4, transportný protokol je TCP a TCP príznaky musia obsahovať iba príznak SYN. Takéto záznamy sa agregujú v zozname podľa páru vytvoreného zo zdrojovej a cieľovej IP adresy. Hodnoty ktoré sa ukladajú do zoznamu sú cieľové porty. Modul priebežne sleduje aký je počet unikátnych cieľových portov pre každý pár IP adries v zozname a ak tento počet prekročí 50, tak modul nahlási tento incident ako vertikálne skenovanie. Útočníkom je zdrojová adresa z páru a obeťou je cieľová adresa z páru.

Kapitola 5

Návrh detekčnej metódy

V kapitole 2 som písal, že komunikácia miner klientov s pool serverom, ktorú sa budem snažiť detegovať je typu klient-server. V kapitole 4 som opísal niekoľko detekčných modulov systému Nemea. Všetky tieto moduly detegujú anomálie v sieťovej prevádzke na základe rôznych vzorov práve pre klient-server služby. Na podobnom princípe navrhnem svoj spôsob detekcie miner klientov, ktorý sa účastnia v ťažení kryptomien. To znamená, že najskôr získam dáta ktoré budu obsahovať komunikácie miner klientov s pool serverom, z týchto dát vytvorím vzor komunikácie a následne navrhnem spôsob akým sa tento vzor bude vyhľadávať v kontrolovaných dátach. Postup popíšem v následujúcich sekciách.

5.1 Analýza dát sieťovej prevádzky

V tejto sekcii popíšem ako som získal dáta o IP tokoch komunikácie miner klientov a serverov. Ďalej z týchto dát vyberiem vlastnosti, ktoré sú spoločné pre komunikácie rôznych miner klientov. Na základe týchto vlastností následne v kapitole 5 navrhnem detekčnú metódu pre detekciu týchto komunikácii.

5.1.1 Získanie dát

Dáta o IP tokoch komunikácií miner klientov a serverov som získal dvomi spôsobmi:

- Sledovaním komunikácie pri ťažení kryptomien na vlastnej stanici.
- Sledovaním komunikácie verejne známych miner serverov.

Prvý spôsob spočíval v registrácii u konkrétneho pool serveru [13] a následným využitím programu cpuminer [16], ktorý je možné využiť na ťaženie kryptomeny Bitcoin alebo Litecoin. Ako prvé som spustil program Wireshark [19], pomocou ktorého som odchytával pakety na sieťovom rozhraní svojho počítača. Následne som spustil program cpuminer a pripojil sa na mnou vybraný pool server. Program cpuminer počas celej svojej doby behu komunikoval s pool serverom. Túto komunikáciu som odchytil do súboru. Ďalším krokom bola konverzia odchytenej komunikácie do IP tokov. Na toto som využil program softflowd [17]. Výsledkom bol súbor obsahujúci IP toky odchytenej komunikácie. Program cpuminer som mal spustený približne jednu hodinu a ukážku z jeho komunikácie je možné vidieť v tabuľke 5.1.

Druhý spôsob využíval zoznam verejne známych miner serverov [15]. Z tohto zoznamu som navštívil každú webovú stránku pool serverov a vyhľadal na nej informácie o IP adrese a

porte na ktorom beží služba pool servera obsluhujúca miner klientov. Výsledkom bol zoznam IP adries a portov pool serverov, na základe ktorého som následne získal komunikácie týchto staníc zo sónd v CESNET sieti. Všetky obdržané komunikácie patria do približne štvorhodinového časového okna od 8:00 do 12:00. Ukážku komunikácií je možné vidieť v tabuľke 5.2.

Zdrojová IP:Zdrojový port	Cieľová IP:Cieľový port	TCP príznaky	Pakety	Byty
192.168.100.1:33368	52.36.117.185:3333	.A....	64	3328
192.168.100.1:33368	52.36.117.185:3333	.AP...	27	1545
192.168.100.1:33368	52.36.117.185:3333	.A...F	24	1248

Tabuľka 5.1: Ukážka komunikácií pri ťažení na vlastnej stanici

Zdrojová IP:Zdrojový port	Cieľová IP:Cieľový port	TCP príznaky	Pakety	Byty
158.194.60.108:49179	74.84.128.158:3333	.A....	1	40
158.194.60.108:49172	74.84.128.158:3333	.A....	1	40
147.175.66.80:44410	52.31.186.94:3333	.AP...	209	22668
194.160.28.150:41610	52.19.8.80:3333	.AP...	264	29208
147.175.66.80:45013	52.18.177.202:3333	.A....	7	364

Tabuľka 5.2: Ukážka komunikácií pri použití verejne známych pool serverov

5.1.2 Analýza dát a výber spoločných vlastností

Nad získanými dátami som vykonal manuálne analýzu dát. Snažil som sa nájsť spoločné vlastnosti rôznych komunikácií miner klientov s pool servermi. Z tejto analýzy som odvodil následujúce spoločné vlastnosti. Treba ale poznamenať, že tieto vlastnosti sú získane pozorovaním relatívne malej časti a teda nie je vylúčené, že existuje komunikácia miner klienta s pool serverom, ktorá bude mať trochu iné vlastnosti.

- Komunikácia býva dlhodobá, často jej dĺžka presahuje hodiny.
- Komunikácia býva rozdelená na väčší počet IP tokov. Zaujímavosťou je, že niektoré komunikácie klientov mávajú iba 1 paket v rámci IP toku a naopak, komunikácie iných klientov mávajú rádovo stovky paketov v rámci IP toku, teda počet paketov v rámci IP toku sa pohybuje vo veľkom rozmedzí.
- Zdrojový port komunikácie sa môže zmeniť, ale bol vždy vyšší ako cieľový port.
- Komunikácie obsahujú vždy TCP príznak ACK, niektoré obsahujú aj TCP príznak PUSH.
- Počet bajtov v rámci paketu býva v rozmedzí od 40 do 118.

5.2 Pasívny spôsob detekcie

Tento spôsob detekcie bude založený na zbieraniu dát o komunikáciach, počítaní štatistických údajov z týchto dát a následnom vyhodnotení týchto štatistických údajov. Navrhnem dva spôsoby vyhodnotení údajov, z ktorých jeden sa bude spoliehať na manuálnu analýzu

dát a druhý bude mať formu rozhodovacieho stromu, ktorý bude výsledkom strojového učenia.

Ako vyplynulo z analýzy dát, komunikácia miner klientov s pool serverom je dlhodobá, s malým dátovým tokom a hlavne je rozdelená do viac IP tokov. Detekciu preto navrhnem tak, že bude pracovať nad dlhším časovým oknom a teda nad viac ako jedným IP tokom. Prichádzajúce záznamy o IP tokoch sa teda budú agregovať do jedného agregovaného záznamu a to podľa trojice *zdrojová IP adresa, cieľová IP adresa, cieľový port*. Túto trojicu volím z toho dôvodu, aby bola docielená agregácia komunikácie jedného klienta s jedným serverom. Zdrojový port v nej nie je zahrnutý z toho dôvodu, pretože ako vyplynulo z analýzy, miner klient môže komunikovať s jedným pool serverom s rôznymi zdrojovými portami. Agregovaný záznam bude obsahovať následujúce položky:

- Počet IP tokov obsahujúcich iba TCP príznak ACK.
- Počet IP tokov obsahujúcich iba TCP príznaky ACK a PUSH.
- Počet všetkých ostatných IP tokov, teda takých ktoré neobsahujú ani TCP príznak ACK ani TCP príznaky ACK a PUSH.
- Počet IP tokov obsahujúcich minimálne TCP príznak SYN.
- Počet IP tokov obsahujúcich minimálne TCP príznak RST.
- Počet IP tokov obsahujúcich minimálne TCP príznak RST.
- Počet IP tokov, v ktorých bol zdrojový port vyšší ako cieľový port.
- Počet paketov vo všetkých IP tokoch.
- Počet bajtov vo všetkých IP tokoch.
- Časová značka prvého IP toku v tomto agregáte, teda prvý videný IP tok.
- Časová značka posledného IP toku v tomto agregáte, teda posledný videný IP tok.

Detekcia sa bude vykonávať až nad takýmito agregovanými záznamami, pričom bude prebiehať v určitých časových intervaloch, napr. 60 sekúnd. V rámci jedného časového intervalu sa pre každý agregovaný záznam vykonajú oba spôsoby pasívnej detekcie.

Prvou metódou, ktorá sa spolieha na manuálnu analýzu je výpočet skóre podobnosti. Toto skóre určuje, ako moc je agregovaný záznam podobný vzorovej komunikácii miner klienta s pool serverom. Vzorovú komunikáciu som odvodil z vlastností získaných v podsekcii 5.1.2. Výpočet skóre podobnosti sa bude vykonávať následujúcim spôsobom:

1. Nastavíme skóre na 0.
2. Ak súčet IP tokov obsahujúcich TCP príznak ACK a TCP príznaky ACK a PUSH tvorí 80% všetkej komunikácie, ku skóre pripočítame 0.2.
3. Ak počet paketov za minútu je v rozmedzí 8 až 30, ku skóre pripočítame 0.2.
4. Ak pomer bajtov voči paketom je v rozmedzí 50 až 130, ku skóre pripočítame 0.2.
5. Ak pomer IP tokov ktorých zdrojové porty sú vyššie ako cieľové porty voči celkovému počtu IP tokov je viac ako 90%, ku skóre pripočítame 0.2.

6. Ak doba trvania komunikácie bude dlhšia než 5 minút, ku skóre pripočítame 0.2.

Výsledné skóre bude určovať podobnosť kontrolovaného agregovaného záznamu voči vzorovej komunikácii. Čím vyššie toto skóre bude, tým podobnejší je záznam vzoru. Potom ak prekročí hodnota skóre určitú hranicu, napr. 80% tak túto komunikáciu prehlásim za komunikáciu miner klienta s pool serverom. Tento spôsob počítania podobnosti dovoľuje to, že kontrolovaný záznam nemusí spĺňať všetky podmienky a aj tak bude označený ako komunikácia miner klienta s pool serverom. Rozdiel v komunikáciach sme si mohli všimnúť v tabuľkách 5.1 a 5.2. Počítanie skóre a rozhodovanie až na základe hodnoty tohoto skóre je teda lepší prístup, ako keby sme mali trvať na splnení všetkých spomenutých podmienok.

Druhý spôsob je založený na strojovom učení. Táto metóda bude mať dve časti. V prvej časti získam sadu ohodnotených údajov o komunikáciach. To znamená sadu skladajúcu sa z agregovaných údajov, ktoré boli popísané vyššie, pričom každý záznam v tejto sade bude mať k sebe priradený identifikátor triedy do ktorej patrí. Trieda komunikácie môže byť buď *bežná komunikácia* alebo *miner komunikácia*.

Túto sadu ohodnotených komunikácií napr. vo formáte CSV[1] použijem pri tvorbe rozhodovacieho stromu pomocou nástroja Weka [9].

Pri tvorbe rozhodovacieho stromu je možné zvoliť z rady algoritmov, pričom každý sa ďalej dá doladiť parametrami pre optimálny výsledok. Výsledkom bude rozhodovací strom vo forme pseudo kódu a tiež v podobe grafickej reprezentácie.

Druhá časť tejto metódy spočíva v použití tohto rozhodovacieho stromu v pasívnej detekčnej metóde na detekciu komunikácie miner klientov s pool servermi.

5.3 Aktívny spôsob detekcie

Pretože pasívny spôsob detekcie sa zameriava na štatistické porovnávanie sledovanej komunikácie a vzorovej komunikácie, nemôžeme s istotou tvrdiť, že detekované komunikácie sú skutočne komunikácie miner klientov s pool serverom. Tým že nevieme s určitosťou povedať či sa jedná o komunikáciu miner klienta s pool serverom alebo nie, z toho vyplýva že nie je možné určiť mieru falošne pozitívnych detekcií. Z tohoto dôvodu navrhnem druhú metódu detekcie, ktorá dokáže s veľmi vysokou pravdepodobnosťou povedať či sa jedná o komunikáciu miner klienta s pool serverom alebo nie.

Aktívny spôsob detekcie spočíva v tom, že detekčný modul sa pokúsi priamo pripojiť k podozrivému serveru a odošle mu výzvu, v ktorej sa tvári že je miner klient a chce ťažiť. Výhodou takéhoto prístupu je, že s istotou vieme povedať, ktorý server je pool server a ktorý server nie je, pretože iba pool server na túto výzvu odpovie očakávaným spôsobom. Naopak nevýhodou je, že aktívny prístup vyžaduje od detekčného modulu pripojenie k internetu a tiež takáto forma kontroly je oveľa pomalšia ako pasívna metóda. Nie je preto možné týmto spôsobom kontrolovať veľké množstvo komunikácii ako tomu je pri pasívnej detekčnej metóde. Z tohoto dôvodu preto navrhujem, že aktívny spôsob detekcie sa bude vykonávať iba nad komunikáciami, ktoré pasívny spôsob detekcie označí za podozrivé, teda komunikácie miner klientov a pool serveru. Týmto spôsobom sa bude aktívny spôsob detekcie vykonávať nad menším počtom komunikácií a zároveň sa tým zaručí aj nízka miera falošne pozitívnych detekcí.

Samotný aktívny test prebieha v dvoch krokoch. V prvom kroku detekčný modul odošle výzvu na podozrivý server. Táto výzva môže vyzerať následovne:

[1] Comma Separated Values

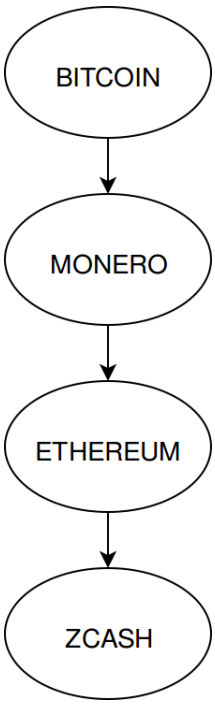
```
{"id": 1, "method": "mining.subscribe",
"params": ["cpuminer/2.4.3"]}
```

Detekčný modul po odoslaní tejto výzvy čaká na odpoveď servera. Ak server neodpovie v určitom časovom intervale prehlásim, že server nie je pool serverom. Ak server na odoslaný požiadavok odpovie, tak detekčný modul v tejto odpovedi vyhľadá špecifický reťazec znakov. Typická odpoveď od pool servera môže vyzerať následovne (jedná sa o skrátenú odpoveď, v ktorej reťazec '***' nahradzuje pre detekciu nepodstatné dáta):

```
{"error":null,"id":1,"result":[["mining.set\_difficulty","1"],
["mining.notify","1"]],"24690500837db5",4]}
{"params":[256],"id":null,"method":"mining.set\_difficulty"}
{"params":["f3799","3917dd2c463cad43***20000002","180375ff",
"5866c43e",true],"id":null,"method":"mining.notify"}
```

V tejto odpovedi detekčný modul vyhľadáva reťazec, napr. *mining.notify*. Ak detekčný modul tento reťazec nájde prehlásim, že server je pool serverom. Naopak ak modul reťazec nenájde, tak prehlásim že server nie je pool serverom.

Aktívny test musí byť schopný detegovať rôzne kryptomeny, pričom ale servery odlišných kryptomien sa správajú inak. Preto som navrhol postup akým sa bude podozrivý server postupne testovať. Tento postup popisuje obrázok 5.1. Podľa tohto obrázka je možné vidieť, že najskôr sa bude testovať server na dotazy kryptomeny Bitcoin. V prípade, že sa aktívnemu testu nepodarí pripojiť na daný server, aktívny test okamžite končí a nepokračuje sa ďalej. Ak server odpovie očakávanou odpoveďou, aktívny test vyhodnotí, že sa jedná o aktuálne testovanú kryptomenu a ukončí sa. V opačnom prípade, kedy server síce odpovie ale inou akou očakávanou odpoveďou, aktívny test pokračuje v poradí ďalšou kryptomenou.



Obr. 5.1: Diagram spôsobu akým detekčný modul testuje rôzne typy kryptomien

5.4 Vylepšenie detekčnej metódy

Spojením pasívneho a aktívneho spôsobu detekcie sa zníži réžia detekcie v rámci aktívnych spojení a tiež sa zníži miera falošne pozitívnych detekcií. Detekčnú metódu je ale ďalej možné vylepšiť takým spôsobom, že nebudem kontrolovať komunikáciu s jedným serverom viac ako jedenkrát. Takéto vylepšenie si môžem dovoliť v prípade ak som nejaký server ohodnotil, teda viem či je pool serverom alebo nie, a nevadí mi že každú ďalšiu komunikáciu s týmto serverom budem vyhodnocovať na základe predošlého výsledku aktívneho testu tohoto serveru. To teda znamená, že ak napríklad nejaký server vyhodnotím aktívnym testom ako pool server, všetky následujúce komunikácie s týmto serverom budú prehlásené za komunikáciu miner klienta s pool serverom ale už bez nutnosti prevádzať pasívny či aktívny spôsob detekcie.

Táto metóda bude využívať zoznam, do ktorého sa po každom aktívnom teste uloží IP adresa testovaného servera, vzdialený port servera a výsledok testu, teda či sa jedná o pool server alebo nie. Tento zoznam by bol na počiatku behu detekčného modulu prázdny a počas jeho behu by sa plnil výsledkami aktívnych testov. Pri každej kontrolovanej a tiež novej komunikácii sa najskôr overí či cieľová IP adresa nie je na zozname ohodnotených serverov. V prípade že tam IP adresa bude, zistím aký bol výsledok aktívneho testu pri tejto IP adrese, podľa tohoto výsledku určím či sa jedná o komunikáciu medzi miner klientom a pool serverom alebo nie. V prípade že tam táto IP adresa nebude, bude sa pokračovať klasicky pasívnym spôsobom detekcie a neskôr poprípade aj aktívnym testom. Je treba ale myslieť na to, že server na danej IP adrese môže po určitom čase zmeniť určenie svojich služieb. To znamená, že po určitom čase server, ktorý nebol pool serverom pre ťaženie kryptomien, sa môže stať takýmto pool serverom a tiež naopak, server ktorý tieto služby ponúkal ich prestal ponúkať. Je teda nutné, aby záznamy v tomto zozname obsahovali časové razítko vzniku a po určitom čase od svojho vzniku boli zo zoznamu odstránené.

Kapitola 6

Implementácia

V tejto kapitole popíšem implementáciu detekčného modulu pre systém Nemea, ktorý bude vykonávať navrhnutý spôsob detekcie ťaženia kryptomien. Detekčný modul a všetky jeho časti boli naimplementované v jazyku C++ s normou C++11.

6.1 Spôsoby ukladania dát

V tejto sekcii popíšem akým spôsobom sú uložené dáta v detekčnom module. Ide predovšetkým o spôsob ukladania IP adries, zoznam podozrivých komunikácií a tiež akým spôsobom je implementovaný zoznam, ktorý uchováva výsledky aktívneho testu.

IP adresa je uložená v špeciálnej štruktúre *ip_addr_t* prítomnej vo frameworku Nemea. Do tejto štruktúry je možné uložiť jak IP adresu verzie 4 tak IP adresu verzie 6. So štruktúrou sa tiež ľahko pracuje a preto ju využívam na ukladanie IP adries v následujúcich dátových štruktúrach.

Pretože z hľadiska nasadenia detekčného modulu môže byť počet rôznych podozrivých komunikácií veľmi veľký, rozhodol som sa implementovať tento zoznam podozrivých komunikácií ako hash tabuľku. Konkrétne ide o špeciálnu verziu hash tabuľky s názvom *fash_hash_table*, ktorej implementácia je prítomná vo frameworku Nemea. Hlavnou výhodou je, že veľkosť tabuľky, teda počet položiek ktoré je do nej možné uložiť, sa špecifikuje na začiatku a počas behu detekčného modulu sa nevykonávajú pre ňu žiadne alokácie pamäte. Nevýhodou na druhej strane je, že pri väčšom zaplnení sa pri pridávaní novej položky môže stať, že musíme nejakú položku z tabuľky odstrániť aby sme do nej mohli pridať novú. Špecifickou vlastnosťou tejto tabuľky je tzv. skrýš, angl. *stash*. Stash je malý úložný priestor, do ktorého sa ukladajú záznamy, ktoré boli z tabuľky odstránené. Pri vyhľadávaní sa záznam najskôr hľadá v tabuľke a následne je prehľadávaná stash. Kľúčom do tejto tabuľky je trojica, ktorá má nasledujúce prvky:

- Zdrojová IP adresa podozrivej komunikácie uložená v štruktúre *ip_addr_t*.
- Cieľová IP adresa podozrivej komunikácie uložená v štruktúre *ip_addr_t*.
- Cieľový port podozrivej komunikácie uložený v premennej typu *uint16_t*.

Jednotlivé prvky kľúča sú uložené v štruktúre, ktorá celá slúži ako kľúč do tabuľky.

Aby bolo možné vykonávať detekcie podľa navrhnutej metódy, musí každý záznam (agregované údaje o všetkých komunikáciach, ktoré zdieľajú rovnaký kľúč) obsahovať následujúce položky s príslušnými dátovými typmi:

- bool flagged - Označenie či daný záznam je komunikácia miner klienta s pool serverom.
- uint8_t pool_id - V prípade, že daný záznam je komunikácia miner klienta s pool serverom, táto položka obsahuje identifikačné číslo pool serveru.
- uint64_t ack_flows - Čítač obsahujúci počet IP tokov, ktoré obsahovali iba TCP ACK príznak.
- uint64_t ackpush_flows - Čítač obsahujúci počet IP tokov, ktoré obsahovali iba TCP ACK a zároveň TCP PUSH príznak.
- uint64_t syn_flows - Čítač obsahujúci počet IP tokov, ktoré obsahovali TCP SYN príznak.
- uint64_t rst_flows - Čítač obsahujúci počet IP tokov, ktoré obsahovali TCP RST príznak.
- uint64_t fin_flows - Čítač obsahujúci počet IP tokov, ktoré obsahovali TCP FIN príznak.
- uint64_t other_flows - Čítač obsahujúci počet IP tokov, ktoré neobsahovali iba TCP ACK príznak alebo neobsahovali iba TCP ACK a zároveň TCP PUSH príznak.
- uint64_t req_flows - Čítač obsahujúci počet IP tokov, ktorých zdrojový port bol väčší ako cieľový port.
- uint64_t packets - Čítač obsahujúci sumu počtu paketov všetkých IP tokov.
- uint64_t bytes - Čítač obsahujúci sumu počtu bajtov všetkých IP tokov.
- uint32_t first_seen - Časová značka, obsahujúca čas príchodu prvého IP toku.
- uint32_t last_seen - Časová značka, obsahujúca čas príchodu posledného IP toku.
- uint32_t last_exported - Časová značka, obsahujúca čas posledného nahlásenia záznamu.

Zoznam obsahujúci výsledky aktívneho testu je implementovaný ako dva zoznamy. Je to hlavne z toho dôvodu, aby sa mohla špecifikovať rôzna veľkosť pre každý z týchto zoznamov, pretože predpokladám že miner komunikácií bude oveľa menej ako bežnej komunikácie. Ak teda aktívny test odhalí nejaký miner pool server, tak IP adresu tohto serveru uložím do zoznamu s názvom blacklist zoznam. V opačnom prípade, teda ak sa nejedná o miner pool server, uložím IP adresu serveru do zoznamu s názvom whitelist zoznam. Whitelist zoznam a aj blacklist zoznam je implementovaný pomocou *fast_hash_table*, rovnako ako zoznam podozrivých komunikácií. Veľkosťou ale tieto dve tabuľky môžu byť rádovo menšie ako zoznam podozrivých komunikácií, pretože sa do nich budú ukladať iba záznamy, ktoré prešli pasívnou detekciou a očakávam, že týchto záznamov bude relatívne málo. Kľúč do oboch tabuliek ale na rozdiel od zoznamu podozrivých komunikácií tvorí iba nasledujúca dvojica:

- IP adresa uložená v štruktúre *ip_addr_t*.
- Cieľový port uložený v premennej typu *uint16_t*.

Podobne ako pri zozname podozrivých komunikácií, sú aj v tomto prípade prvky kľúča uložené v štruktúre, ktorá celá slúži ako kľúč do tabuľky.

Pretože tieto zoznamy slúžia iba na zistenie, či sa daná IP adresa a port v nich nachádza alebo nie, položky týchto zoznamov obsahujú iba časovú značku kedy boli do zoznamu pridané. Podľa tejto časovej značky sa následne kontroluje, či je daný záznam platný alebo nie.

6.2 Štruktúra modulu

Detekčný modul sa skladá z troch hlavných častí, pričom každá táto časť beží ako samostatné vlákno.

Prvá časť sa venuje spracúvaniu prichádzajúcich UniRec záznamov. Pre každý prichádzajúci záznam sa zistí, či sa jedná o IP tok s transportným protokolom TCP. V prípade, že sa jedná o IP tok s transportným protokolom TCP, tak sa z tohto UniRec záznamu získajú všetky potrebné údaje, pre vytvorenie záznamu v zozname podozrivých komunikácií, ktoré som opísal vyššie. Tieto údaje sa následne uložia do zoznamu podozrivých komunikácií. Spôsob uloženia je taký, že pre každý údaj je v tomto zozname vytvorený čítač, ktorý obsahuje sumu hodnôt pre daný údaj naprieč všetkými komunikáciami s rovnakým kľúčom, ktoré detekčný modul spracoval (viď zoznam podozrivých komunikácií).

Z komunikácie sa tiež vytvára kľúč, ktorý ju jednoznačne identifikuje v zozname podozrivých komunikácií. V prípade, že kľúč aktuálne spracovávanej komunikácie sa už nachádza v zozname podozrivých komunikácií, tak sa záznam s týmto kľúčom v zozname podozrivých komunikácií aktualizuje údajmi z aktuálne spracovávaného UniRec záznamu.

Okrem toho, že sa v tejto časti vytvárajú nové záznamy do zoznamu podozrivých komunikácií a aktualizujú tie záznamy, ktoré sa tam už nachádzajú, tak sa tiež kontroluje, či sa cieľová IP adresa spracovávanej komunikácie už niekedy v minulosti kontrolovala aktívnym testom, teda či sa nachádza na blacklist alebo whitelist zozname. V prípade. že sa cieľová IP adresa komunikácie nachádza na whitelist zozname, tak sa táto komunikácia vôbec nespracováva. V prípade, že sa cieľová I adresa nachádza na blacklist zozname, tak sa táto komunikácia okamžite nahlási ako ťaženie kryptomeny.

Ako posledné má táto časť za úlohu aktualizovať vnútorný čas detekčného modulu, ktorý sa využíva pre aktívny časovač, neaktívny časovač a tiež pre kontrolu záznamu v blacklist a whitelist zozname. Čas sa aktualizuje na základe časovej značky, ktorú obsahujú spracovávané UniRec záznamy.

Druhá časť detekčného modulu je určená na postupné prechádzanie zoznamu podozrivých komunikácií. Zoznam podozrivých komunikácií sa prechádza periodicky po určitých časových intervaloch. Postup kontroly každého záznamu je nasledujúci. Cieľová IP adresa a cieľový port podozrivej komunikácie, ktorú záznam obsahuje sa kontroluje voči blacklist zoznamu. Ak sa cieľová IP adresa s konkrétnym cieľovým portom na blacklist zozname nachádza, táto komunikácia je označená ako komunikácia miner klienta s pool serverom. Cieľová IP adresa a cieľový port podozrivej komunikácie, ktorú záznam obsahuje sa kontroluje voči whitelist zoznamu. Ak sa cieľová IP adresa s konkrétnym cieľovým portom na whitelist zozname nachádza, táto komunikácia je odstránená zo zoznamu podozrivých komunikácií. V prípade, že sa cieľová IP adresa s cieľovým portom podozrivej komunikácie nenachádzala ani na whitelist zozname ani na blacklist zozname, je táto komunikácia testovaná pasívnym testom. Ak pasívny test rozhodne, že sa nejedná o miner komunikáciu, pokračuje sa spracovávaním ďalšieho záznamu v zozname podozrivých komunikácií. V prípade, že pasívny test rozhodne, že sa jedná o miner komunikáciu, tak sa vykoná aktívny

test. Detekčný modul sa pokúsi pripojiť na cieľovú IP adresu a cieľový port podozrivej komunikácie. Ak aktívny test rozhodne, že sa jedná o miner pool server, tak pridá tento server aj s portom do blacklist zoznamu a podozrivú komunikáciu, ktorá obsahuje tento server ako cieľovú IP adresu označí ako komunikáciu miner klienta s pool serverom. Ak ale aktívny test rozhodne, že sa nejedná o miner pool server, tak pridá tento server aj s portom do whitelist zoznamu, podozrivú komunikáciu odstráni zo zoznamu podozrivých komunikácií a pokračuje spracovávaním ďalšieho záznamu zo zoznamu podozrivých komunikácií. Ďalej sa záznam kontroluje či vypršal neaktívny časovač. To znamená, že sa zistí časová značka poslednej aktivity komunikácie s rovnakým kľúčom (kedy ju modul naposledy prijal na vstupe) a táto hodnota sa odčíta od aktuálneho vnútorného času detekčného modulu. Výsledkom je doba, po ktorú bola daná komunikácia neaktívna a táto doba sa kontroluje či prekročila určitú hranicu. Ak ju prekročila, komunikácia sa kontroluje či bola označená ako miner komunikácia. Ak bola takto označená, tak sa tento záznam odošle výstupným TRAP rozhraním. Nakoniec sa záznam odstráni zo zoznamu podozrivých komunikácií. Ako posledné sa záznam kontroluje či vypršal aktívny časovač. To znamená, že sa zistí časová značka prvej aktivity komunikácie s rovnakým kľúčom (kedy ju detekčný modul prvýkrát prijal na vstupe) a táto hodnota sa odčíta od aktuálneho vnútorného času detekčného modulu. Výsledkom je doba, po ktorú bola daná komunikácia aktívna a táto doba sa kontroluje či prekročila určitú hranicu. Ak ju prekročila, komunikácia sa kontroluje či bola označená ako miner komunikácia. Ak bola takto označená, tak sa tento záznam odošle výstupným TRAP rozhraním. Na rozdiel od neaktívneho časovača, sa záznam zo zoznamu podozrivých komunikácií neodstráni ale sa len časová značka prvej aktivity komunikácie nastaví na aktuálny vnútorný čas detekčného modulu. Takto sa docieli, že záznam bude sledovaný aj naďalej ale ku ďalšiemu nahláseniu dôjde až po vypršaní ďalšie aktívneho alebo neaktívneho časovaču.

Tretia časť detekčného modulu, vykonáva len jednu funkciu a tou je kontrola, či záznamy vo whitelist a blacklist zozname sú platné. Táto kontrola sa vykonáva periodicky, napr. každú hodinu. Prechádza sa postupne každý záznam oboch zoznamov a kontroluje sa, či hodnota časovej značky vzniku záznamu odčítaná od aktuálneho času je vyššia ako nejaký prah. V prípade, že tento prah je prekročený, je záznam zo zoznamu odstránení.

6.3 Vstupy a výstupy modulu

Vstupom a výstupom detekčného modulu je myslená UniRec šablóna. Vstupná UniRec šablóna obsahuje základný formát *COLLECTOR_FLOW*, ktorý obsahuje nasledujúce položky:

- SRC_IP - Zdrojová IP adresa. Môže obsahovať IPv4 alebo IPv6 adresu.
- DST_IP - Cieľová IP adresa. Môže obsahovať IPv4 alebo IPv6 adresu.
- SRC_PORT - Zdrojový port transportnej vrstvy (TCP/UDP).
- DST_PORT - Cieľový port transportnej vrstvy (TCP/UDP).
- PROTOCOL - Číslo protokolu transportnej vrstvy.
- TCP_FLAGS - V prípade že sa jedná o TCP tok, tak položka obsahuje TCP príznaky spojené pomocou bitového súčtu (OR).
- TIME_FIRST - Časová značku vzniku IP toku.

- TIME_LAST - Časová značka konca IP toku, teda jeho exportovanie.
- PACKETS - Počet paketov v rámci celého IP toku.
- BYTES - Počet bajtov v rámci celého IP toku.
- LINK_BIT_FIELD - Bitové pole identifikujúce exportér, ktorý daný záznam o IP toku exportoval. Využíva sa iba v CESNET sieti.
- DIR_BIT_FIELD - Položka udávajúca smer IP toku. Využíva sa iba v CESNET.
- TOS - Položka Type of Contents nachádzajúcu sa v hlavičke IP protokolu.
- TTL - Položka Time To Live nachádzajúcu sa v hlavičke IP protokolu.

Na výstupe detekčného modulu sa nachádza šablóna obsahujúca položky, ktoré opisujú detekovanú a nahlásenú udalosť. Táto šablóna má formát *TIME_FIRST,TIME_LAST, SRC_IP,DST_IP,DST_PORT,EVENT_SCALE*. Sémantika položiek v šablóne je nasledujúca:

- SRC_IP - IP adresa detekovaného ťažiara kryptomeny.
- DST_IP - IP adresa detekovaného pool servera.
- DST_PORT - Port detekovaného pool servera.
- TIME_FIRST - Časová značka prvého výskytu komunikácie.
- TIME_LAST - Časová značka posledného výskytu komunikácie
- EVENT_SCALE - Položka obsahuje intenzitu komunikácie, ktorá predstavuje počet detekcií za určitý časový interval.

6.4 Konfigurácia

Niektoré vlastnosti detekčného modulu je možné konfigurovať prostredníctvom konfiguračného súboru. Konfiguračný súbor ma formát XML, pričom jeho štruktúra je pevne daná a špecifikovaná detekčným modulom. Detekčný modul tento konfiguračný súbor načíta pri svojom spustení a nastaví konkrétne hodnoty. Popis prvkov, ktoré obsahuje konfiguračný súbor:

- blacklist_file - Názov súbor, z ktorého bude inicializovaný blacklist zoznam. V prípade, že sú k dispozícií informácie o miner pool serveroch ešte pred samotným spustením programu, je možné ich pridať do tohto súboru a detekčný modul bude na základe nich vykonávať detekciu. Detekčný modul očakáva, na každom riadku tohto súboru informácie o miner pool serveri vo formáte *IP Adresa:port*. V prípade, že túto možnosť využiť nechceme, je možné zadať buď názov prázdneho súboru alebo znak - (pomlčka), ktorý detekčnému modulu hovorí aby dané nastavenie nepoužíval.
- whitelist_file - Názov súbor, z ktorého bude inicializovaný whitelist zoznam. Ak vieme o nejakých serveroch, že nie sú miner pool servermi a chceme aby sa nad nimi nevykonávala detekcia, tak informácie o nich zahrnieme do tohto súboru a detekčný modul ich bude ignorovať. Detekčný modul očakáva, na každom riadku tohto súboru informácie

o serveri vo formáte *IP Adresa:port*. V prípade, že túto možnosť využiť nechceme, je možné zadať buď názov prázdneho súboru alebo znak - (pomlčka), ktorý detekčnému modulu hovorí aby dané nastavenie nepoužíval.

- store_blacklist_file - Názov súboru, do ktorého sa pri ukončovaní detekčného modulu zapíše blacklist zoznam. Túto možnosť môžeme využiť v prípade, že chceme uchovať zoznam detekovaných miner pool serverov, napr. pre opätovné použitie použitie neskôr. Ak túto možnosť využiť nechceme, tak zadáme znak - (pomlčka) čím detekčnému modulu hovoríme aby žiadne informácie neukladal.

- store_whitelist_file - Názov súboru, do ktorého sa pri ukončovaní detekčného modulu zapíše whitelist zoznam. Túto možnosť môžeme využiť v prípade, že chceme uchovať zoznam serverov, ktoré boli pasívnym testom označené ako falošne pozitívne. Ak túto možnosť využiť nechceme, tak zadáme znak - (pomlčka) čím detekčnému modulu hovoríme aby žiadne informácie o serveroch neukladal.

- conn_timeout - Čas v sekundách, po ktorý má detekčný modul čakať v rámci aktívneho testu na nadviazanie spojenia so serverom. Niektorým serverom trvá odpovedať na dotaz dlhšie, či už kvôli oneskoreniu spôsobeným vyťaženou sieťou alebo vyťažením samotného servera. Naopak čím je hodnota časovača vyššia, tým dlhšie sa bude čakať na servery, ktoré nemusia ani odpovedať a tým sa zvyšuje doba trvania jedného aktívneho testu. Je teda nutné zvoliť takú hodnotu, ktorá pokryje dobu na odpoveď väčšiny serverov a tiež zbytočne nebude brzdiť dobu trvania aktívneho testu.

- read_timeout - Čas v sekundách, po ktorý ma detekčný modul čakať v rámci aktívneho testu na odpoveď serveru. Konkrétne sa jedná o dobu, kedy už bolo nadviazané spojenie, detekčný modul poslal na server dotaz a čaká na odpoveď. Rovnako ako pri možnosti conn_timeout aj tu je nutné zvážiť ako dlho sa má čakať aby aktívny test netrval zbytočne moc dlho.

- timeout_active - Čas v sekundách, po ktorý sa budú uchovávať dáta o aktívnej podozrivej komunikácií v zozname podozrivých komunikácií. Záznamy v zozname podozrivých komunikácií sa periodicky kontrolujú, pričom ak pri tejto kontrole sa zistí, že doba od prvej aktivity podozrivej komunikácie je väčšia ako tento timeout, záznam je zo zoznamu odstránený.

- timeout_inactive - Čas v sekundách, po ktorý sa budú uchovávať dáta o neaktívnej podozrivej komunikácií v zozname podozrivých komunikácií. Záznamy v zozname podozrivých komunikácií sa periodicky kontrolujú, pričom ak pri tejto kontrole sa zistí, že doba od poslednej aktivity podozrivej komunikácie je väčšia ako tento timeout, záznam je zo zoznamu odstránený.

- check_period - Čas v sekundách, ktorý ma detekčný modul čakať medzi jednotlivými periodickými kontrolami.

- stratum_check - Určuje či sa má po úspešnom pasívnom teste vykonať aktívny test. Môže nadobúdať hodnoty *true* (aktívny test sa bude vykonávať) a *false* (aktívny test sa nebude vykonávať).

- score_treshold - Minimálne skóre, ktoré musí byť prekročené aby sa prehlásilo, že podozrivá komunikácia kontrolovaná pasívnym testom je miner komunikácia.

- suspect_db_size - Maximálny možný počet podozrivých komunikácií, ktorý môže byť uložený v zozname podozrivých komunikácií. Jedná sa teda o veľkosť zoznamu podozrivých komunikácií, pričom jednotkou je podozrivá komunikácia.
- suspect_db_stash_size - Počet podozrivých komunikácia, ktoré je možné uložiť do stash úložiska zoznamu podozrivých komunikácií.
- blacklist_db_size - Maximálny možný počet miner pool serverov, ktorý môže byť uložený v blacklist zozname. Jedná sa teda o veľkosť blacklist zoznamu, pričom jednotkou je informácia o miner pool servery.
- blacklist_db_stash_size - Počet miner pool serverov, ktoré je možné uložiť do stash úložiska blacklist zoznamu.
- whitelist_db_size - Maximálny možný počet falošne označených serverov, ktorý môže byť uložený vo whitelist zozname. Jedná sa teda o veľkosť whitelist zoznamu, pričom jednotkou je informácia o servery.
- whitelist_db_stash_size - Počet falošne označených serverov, ktoré je možné uložiť do stash úložiska whitelist zoznamu.

6.5 Detekčné metódy

Každá časť pasívnej detekčnej metódy, teda detekcia na základe skóre podobnosti a detekcia na základe rozhodovacieho stromu, sú implementované ako samostatné funkcie. Algoritmus detekcie na základe skóre podobnosti bol implementovaný podľa návrhu, viď sekcia 5.2. Algoritmus detekcie na základe rozhodovacieho stromu, bol implementovaný ako množstvo jednoduchých podmienok. Cieľom bolo vytvoriť implementáciu, ktorá nie je moc náročná na výpočet, aby ju bolo možné použiť aj pri vyhodnocovaní na vysokorýchlostných sieťach.

Aktívny test je implementovaný pomocou štandardných soketov. Ako vyplynulo z návrhu, aktívny test by mal byť schopný detegovať pool server rôznych kryptomien. Implementácia aktívneho testu toto rieši takým spôsobom, že pre každú podporovanú kryptomenu existuje dotaz, ktorý detekčný modul posiela pool serveru a tiež regulárny výraz, ktorý popisuje časť odpovede pool serveru, podľa ktorej je možná detekcia. V nasledujúcich riadkoch popíšem dotazy a odpovede pre jednotlivé kryptomeny. Každý dotaz končí znakom *\x0a*, ktorý predstavuje znak s ASCII hodnotou 10 a ktorý značí pool serveru ukončenie dotazu.

Kryptomena Bitcoin (BTC) je najrozšírenejšia a najpoužívanejšia a preto sa pomocou aktívneho testu kontroluje ako prvá. Tento dotaz je platný aj pre kryptomenu Litecoin (LTC) a vyzerá následovne:

```
{"id": 1, "method": "mining.subscribe",
 "params": ["cpuminer/2.4.3"]}\x0a
```

Regulárny výraz, ktorý sa vyhľadáva v odpovedi pool servera je nasledovný:

```
minig.notify
```

Ako ďalšia sa kontroluje kryptomena Monero (XMR). Dotaz pre túto kryptomenu vyzerá následovne:

```
{"method": "login", "params": {"login": "42",
 "pass": "x", "agent": "xmr/1.0"}, "id": 1}\x0a
```

Regulárny výraz, ktorý sa vyhľadáva v odpovedi pool servera je nasledovný:

```
blob.*job_id.*target
```

Tretia v poradí sa kontroluje kryptomena Ethereum (ETH), pre ktorú dotaz vyzerá následovne:

```
{"worker": "eth1.0", "jsonrpc": "2.0",
 "params": ["0x42/k.work1/email@mail", "x"],
 "id": 2, "method": "eth_submitLogin"}\x0a
```

Regulárny výraz, ktorý sa vyhľadáva v odpovedi pool servera je nasledovný:

```
jsonrpc.*result":[ \t]*true
```

Ako posledná sa kontroluje kryptomena ZCash (ZEC), pre ktorú dotaz vyzerá následovne:

```
{"id": 1, "method": "mining.subscribe", "params":
 ["equihashminer", null, "zec", "6666"]}\x0a
```

Regulárny výraz, ktorý sa vyhľadáva v odpovedi pool servera je nasledovný:

```
mining.set_target
```

Jednotlivé dotazy pre pool servery nemusia nutne byť validné iba pre konkrétnu kryptomenu ale môžu byť validné aj pre iné kryptomeny. Príkladom je prvý dotaz, ktorý je platný jak pre Bitcoin tak pre Litecoin kryptomenu. Je teda možné, že okrem explicitne vymenovaných kryptomien, môže byť aktívnym testom detekované aj iné kryptomeny. Na druhej strane ale budú tieto ostatné kryptomeny falošne označené ako kryptomeny, ktorých dotaz ich detegoval. Táto vlastnosť ale nie je až taká negatívna, pokiaľ nám ide iba o detekciu pool servera.

Kapitola 7

Vyhodnotenie

V tejto kapitole predložím a zhodnotím namerané výsledky implementovaného detekčného modulu. Zhodnotím tiež výsledky rôznych rozhodovacích stromov vytvorených pre pasívnu detekciu, pričom tieto výsledky porovnám oproti pasívnej detekcii s využitím skóre podobnosti. Všetky experimenty prebiehali nad živými dátami niekoľkých rôzne veľkých sietí.

7.1 Rozhodovacie stromy

Pomocou nástroja Weka som vytvoril niekoľko rozhodovacích stromov ako kandidátov na pasívnu detekciu. Porovnal som ich voči sebe vzhľadom na obecnosť a úspešnosť vyhodnotenia. Pretože grafické zobrazenie rozhodovacích stromov je príliš zložité a pre účely vyhodnotenia nepodstatné, nebudem ich tu zobrazovať.

Prvý kandidát je rozhodovací strom vytvorený algoritmom J48. Tento rozhodovací strom vznikol na základe sady, ktorá obsahovala 273 miner komunikácií a 5000 bežných komunikácií. Výhodou tohto stromu je jeho obecnosť. Tá vychádza z toho, že tento strom má relatívne málo úrovní, len 10. Spôsob vyhodnotenia úspešnosti rozhodovacieho stromu zobrazuje *confusion matrix*. Ide o maticu, ktorá zobrazuje ako dopadla klasifikácia pre jednotlivé triedy, teda počet správne ohodnotených a počet nesprávne ohodnotených. Toto ohodnotenie je možné vidieť v tabuľke 7.1.

a	b	<– klasifikované ako
4977	23	a = bežná komunikácia
15	258	b = miner komunikácia

Tabuľka 7.1: Confusion matrix prvého rozhodovacieho stromu

Druhým kandidátom je rozhodovací strom vytvorený algoritmom REPTree. Tento rozhodovací strom vznikol na základe sady, ktorá obsahovala 273 miner komunikácií a 356574 bežných komunikácií. Výhodou tohto algoritmu je, že je možne špecifikovať maximálnu hĺbku stromu. Týmto sa dá určovať miera obecnosti výsledného rozhodovacieho stromu. Skúšal som rôzne hĺbky stromu a sledoval jeho úspešnosť. Za najvhodnejšiu hĺbku som uznal 10, pretože predstavuje akýsi stred medzi obecnosťou a úspešnosťou v klasifikácii miner komunikácií. Okrem tohto nastavenia, som pred samotnou tvorbou rozhodovacieho stromu použil filter *ClassBalancer*, ktorý upravil váhu jednotlivým triedam aby sa zmiernil dopad veľkého rozdielu v počte komunikácií v jednotlivých triedach. Confusion matrix pre tento rozhodovací strom je možné vidieť v tabuľke 7.2.

a	b	<– klasifikované ako
172301	6123	a = bežná komunikácia
4575	173849	b = miner komunikácia

Tabuľka 7.2: Confusion matrix druhého rozhodovacieho stromu

Tretí, posledný kandidát je rozhodovací strom vytvorený opäť algoritmom J48. Tentokrát ale rozhodovací strom vznikol na základe rovnakej sady ako druhý kandidát. Konkrétne obsahovala 273 miner komunikácií a 356574 bežných komunikácií. Výsledný strom, môže trpieť na preučenie následkom veľkej dátovej sady, no pretože bol schopný detekovať správne všetky miner komunikácie v testovacej sade, tak je vhodný na pasívnu detekciu. Confusion matrix tohto rozhodovacieho stromu je možné vidieť v tabuľke 7.3. Pretože počet komunikácií v jednotlivých triedach bol značne rozdielny, bolo nutné na dáta použiť filter *ClassBalancer*, podobne ako tomu bolo pri druhom kandidátovi.

a	b	<– klasifikované ako
177957	466	a = bežná komunikácia
0	178423	b = miner komunikácia

Tabuľka 7.3: Confusion matrix tretieho rozhodovacieho stromu

Pretože úlohou pasívnej detekcie je detegovať všetky miner komunikácie aj za cenu falošne pozitívnych detekcií, za najvhodnejší rozhodovací strom považujem tretieho kandidáta, ktorý síce kvôli svojej menšej obecnosti bude detekovať viac falošne pozitívnych detekcií no na druhej strane deteguje všetky miner komunikácie. O vyfiltrovanie falošne pozitívnych detekcií sa následne postará aktívny test.

7.2 Detekované udalosti

Následujú výsledky a zhodnotenie detekovaných udalostí v rámci rôznych sietí. Ako vyplynulo z predošlej sekcie o rozhodovacích stromoch, tretí kandidát je najvhodnejší pre pasívnu detekciu a práve pre tento rozhodovací strom sú následujúce výsledky.

Prvá sieť na ktorej sa detekčný modul testoval je relatívne malou sieťou s objemom dátovej prevádzky približne 4000 IP tokov za jednu sekundu. Detekčný modul na tejto sieti bol spustený niekoľko dní. Výsledky z tohto obdobia je možné vidieť v tabuľke 7.4.

Záznamy v zozname podozrivých komunikácií	133569719
Skontrolovaných podozrivých komunikácií pasívnou detekciou	532966185
Komunikácií ktoré boli označené algoritmom so skóre podobnosti	8267176
Komunikácií ktoré boli označené rozhodovacím strome	134296951
Komunikácií ktoré boli označené obomi pasívnymi detekciami zároveň	19134
Komunikácií ktoré boli označené aktívnom teste	8
Všetkých detekovaných miner komunikácií	3552

Tabuľka 7.4: Výsledky detekcie v prvej sieti

Ako je možné vidieť z tabuľky 7.4, zoznam podozrivých komunikácií obsahuje menej komunikácií než je počet skontrolovaných. To je preto, pretože niektoré komunikácie mohli byť skontrolované pasívnou metódou viac krát. Ďalej v tejto tabuľke môžeme vidieť, že

pasívna detekcia založená na skóre podobnosti dokázala relatívne dobre zredukovať počet podozrivých komunikácií. Naproti tomu pasívna detekcia založená na rozhodovacom strome bola menej efektívna. Toto môže byť spôsobené práve preučením alebo odlišným typom dát, než tých na základe ktorých bol tento rozhodovací strom vytvorený. Zaujímavým javom je, že prienik oboch pasívnych metód je značne malý. Kedže obe metódy boli vytvorené tak, aby detegovali všetky vzorové komunikácie miner klientov s pool serverom aj za cenu falošne pozitívnych detekcií, môžem usúdiť, že je veľmi pravdepodobné, že práve v prieniku týchto metód sa nachádzajú komunikácie miner klientov s pool serverom. Komunikácií v tomto prieniku je už naozaj málo (vzhľadom na to, že toto je údaj z niekoľkých dní), čo veľmi odľahčuje aktívnu detekciu. Ako vyplýva z tabuľky, na tejto sieti bolo detekovaných 8 ťažiarov kryptomien v rámci celého merania.

Druhá sieť na ktorej bol detekčný modul testovaný má priemerný počet IP tokov za sekundu približne 4500. Ako v prípade prvej siete aj tu bol detekčný modul spustený niekoľko dní, približne rovnako ako pri prvej sieti. Výsledky z tohto obdobia sú zobrazené v tabuľke 7.5.

Záznamy v zozname podozrivých komunikácií	144019771
Skontrolovaných podozrivých komunikácií pasívnou detekciou	534733277
Komunikácií ktoré boli označené algoritmom so skóre podobnosti	8273099
Komunikácií ktoré boli označené rozhodovacím strome	130878099
Komunikácií ktoré boli označené obomi pasívnymi detekciami zároveň	20735
Komunikácií ktoré boli označené aktívnom teste	16
Všetkých detekovaných miner komunikácií	13010

Tabuľka 7.5: Výsledky detekcie v druhej sieti

Z výsledkov pre túto sieť môžeme vidieť, že sa podobá prvej sieti a to nielen počtom skontrolovaných komunikácií ale aj ich charakteristikou, z ktorej vyplynuli veľmi podobné výsledky pre pasívnu detekciu. V tejto sieti ale detekčný modul pomocou aktívneho testu detegoval až 16 unikátnych miner pool serverov.

Tretia sieť na ktorej bol spustený detekčný modul má priemerný počet IP tokov za sekundu približne 5000. Doba trvania behu detekčného modulu na tejto sieti je pár dní ako u predchádzajúcich sietí. Výsledky pre túto sieť sú zobrazené v tabuľke 7.6.

Záznamov v zozname podozrivých komunikácií	151119305
Skontrolovaných podozrivých komunikácií pasívnou detekciou	506879571
Komunikácií ktoré boli označené algoritmom so skóre podobnosti	6831732
Komunikácií ktoré boli označené rozhodovacím strome	119424538
Komunikácií ktoré boli označené obomi pasívnymi detekciami zároveň	25266
Komunikácií ktoré boli označené aktívnom teste	4
Všetkých detekovaných miner komunikácií	2226

Tabuľka 7.6: Výsledky detekcie v tretej sieti

Môžeme si všimnúť, že aj keď v tejto sieti je vyšší počet IP tokov za sekundu ako v predošlej sieti, počet detekovaných ťažiarov kryptomien je menší.

V poradí štvrtá sieť je o niečo menšia oproti ostatným, priemerný počet IP tokov za sekundu je približne 1000. Detekčný modul na tejto sieti bol spustený o pár dní dlhšie než na prechádzajúcich sieťach. Výsledky z tejto siete je možné vidieť v tabuľke 7.7.

Záznamy v zozname podozrivých komunikácií	19218740
Skontrolovaných podozrivých komunikácií pasívnou detekciou	63901676
Komunikácií ktoré boli označené algoritmom so skóre podobnosti	1142974
Komunikácií ktoré boli označené rozhodovacím strome	14282590
Komunikácií ktoré boli označené obomi pasívnymi detekciami zároveň	424
Komunikácií ktoré boli označené aktívnom teste	0
Všetkých detekovaných miner komunikácií	0

Tabuľka 7.7: Výsledky detekcie vo štvrtej sieti

Z výsledkov je vidieť, že sa naozaj jedná o menšiu sieť, čo dokazuje počet záznamov v zozname podozrivých komunikácií. Taktiež je vidieť menej komunikácií, ktoré boli označené pasívnou metódou. V tejto sieti bolo aktívnym testom skontrolovaných iba 424 podozrivých serverov no ani jeden z nich nebol miner pool server.

Predošlá sieť je síce menšia ako boli siete pred ňou, ale ak si vypočítame pomery medzi jednotlivými výsledkami v rámci jednej siete, tak zistíme, že tieto pomery sú si veľmi podobné. Z tohto môžeme odvodiť, približný počet detekcií pre pasívnu metódu na iných sieťach a tiež približný počet aktívnych testov, ktoré sa budú vykonávať. Počet aktívnych testov nám potom môže napovedať koľko spojení modul vytvorí. Pretože každý jeden aktívny test trvá istý čas, príliš veľké množstvo aktívnych testov nielenže vytvorí veľké množstvo spojení ale aj spomalí kontrolu všetkých ostatných podozrivých komunikácií. To je z toho dôvodu, že v rámci jednej iterácie detekčného algoritmu, ktorý má trvať približne 60 sekúnd, sa bude vykonávať veľa aktívnych testov a teda detekčný modul sa do ďalšej iterácie dostane až keď všetky tieto aktívne testy dokončí.

Následujúca a tiež posledná sieť na ktorej bol spustený detekčný modul, je príkladom vysokorýchlostnej siete. Priemerný počet IP tokov za sekundu v tejto sieti je približne 100000. V tejto sieti nastáva vyššie opísaný problém, teda počet podozrivých komunikácií je veľmi veľký, z čoho vyplýva veľký počet aktívnych testov a tie spomaľujú detekčný modul. V prípade tejto siete, bolo spomalenie tak veľké, že jedna iterácia detekčnej metódy trvala viac ako tri dni. Toto môže byť veľmi nežiadúce, ak požadujeme nahlasovať detekované udalosti v kratšom časovom intervale. Výsledky tejto siete sú zobrazené v tabuľke 7.8.

Záznamy v zozname podozrivých komunikácií	912477672
Skontrolovaných podozrivých komunikácií pasívnou detekciou	3615055037
Komunikácií ktoré boli označené algoritmom so skóre podobnosti	36773286
Komunikácií ktoré boli označené rozhodovacím strome	667259156
Komunikácií ktoré boli označené obomi pasívnymi detekciami zároveň	118677
Komunikácií ktoré boli označené aktívnom teste	33
Všetkých detekovaných miner komunikácií	7954

Tabuľka 7.8: Výsledky detekcie v prvej sieti

7.3 Zhodnotenie výsledkov

Ako bolo možné vidieť z výsledkov testovania, detekčný modul na sieťach s menšou dátovou prevádzkou pracuje podľa očakávania a je schopný detegovať ťaženie kryptomien. Problém nastáva pri behu na vysokorýchlostných sieťach s veľkým počtom IP tokov za sekundu. Tento problém vychádza zo spôsobu detekcie. Pasívna detekcia filtruje veľkú časť sieťovej

prevádzky no pri veľkom počte IP tokov sa aj tak nevyfiltruje veľa falošne pozitívnych komunikácií. Všetky tieto nevyfiltrované podozrivé komunikácie sa musia skontrolovať aktívnym testom. Pripojenie na podozrivý server trvá istý čas a teda, čím viac je podozrivých komunikácií tým dlhšie detekčnému modulu trvá jedna iterácia skrz zoznam podozrivých komunikácií. Toto má za následok, že detekcia miner komunikácií na väčšej sieti trvá dlhšie a vytvára tiež viac spojení. Treba ale brať do úvahy aj blacklist a whitelist zoznamy, ktoré po určitom čase odľahčia detekčnému modulu od vykonávania veľkého počtu aktívnych testov. Je teda možné, že detekčný modul sa dostane do fáze kedy aj na väčšej sieti bude jedna iterácia detekčnej metódy trvať rozumne krátky čas.

Kapitola 8

Záver

V úvode tejto práci som popísal, že kryptomeny sú digitálne meny využívajúce kryptografiu na rôzne úkony tak, aby bola zabezpečená napr. integrita alebo autenticita. Vysvetlil som rozdiel medzi konceptami tvorby mincí proof-of-work a proof-of-stake. Ďalej som naznačil, že práve koncept tvorby mincí proof-of-work môže byť potenciálne nežiadúci, ak proces ťaženia mincí prebieha na stanici bez vedomia majiteľa. Uviedol som tiež prehľad najpoužívanejších kryptomien, na ktoré sa ďalej mala zamerať detekcia. V ďalšej kapitole som vysvetlil čo je to IP tok, opísal som protokoly NetFlow, IPFIX a popísal ako sa využívajú pri monitorovaní sieťovej prevádzky. Ďalej som popísal čo je to exportér a kolektor v rámci problematiky monitorovania sieťovej prevádzky a popísal som rôzne monitorovacie architektúry. Následujúca kapitola sa zamerala na framework Nemea a jeho súčasti. Tento framework na základe vlastného formátu správ o IP toku s názvom UniRec, dokáže detegovať rôzne sieťové incidenty a tiež poskytuje vývojárske nástroje pre tvorbu nových modulov pre tento framework. Okrem popisu samotného frameworku Nemea som popísal aj detekčné moduly HostStatsNemea, BruteForceDetector, SIP Brute-Force Detector, Vportscan detector, ktoré všetky detegujú nežiadúcu komunikáciu v sieťovej prevádzke na základe jej vzoru. Tieto detekčné moduly slúžili ako inšpirácia pre navrhovaný detekčný modul. V prvej časti ďalšej kapitoly som opísal dva spôsoby akými som získal dáta o sieťovej prevádzke s komunikáciami miner klientov s pool servermi. Na základe analýzy týchto komunikácií boli navrhnuté dve detekčné metódy. Prvá detekčná metóda, nazývaná tiež pasívnou metódou, bola založená na analýze dát o IP tokoch a vyhľadávaní vzorov v týchto dátach. Rozdeľovala sa na dve metódy, kde jedna metóda bola založená na vypočítaní skóre podobnosti a druhá využívala strojové učenie na tvorbu rozhodovacích stromov. Druhá detekčná metóda, nazývaná tiež aktívnou metódou, spočívala v dotazovaní podozrivých serverov a následnom vyhľadávaní známych vzorov v odpovediach, ktoré dotazované servery poslali. Následujúca kapitola popisovala implementáciu v jazyku C++ s využitím frameworku Nemea. Popísal som spôsoby uloženia dát v module, vstupy a výstupy modulu a tiež jeho konfiguráciu. Na záver nasledovala kapitola obsahujúca zhodnotenie rôznych rozhodovacích stromov, ktoré som vytvoril pomocou nástroja Weka. Taktiež som v tejto kapitole zhodnotil detekované udalosti detekčného modulu z rôznych sietí.

Z detekovaných udalostí pri testovaní detekčného modulu vyplynulo, že detekční modul je schopný detegovať IP adresy, na ktorý prebieha ťaženie kryptomien. Je ale nutné dodať, že pri sieťach s veľmi vysokým počtom IP tokov za sekundu, detekčný modul kontroluje veľký počet IP adries a teda celkový čas detekcie je značne vyšší ako pri relatívne malých sieťach. Aj cez túto negatívnu vlastnosť, bude detekčný modul určite prínosom pre detekciu

ťaženia kryptomien vo firemných, školských alebo iných malých až stredne veľkých sieťach, kde je možná detekcia ťaženia kryptomien iba na základe informácií o IP tokoch.

Možné rozšírenie tejto práce by mohlo byť v zlepšení alebo navrhnutí a implementácií ďalších pasívnych detekčných metód, ktoré ešte viac znížia počet falošne detekovaných udalostí, čím sa následne zníži aj réžia aktívneho testu. Ďalším rozšírením by mohla byť úprava aktívneho testu aby bol schopný detegovať viac kryptomien.

Literatúra

[1] IP Flow Information Export (IPFIX) Entities - IANA [online]. 2007 [cit. 2017-01-05]. URL http://www.iana.org/assignments/ipfix/ipfix.xhtml

[2] NetFlow Services Solutions Guide - Cisco Systems [online]. 2007 [cit. 2017-01-05]. URL http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.html

[3] NetFlow - Wikipedia [online]. 2014 [cit. 2017-05-18]. URL http://en.wikipedia.org/wiki/NetFlow

[4] Nfdump [online]. 2014 [cit. 2017-05-19]. URL http://nfdump.sourceforge.net/

[5] SMI Network Management Private Enterprise Codes - IANA [online]. 2015 [cit. 2017-01-05]. URL http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers

[6] Cryptography-based Prefix-preserving Anonymization [online]. 2015 [cit. 2017-05-18]. URL http://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/

[7] Crypto-Currency Market Capitalizations [online]. 2017 [cit. 2017-01-05]. URL https://coinmarketcap.com/

[8] List of all traded alternative cryptocurrencies with blocks, difficulty, hashrate and marketcap [online]. 2017 [cit. 2017-01-05]. URL https://www.cryptocoincharts.info/coins/info

[9] Weka 3 - Data Mining with Open Source Machine Learning Software in Java [online]. 2017 [cit. 2017-05-17]. URL http://www.cs.waikato.ac.nz/ml/weka/

[10] FastBit: An Efficient Compressed Bitmap Index Technology [online]. 2017 [cit. 2017-05-18]. URL https://sdm.lbl.gov/fastbit

[11] Flowmon - Flowmon Networks [online]. 2017 [cit. 2017-05-18]. URL https://www.flowmon.com/cs/products/flowmon/probe

[12] IPFIXcol - CESNET [online]. 2017 [cit. 2017-05-18]. URL https://github.com/CESNET/ipfixcol/

[13] Antpool - The most advanced bitcoin mining pool on the planet [online]. 2017 [cit. 2017-05-19].
URL https://www.antpool.com/

[14] CESNET, zájmové sdružení právnických osob [online]. 2017 [cit. 2017-05-19].
URL https://www.cesnet.cz/

[15] Comparison of mining pools - Bitcoin Wiki [online]. 2017 [cit. 2017-05-19].
URL https://en.bitcoin.it/wiki/Comparison_of_mining_pools

[16] CPU miner for Litecoin and Bitcoin [online]. 2017 [cit. 2017-05-19].
URL https://github.com/pooler/cpuminer

[17] softflowd - fast software NetFlow probe [online]. 2017 [cit. 2017-05-19].
URL http://www.mindrot.org/projects/softflowd/

[18] Warden - CESNET [online]. 2017 [cit. 2017-05-19].
URL https://warden.cesnet.cz

[19] Wireshark [online]. 2017 [cit. 2017-05-19].
URL https://www.wireshark.org/

[20] Antonopoulos, A.: *Mastering Bitcoin*. O'Reilly Media, Incorporated, Apríl 2014, ISBN 9781449374044.

[21] Bartoš, V.; Žádnik, M.; Čejka, T.: Nemea: Framework for stream-wise analysis of network traffic. CESNET Technical report 9/2013, December 2013.

[22] Beigel, O.: Is Bitcoin Mining Profitable in 2017? [online]. 2016 [cit. 2017-01-05].
URL
https://99bitcoins.com/bitcoin-mining-profitable-beginners-explanation/

[23] Chokun, J.: Who Accepts Bitcoins As Payment? List of Companies, Stores, Shops [online]. 2016 [cit. 2017-01-05].
URL https://99bitcoins.com/who-accepts-bitcoins-payment-companies-stores-take-bitcoins/

[24] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, Október 2004.
URL http://www.ietf.org/rfc/rfc3954.txt

[25] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011, September 2013.
URL http://www.ietf.org/rfc/rfc7011.txt

[26] Davis, J.: The Crypto-Currency: Bitcoin and its mysterious inventor [online]. Október 2011 [cit. 2017-01-05].
URL http://www.newyorker.com/magazine/2011/10/10/the-crypto-currency

[27] Dwork, C.; Naor, M.: *Pricing via Processing or Combatting Junk Mail*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, ISBN 978-3-540-48071-6, s. 139–147, doi:10.1007/3-540-48071-4_10.
URL http://dx.doi.org/10.1007/3-540-48071-4_10

[28] Greenberg, A.: Crypto Currency [online]. Apríl 2011 [cit. 2017-01-05].
URL http://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html

[29] Jakobsson, M.; Juels, A.: *Proofs of Work and Bread Pudding Protocols(Extended Abstract)*. Boston, MA: Springer US, 1999, ISBN 978-0-387-35568-9, s. 258–272, doi:10.1007/978-0-387-35568-9_18.
URL http://dx.doi.org/10.1007/978-0-387-35568-9_18

[30] King, S.: PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake [online]. August 2012 [cit. 2017-01-05].
URL https://peercoin.net/assets/paper/peercoin-paper.pdf

[31] Tasca, P.: Digital Currencies: Principles, Trends, Opportunities, and Risks. *SSRN Electronic Journal*, September 2015, ISSN 1556-5068.
URL https://ssrn.com/abstract=2657598

[32] Wang, L.; Liu, Y.: *Exploring Miner Evolution in Bitcoin Network*. Cham: Springer International Publishing, 2015, ISBN 978-3-319-15509-8, s. 290–302, doi:10.1007/978-3-319-15509-8_22.
URL http://dx.doi.org/10.1007/978-3-319-15509-8_22

Príloha A

Obsah CD

- Zdrojový kód vyvíjaného Nemea modulu.
- Manuál pre sprevádzkovanie modulu.
- Elektronická verzia diplomovej práce vo formáte PDF.
- Zdrojové texty diplomovej práce pre systém LaTeX.