Univerzita Pavla Jozefa Šafárika v Košiciach

Prírodovedecká fakulta

DETEKCIA ÚTOKOV NA ZÁKLADE ŠTATISTICKEJ ANALÝZY SIEŤOVEJ PREVÁDZKY (NETFLOW)

DIPLOMOVÁ PRÁCA

Študijný odbor:	Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci záverečnej práce:	RNDr. Rastislav Krivoš-Belluš, PhD.
Konzultant:	Mgr. Lukáš Hlavička, RNDr. Martina Hančová, PhD.

Košice 2014

Bc. Ľubomír Nagajda

Poďakovanie

Rád by som poďakoval konzultantovi diplomovej práce Mgr. Lukášovi Hlavičkovi, PhD., RNDr. Martine Hančovej a vedúcemu práce RNDr. Rastislavovi Krivoš-Bellušovi PhD., za cenné pripomienky a za obetavosť počas tvorby mojej diplomovej práce.

Namiesto tejto strany vložte
zadanie z informačného systému
podpísané vedúcim ústavu!

Abstrakt

Pri exponenciálne rastúcom počte zariadení rastie aj počet hrozieb, preto sa bezpečnosť stáva prioritou. V minulosti bolo vytvorených mnoho metód na identifikáciu hrozieb v sieti, ale rastúce množstvo užívateľov prináša mnohé problémy súvisiace s rozsiahlými sieťami. V mojej práci som navrhol algoritmus na detekciu štatistického profilu (baseline) sieťovej prevádzky a detekciu útokov na základe štatistických odchýlok od tohto profilu. Dáta budu získavané NetFlow protokolom.

Abstract

When exponentially growing numbers of an increasing number of threats, therefore, safety becomes a priority. In the past it was created many methods to identify threats in the network, but a growing number of users can enjoy many of the problems associated with large-scale networks. In my work I have proposed an algorithm to detect statistical profile (baseline) network traffic and detect attacks based on statistical deviations from that profile. data will be obtained NetFlow protocol.

Obsah

Kapitola 1

Úvod

Pri exponenciálne rastúcom počte zariadení rastie aj počet hrozieb, preto sa bezpečnosť stáva prioritou. V minulosti bolo vytvorených mnoho metód na identifikáciu hrozieb v sieti, ale rastúce množstvo užívateľov prináša mnohé problémy súvisiace s rozsiahlými sieťami.

V rozsiahlej sieti je prakticky nemožné kontrolovať obsah každého paketu v sieti. Protokol NetFlow od spoločnosti Cisco Systems nám však umožnuje kontrolovať aj veľký prietok v reálnom čase. Množstvo dát vytvorených pre NetFlow je síce menšie ako čisté dáta v sieti, ale neustále spracovávanie a uchovávanie týchto dát je náročné na prostriedky a preto je rozumné vytvárať štruktúry a postupy, ktoré nepotrebujú uchovávanie všetkých dát, ale vedia popísať ich správanie. V našej práci sme teda navrhli algoritmy na detekciu štatistického profilu (baseline) sieťovej prevádzky, ktorý nám ponúka predstavu o tom, ako prevádzka vyzerá a detekciu útokov na základe štatistických odchýlok od tohto profilu. Pri vytváraní baseline budeme dbať na modularitu cieľov a teda na možnosť vytvoriť štatistický profil voči konkrétnym aspektom ako je IP, port, protokol, či dokonca kombinácie aspektov a známe útoky. Pre detekciu odchýlok dát od štatistického profilu navrhneme algoritmy s odlišnými metódami na detekciu odľahlých hodnôt, ktoré v tomto profile predstavujú potencionálne útoky. Dáta sú získavané práve protokolom NetFlow. Fungovanie protokolu, jeho výhody, analýzu problému, priblíženie existujúcich a návrh vlastného riešenia opisujeme v teoretickej časti v kapitole 1. V kapitole 2 sa zameriavame na výsledný program ktorý sme navrhli a naprogramovali v jazyku Java s využitím navrhnutých algoritmov. Kapitolu 4 sme venovali práve testovaniu riešenia a analýze výsledkov. Samotný zdrojový kód je dostupný na priloženom CD.

Kapitola 2

Teoretická časť

Táto kapitola obsahuje teoretické poznatky potrebné na zostavenie riešenia, jeho návrh a opis protokolu NetFlow. Mnohé špecifiká NetFlow sa v práci nenachádzajú, keďže spadajú mimo záber tejto diplomovej práce.

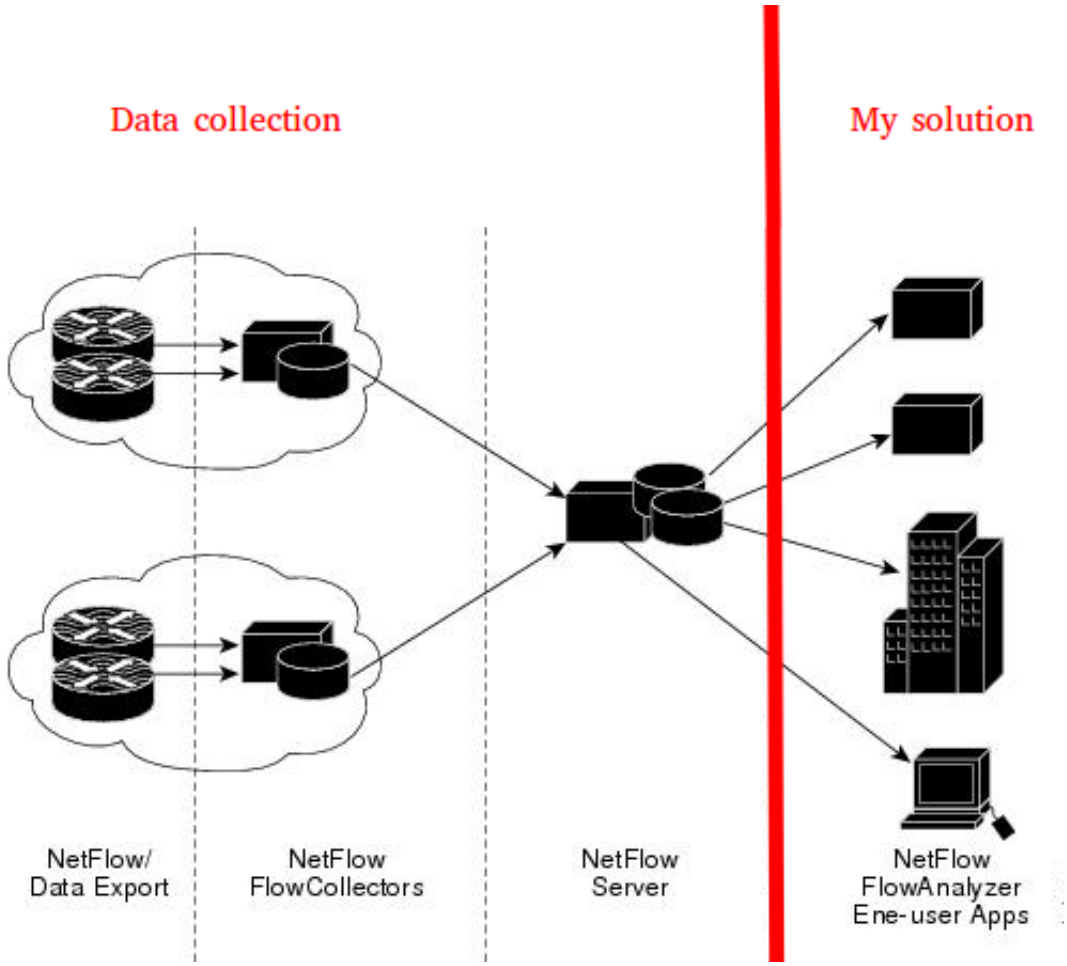
2.1 NetFlow

Existuje viacero formátov na vytváranie, prenos a uchovanie záznamov o tokoch. Tento formát definuje spôsob zberu, tvorbu, výmenu a uchovanie informácií o IP tokoch. Najznámejší (otvorený formát vyvíjaný firmou Cisco Systems) je NetFlow. Z NetFlow verzie v9 vychádza IETF standard Internet Protocol Flow Information eXport (IPFIX). Ďalším proprietárnym formátom je sFlow, ktorý využíva vzorkovanie (s ako sampling). V našej práci budeme využívať práve NetFlow.

2.1.1 IP Flow

Základ NetFlow technológie tvorí práve IP Flow (IP tok). Tok je v terminológii NetFlow definovaný ako sekvencia paketov so zhodnou päticou údajov:

- zdrojová IP adresa
- cieľová IP adresa
- zdrojový port pre TCP a UDP (hodnota 0 pre ostatné protokoly)
- cieľový port pre UDP a TCP, kód a type pre ICMP (0 pre ostatné protokoly)
- IP protokol



Obr. 2.1: Schéma zberu, ukladania a spracovania NetFlow dát.

2.1.2 Architektúra

Zvyčajná architektúra sa skladá z niekoľkých sond (probes) a jedného kolektora (collector). Ako sondy môžu slúžiť aj smerovače alebo zariadenia v sieti. Pre lepší výkon je však vhodné používať špeciálne pasívne sondy. Okrem rýchlosti spracovávania je ich výhodou hlavne to, že je ich možné zapojiť na ľubovoľné miesto v sieti (narozdiel od smerovačov).

Na obrázku 2.1 je zobrazená obvyklá architektúra [1], kde ako sonda slúži práve smerovač posielajúci dáta na kolektor, ktorý data ukladá na úložisko, odkiaľ su neskôr spracovávané. NetFlow záznamy sú zo sondy (probe) vysielané na kolektory pomocou UDP alebo SCTP protokolu. Keďže sondy po odoslaní záznamu informácie neuchovávajú, je možné, že sa záznam stratí.

[1] [6] NetFlow Services Solutions Guide. s.24. Dostupné z WWW: [http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.pdf]

2.1.3 Verzie NetFlow

NetFlow protokol vznikol v niekoľkých verziách. Prvá verzia v1 sa veľmi neuchytila a verzie v2-v4 ani neboli uvedené do prevádzky. Prvá používanejšia verzia v5 je ešte v dnešnej dobe najpoužívanejšou, aj keď sa vo veľkej miere začala používať verzia v9, na základe ktorej vznikol aj štandardizovaný protokol IPFIX. Verzia v9 priniesla celú radu nových a zaujímavých vecí. Došlo k zmene architektúry formátu, ktorý je označený flexibilný a rozšíriteľný, čo znamená, že záznamy tokov sú definované pomocou šablón (templates), ktoré sú rozšíriteľné o nové položky. Šablóny sú exportované spolu s NetFlow dátami na príslušný kolektor, aby bolo možné určiť formát dát. Taktiež NetFlow v9 priniesol podporu viacerých technológií: multicast, IPv6, Egress NetFlow, Multiprotocol Label Switching (MPLS) a Border Gateway Protocol (BGP) next hop. Medzitým ešte vyšli verzie v6, ktorá podporovala tunelovú prevádzku a v7, ktorá poskytovala aj informácie zo switchov.

2.2 Návrh riešenia

Táto kapitola obsahuje popis existujúcich riešení popisujúcich podobný problém, ako aj motiváciu a dôvod k postupom ktoré sme zvolili,základy štatistiky potrebné k porozumeniu navrhnutých algoritmov a algoritmy samotné. Finálne riešenie bude obsahovať naimplementované tri štatistické prístupy na odhalenie anomálií v sieti s možnostou ďalšieho rozširovania. Každému algoritmu je venovaná jedna podsekcia (2.4.3–2.4.5).

2.2.1 Existujúce riešenia a motivácia

Existujú mnohé nástroje pre monitorovanie siete založené na rôznych princípoch ako aj riešenia založené na protokole NetFlow. Pravdepodobne najvýhodnejšie je využívanie komerčných riešení, ktorým sme sa v tejto práci vôbec nevenovali, pretože nieje možné zistiť, ako fuguju a prispôsobiť si ich svojim potrebám. Takisto existuje niekoľko freeware riešení, ktoré su pre nás nevhodné pre rovnaké dôvody. Pri zisťovaní dostupných riešení sme však testovali aj riešenia ako NTop a NfSen, ktoré sú open-source a zároveň ponúkajú možnosť pridávania pluginov, čo nás pôvodne motivovalo práve k vývoju pluginu pre jeden z týchto programov. Pri hlbšom skúmaní sme si však uvedomili, že v prípade, že by sme chceli v produkčnom prostredí zmeniť collector alebo kontrolovať historické dáta, bol by s týmto problém, lebo tieto programy

majú závislosti na vlastných balíčkoch a zmena by bola príliš náročná. Preto sme sa rozhodli navrhnúť a naimplementovať vlastné riešenie, ktoré bude štatistickými metódami odhaľovať možnosť útokov s využitím protokolu NetFlow a zároveň bude možné akúkoľvek časť programu vymeniť (collector, databázu, či dokonca samotné algoritmy).

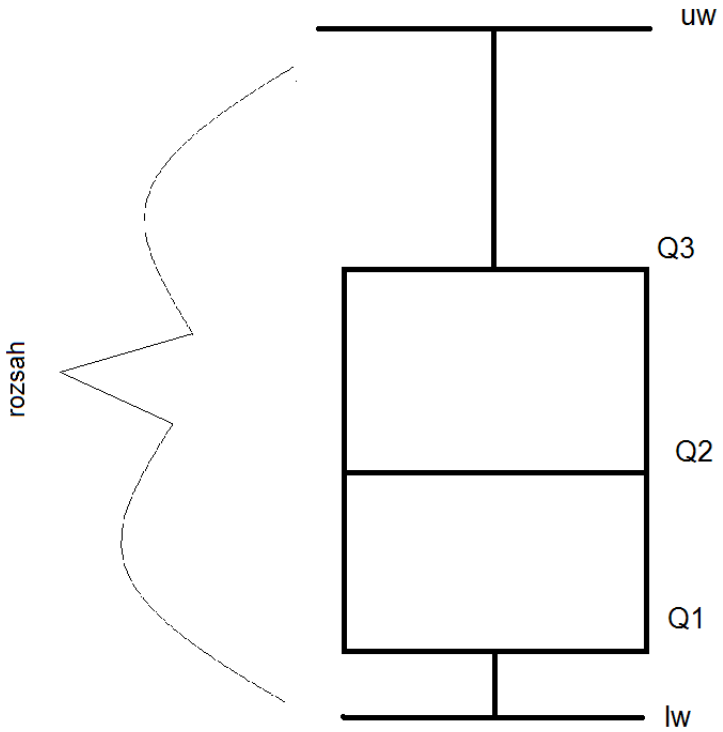
2.2.2 Základné štatistické pojmy

Zadefinujme si popisné štatistické pojmy[2], ktoré budeme využívať na identifikáciu anomálií v netflow dátach. Popisná (deskriptívna) štatistika sa používa na opis a zahŕňa jeden dátový súbor.

Pojmy ktoré budeme dalej používať:

- Medián–Prvok, ktorý oddeľuje hornú polovicu v usporiadanej vzorke od dolnej polovice. V prípade, že vzorka nemá jednoznačný centrálny prvok, sú dva centrálne prvky zrátané a vydelené dvoma, čo vedie k získaniu mediánu.
- Kvartil–Jedna z troch hodnôt, ktoré rozdeľujú vzorku na štyri rovnaké diely. Každý z nich predstavuje presne jednu štvrtinu vzorky. Druhý kvartil predstavuje medián Q_2. Využívať budeme hlavne prvý $Q_1(25\%)$ a tretí kvartil $Q_3(75\%)$.
- Odľahlé hodnoty (Anomálie)–Komponent, ktorý je vzdialený od zvyšku vzorky. Odľahlé hodnoty predstavujú odchýlky vo vzorke sieťovej prevádzky. V tejto práci sú odľahlé hodnoty chápané ako potencionálne útoky.
- Medzikvartilová vzdialenosť(IQR)–Vzdialenosť medzi prvým a tretím kvartilom
- Baseline–Pod pojmom baseline si budeme predstavovať štatistickú krivku opisujúcu bežnú komunikáciu na v sieti zostavenú z predchadzajúcich dát.
- Smerodajná odchýlka– Kvadratický priemer odchyliek hodnôt od ich aritmetického priemeru.

[2][1] GRAHAM J. G. UPON, IAN COOK: Understanding Statistics.



Obr. 2.2: Grafické zobrazenie (krabicový diagram).
uw-maximálna hodnota. Všetko nad uw prehlásime za odľahlú hodnotu.
ul-minimálna hodnota. Všetko pod ul prehlásime za odľahlú hodnotu.

Odľahlá hodnota v štatistike je hodnota, ktorá je vzdialená od ostatných pozorovaní. Takáto hodnota naznačuje nezvyčajné správanie, alebo chyby meraní. V našom prípade je odľahlá hodnota signál pre možný útok, preto naše algoritmy budú rôzne prístupy na odhalenie odľahlých hodnôt.

2.2.3 Metóda na odhalenie odľahlých hodnôt (anomálií) M_1

Zaveďme si prvý model na detekciu odľahlých hodnôt (potencionálnych útokov), v ktorom dáta budeme deliť pomocou piatich bodov: maximum, minimum, medián, 1. a 3. kvartil(Q_1 a Q_3). Tento model budeme ďalej označovat ako *model* M_1 Vzdialenosť medzi Q_3 a Q_1 nazývame medzikvartilová vzdialenosť (interquartile range–IQR). Zadefinujme minimálny bod ako A*IQR+C vzdialenosť od Q_1 a maximum ako A*IQR+c vzdialenosť od Q_3 . Konštanty A a C si volíme podľa toho, ako veľkú toleranciu chceme. Bežne používaná hodnota pre A je $1,50$. Všetky hodnoty, ktoré sú mimo intervalu medzi minimom a maximom nazveme anomáliou.

$IQR = Q_3 - Q_1$

$MIN = Q_3 - (A * IQR + C)$

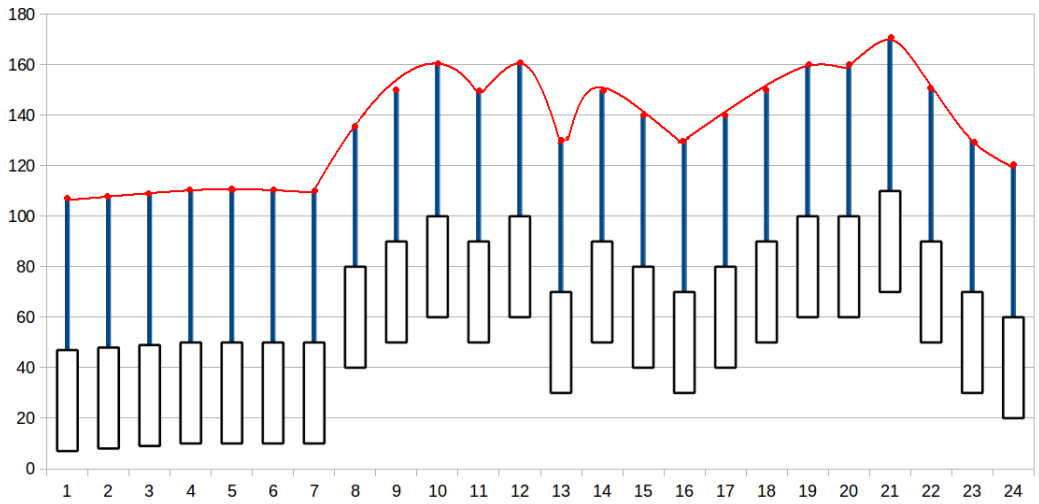
$MAX = Q_3 + (A * IQR + C)$

Po odstránení anomálie sa vzorka zmení, a preto je potrebné hodnoty odznova preratávať. Keďže minimá sú pre nás nezaujímavé a nenaznačujú žiaden útok, zameriame sa len na maximum.

Na odstránenie anomálií sme použili jednoduchý algoritmus:

Algorithm 1 Detekcia anomálií(A,C)

$B = getBaseline()$
$X = getTraffic()$
$Q_1 = prvyKvartil(B)$
$Q_3 = tretiKvartil(B)$
$MAX = Q_3 + A * (Q_3 - Q_1) + C$
for all $element : X$ **do**
  if $element > MAX$ **then**
    $WARLOG(element)$
    $X.remove(element)$
  end if
end for
$updateBaseline(B, X)$



Obr. 2.3: M1 Baseline - Grafické zobrazenie kvartilov pre celý deň. červená čiara - označuje hranicu, nad ktorou sú všetky hodnoty prehlásené za odľahlé

2.2.4 Metóda na odhalenie odľahlých hodnôt (anomálií) M_2

Zaveďme si druhý model na detekciu odľahlých hodnôt (potencionálnych útokov) inšpirovaný Grubssovým testom na odľahlé hodnoty. Tento model budeme dalej označovať ako *model* M_2 a výpočet odľahlých hodnôt budeme realizovať pomocou rovnice[3]

$G \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(N,N-2)}}{N-2+t^2_{a/(N,N-2)}}}$,

kde N je počet prvkov a $t^2_{a/(N,N-2)}$ zastupuje hornú hranicu t-rozdelenia s $N-2$ stupňami voľnosti a hladine významnosti α/N. Ak by sa jednalo o obojstranný test,tak N nahradíme 2N. Pre naše potreby, ale potrebujeme len maximálne hodnoty. Zadefinujme si teda G, výpočet odľahlej hodnoty ako

$G = \frac{Y_{max}-\overline{Y}}{S}$

kde Y_{max} je maximálna hodnota, $\overline{Y}$ je priemer a S je smerodajná odkýlka vyrátaná ako

$S^2 = \frac{1}{N-1} \sum (Y_i - \overline{Y})^2$

kde Y_i sú všetky hodnoty.

t-rozdelenie

Hodnoty pre t-rozdelenie berieme z tabuľky študentovho t-rozdelenia, vyrátanej komulatívnou distribučnou funkciou. T-rozdelenie je symetrecké a teda

$t_{1-\alpha,v} = -t_{\alpha,v}$

T tabuľka môže byť použitá pre jednostranné (dolný a horný) a obojstranné testy pre príslušnú hodnotu α.

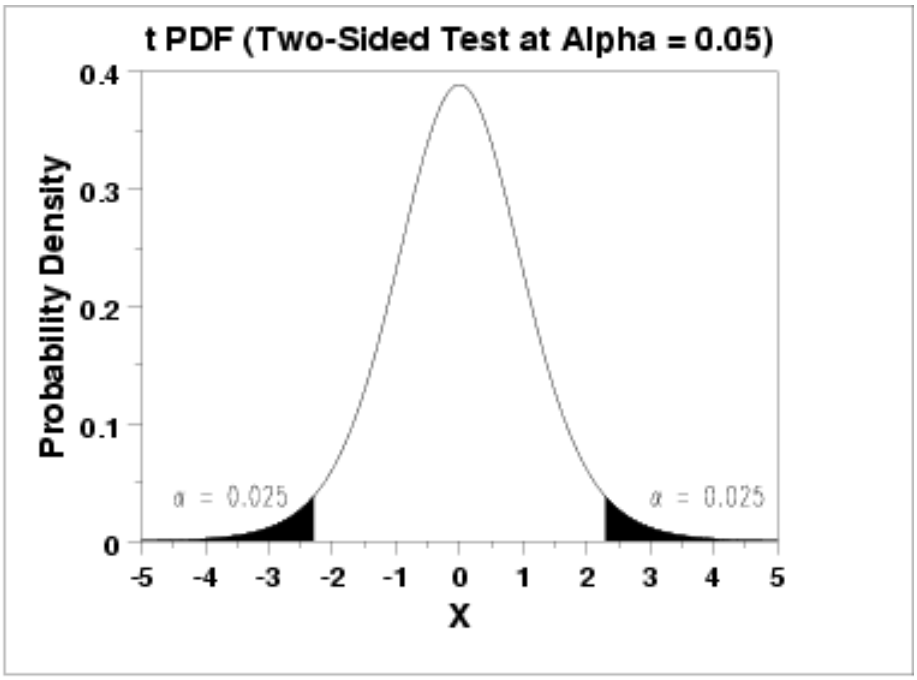
Hladina významnosti α je znázornená v nižšie uvedenom grafe, ktorý zobrazuje rozdelenie s 10 stupňami voľnosti. Najčastejšie sa používa hladina významnosti $\alpha = 0,05$. Pri obojstrannom teste určíme $1-\alpha/2$, alebo $1-0,05/2 = 0,975$, keď $\alpha = 0.05$. Keď je absolútna hodnota štatistického výsledku skúšky väčšia než kritická hodnota (0,975), potom odmietame nulovú hypotézu. Vzhľadom k symetrii rozdelenia t, sa sploštia iba pozitívne kritické hodnoty v nasledujúcej tabuľke .

[3]Vzorce a hodnoty su inšpirované zdrojom:

[7] ZEY CH., NIST/SEMATECH e-Handbook of Statistical Methods,

1.3.6.7.2. Critical Values of the Student's t Distribution,

Dostupný z WWW: [http://www.itl.nist.gov/div898/handbook/eda/section3/eda3672.htm]



Obr. 2.4: Obojstranný test pre $\alpha = 0.05$

Vzhľadom k tomu, zadaná hodnota pre α :

- Pri obojstrannom teste, je potrebné nájsť stĺpec zodpovedajúci $1 - \alpha/2$ a odmietnuť nulovú hypotézu, ak je absolútna hodnota štatistického výsledku skúšky väčšia než hodnota $t_{1-\alpha/2,N}$ v nižšie uvedenej tabuľke .
- Pre horný, jednostranný test je potrebné nájsť stĺpec zodpovedajúci $1 - \alpha$ a odmietnuť nulovú hypotézu, ak je štatistický výsledok skúšok väčší než hodnota tabuľky .
- Pre dolný, jednostranný test je potrebné nájsť stĺpec zodpovedajúci $1 - \alpha$ a zamietnuť nulovú hypotézu, ak je štatistický výsledok skúšok menší než záporné hodnoty tabuľky .

Nás však zaujíma len jednostranný (horný) test.

Tabuľka je potrebná len pre počet prvkov menší ako 100. Pre vyššie čísla predpokladáme normálne rozdelenie a zachovávame konštantnú hodnotu.

Tabuľka pre t-rozdelenie:

N	0.90	0.95	0.975	0.99	0.995	0.999
1.	3.078	6.314	12.706	31.821	63.657	318.313
2.	1.886	2.920	4.303	6.965	9.925	22.327
3.	1.638	2.353	3.182	4.541	5.841	10.215
4.	1.533	2.132	2.776	3.747	4.604	7.173
5.	1.476	2.015	2.571	3.365	4.032	5.893
6.	1.440	1.943	2.447	3.143	3.707	5.208
7.	1.415	1.895	2.365	2.998	3.499	4.782
8.	1.397	1.860	2.306	2.896	3.355	4.499
9.	1.383	1.833	2.262	2.821	3.250	4.296
10.	1.372	1.812	2.228	2.764	3.169	4.143
11.	1.363	1.796	2.201	2.718	3.106	4.024
12.	1.356	1.782	2.179	2.681	3.055	3.929
13.	1.350	1.771	2.160	2.650	3.012	3.852
14.	1.345	1.761	2.145	2.624	2.977	3.787
15.	1.341	1.753	2.131	2.602	2.947	3.733
16.	1.337	1.746	2.120	2.583	2.921	3.686
17.	1.333	1.740	2.110	2.567	2.898	3.646
18.	1.330	1.734	2.101	2.552	2.878	3.610
19.	1.328	1.729	2.093	2.539	2.861	3.579
20.	1.325	1.725	2.086	2.528	2.845	3.552
21.	1.323	1.721	2.080	2.518	2.831	3.527
22.	1.321	1.717	2.074	2.508	2.819	3.505
23.	1.319	1.714	2.069	2.500	2.807	3.485
24.	1.318	1.711	2.064	2.492	2.797	3.467
25.	1.316	1.708	2.060	2.485	2.787	3.450
26.	1.315	1.706	2.056	2.479	2.779	3.435
27.	1.314	1.703	2.052	2.473	2.771	3.421
28.	1.313	1.701	2.048	2.467	2.763	3.408
29.	1.311	1.699	2.045	2.462	2.756	3.396
30.	1.310	1.697	2.042	2.457	2.750	3.385

31.	1.309	1.696	2.040	2.453	2.744	3.375
32.	1.309	1.694	2.037	2.449	2.738	3.365
33.	1.308	1.692	2.035	2.445	2.733	3.356
34.	1.307	1.691	2.032	2.441	2.728	3.348
35.	1.306	1.690	2.030	2.438	2.724	3.340
36.	1.306	1.688	2.028	2.434	2.719	3.333
37.	1.305	1.687	2.026	2.431	2.715	3.326
38.	1.304	1.686	2.024	2.429	2.712	3.319
39.	1.304	1.685	2.023	2.426	2.708	3.313
40.	1.303	1.684	2.021	2.423	2.704	3.307
41.	1.303	1.683	2.020	2.421	2.701	3.301
42.	1.302	1.682	2.018	2.418	2.698	3.296
43.	1.302	1.681	2.017	2.416	2.695	3.291
44.	1.301	1.680	2.015	2.414	2.692	3.286
45.	1.301	1.679	2.014	2.412	2.690	3.281
46.	1.300	1.679	2.013	2.410	2.687	3.277
47.	1.300	1.678	2.012	2.408	2.685	3.273
48.	1.299	1.677	2.011	2.407	2.682	3.269
49.	1.299	1.677	2.010	2.405	2.680	3.265
50.	1.299	1.676	2.009	2.403	2.678	3.261
51.	1.298	1.675	2.008	2.402	2.676	3.258
52.	1.298	1.675	2.007	2.400	2.674	3.255
53.	1.298	1.674	2.006	2.399	2.672	3.251
54.	1.297	1.674	2.005	2.397	2.670	3.248
55.	1.297	1.673	2.004	2.396	2.668	3.245
56.	1.297	1.673	2.003	2.395	2.667	3.242
57.	1.297	1.672	2.002	2.394	2.665	3.239
58.	1.296	1.672	2.002	2.392	2.663	3.237
59.	1.296	1.671	2.001	2.391	2.662	3.234
60.	1.296	1.671	2.000	2.390	2.660	3.232

61.	1.296	1.670	2.000	2.389	2.659	3.229
62.	1.295	1.670	1.999	2.388	2.657	3.227
63.	1.295	1.669	1.998	2.387	2.656	3.225
64.	1.295	1.669	1.998	2.386	2.655	3.223
65.	1.295	1.669	1.997	2.385	2.654	3.220
66.	1.295	1.668	1.997	2.384	2.652	3.218
67.	1.294	1.668	1.996	2.383	2.651	3.216
68.	1.294	1.668	1.995	2.382	2.650	3.214
69.	1.294	1.667	1.995	2.382	2.649	3.213
70.	1.294	1.667	1.994	2.381	2.648	3.211
71.	1.294	1.667	1.994	2.380	2.647	3.209
72.	1.293	1.666	1.993	2.379	2.646	3.207
73.	1.293	1.666	1.993	2.379	2.645	3.206
74.	1.293	1.666	1.993	2.378	2.644	3.204
75.	1.293	1.665	1.992	2.377	2.643	3.202
76.	1.293	1.665	1.992	2.376	2.642	3.201
77.	1.293	1.665	1.991	2.376	2.641	3.199
78.	1.292	1.665	1.991	2.375	2.640	3.198
79.	1.292	1.664	1.990	2.374	2.640	3.197
80.	1.292	1.664	1.990	2.374	2.639	3.195
81.	1.292	1.664	1.990	2.373	2.638	3.194
82.	1.292	1.664	1.989	2.373	2.637	3.193
83.	1.292	1.663	1.989	2.372	2.636	3.191
84.	1.292	1.663	1.989	2.372	2.636	3.190
85.	1.292	1.663	1.988	2.371	2.635	3.189
86.	1.291	1.663	1.988	2.370	2.634	3.188
87.	1.291	1.663	1.988	2.370	2.634	3.187
88.	1.291	1.662	1.987	2.369	2.633	3.185
89.	1.291	1.662	1.987	2.369	2.632	3.184
90.	1.291	1.662	1.987	2.368	2.632	3.183

91.	1.291	1.662	1.986	2.368	2.631	3.182
92.	1.291	1.662	1.986	2.368	2.630	3.181
93.	1.291	1.661	1.986	2.367	2.630	3.180
94.	1.291	1.661	1.986	2.367	2.629	3.179
95.	1.291	1.661	1.985	2.366	2.629	3.178
96.	1.290	1.661	1.985	2.366	2.628	3.177
97.	1.290	1.661	1.985	2.365	2.627	3.176
98.	1.290	1.661	1.984	2.365	2.627	3.175
99.	1.290	1.660	1.984	2.365	2.626	3.175
100.	1.290	1.660	1.984	2.364	2.626	3.174
∞	1.282	1.645	1.960	2.326	2.576	3.090

2.2.5 Metóda na odhalenie odľahlých hodnôt (anomálií) M_3

Zaveďme si tretí model na detekciu odľahlých hodnôt (potencionálnych útokov) ako spojenie už spomínaného algoritmu odľahlých hodnôt M1 a algoritmu pre exponenciónálne vyhladzovanie (Exponentional smoothing)[4].

Exponencionálne vyhladzovanie–Technika, ktorá môže byť použitá na dáta usporiadané v čase ako prognóza. Narozdiel od jednoduchého pohybujúceho sa priemeru, exponencionálny priraďuje exponencionálne klesajúce váhy v čase. Zvyčajne je táto technika aplikovaná na finančný trh a ekonomické údaje, ale môže byť použitý na akýkoľvek diskrétny súbor opakovaných meraní.

Hrubé dáta, ktoré symbolizujú meranie, označme Y_t a výstup algoritmu S_t, ktorý je odhad práve toho, ako by mala vyzerať nameraná hodnota. Ak sa postupnosť začína pre $t = 0$, tak si takto zadefinujme rekurzívny vzťah:

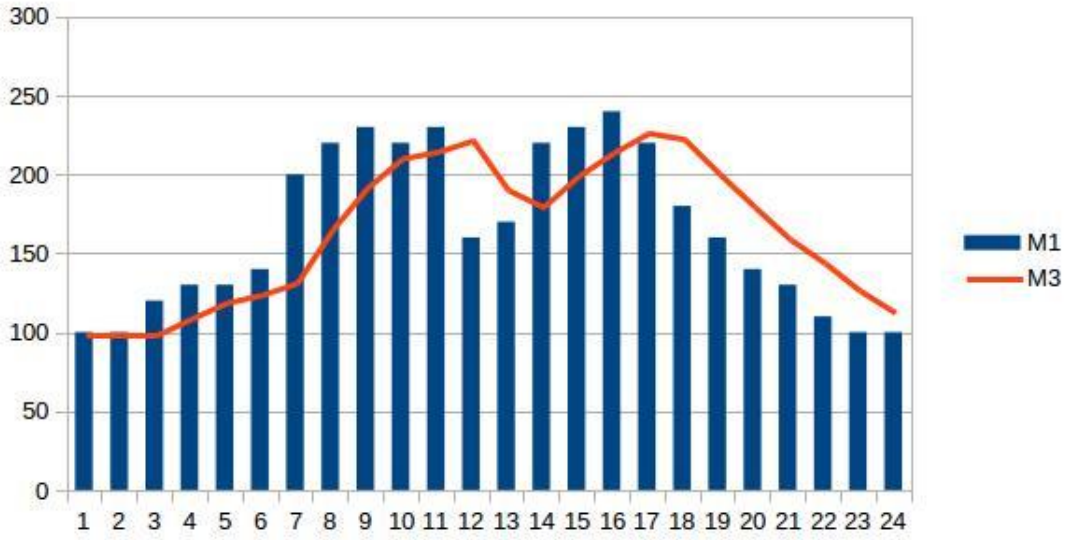
$S_0 = Y_0$

$S_t = \alpha * Y_{t-1} + (1 - \alpha) * S_{t-1}, t > 0,$

kde α je faktor vyhladzovania a platí, že $0 < \alpha < 1$

V našom prípade budú hodnoty Y práve horné hranice z algoritmu M1 a práve takto získavame aj predpoklad z vývoja hodnôt predošlých meraní(hodín).

[4]Použitie tejto metódy bolo inšpirované prezentáciou: János Mohácsi, Gábor Kiss; Anomaly detection for NFSen/nfdump NeFflow engine - with Holt-Winters algorithm. Dostupné na WWW: [http://bakacsin.ki.iif.hu/ kissg/project/nfsen-hw/JRA2-meeting-at-Espoo_slides.pdf]



Obr. 2.5: M3 Baseline - Grafické zobrazenie kvartilov pre celý deň. červená čiara - označuje hranicu, nad ktorou sú všetky hodnoty prehlásené za odľahlé

Kapitola 3

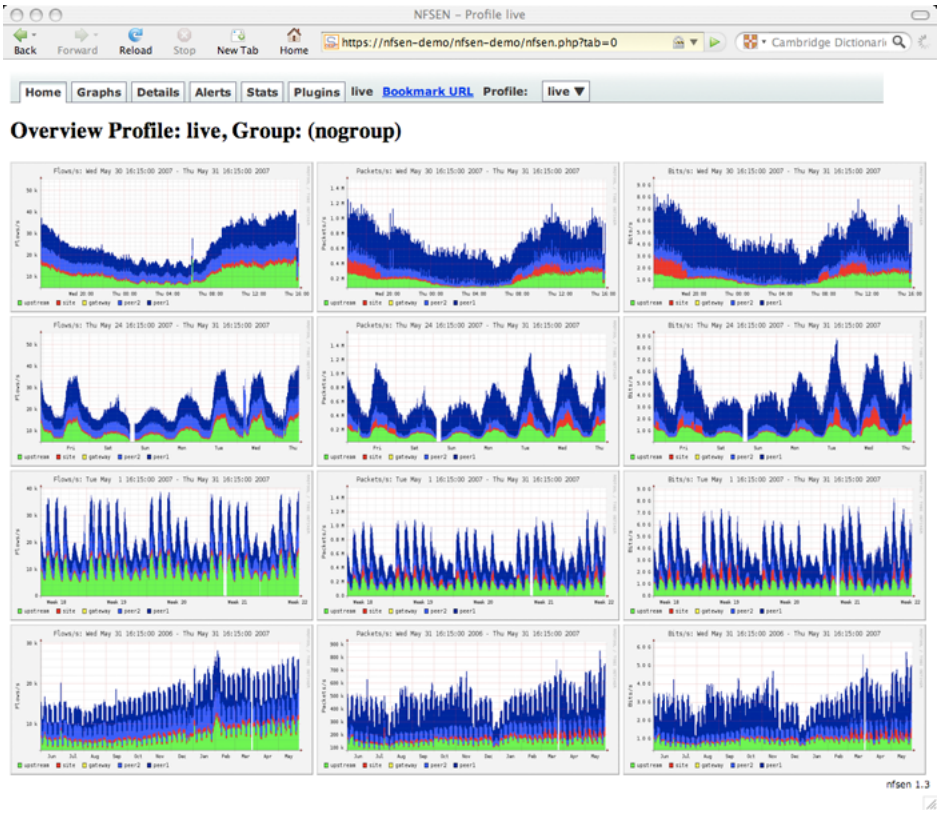
Praktická časť

3.1 Získavanie údajov

Vstupné dáta výsledného programu sú čítané z úložiska a program samotný nemá za úlohu tieto dáta odchytávať. Na testovanie programu sme sa rozhodli využívať voľne šíriteľné open-source sondy na odchytávanie NetFlow záznamov, kolektory a takisto aj nástroje pre ich analýzu. Väčšina z týchto nástrojov bohužial podporuje len NetFlow verzie v5 a podpora v9 a IPFIX je slabšia.

Nfdump tools–skupina nástrojov pre Unixové systémy podporujúce NetFlow verzie v5,v7 a v9.
Medzi základné nástroje nfdump patrí:

- nfcapd (netflow capture daemon)–deamon, správajúci sa ako kolektor, ukladajúci dáta na disk s názvom vo formáte nf-capd.RRRMMDDHHmm, pričom vytvára nový súbor defaultne po 5 minútach.
- nfdump (netflow dump)–načíta a zobrazí dáta uložené kolektorom. Dokáže filtrovať a syntax je podobná nástroju tcp-dump.
- ndprofile (netflow profiler)–filtruje záznamy uložené kolektorom podľa vopred definovaných filtrov (profilov).
- nfpreplay (netflow replay)–preposiela záznamy uložené kolektorom inému kolektoru v sieti.

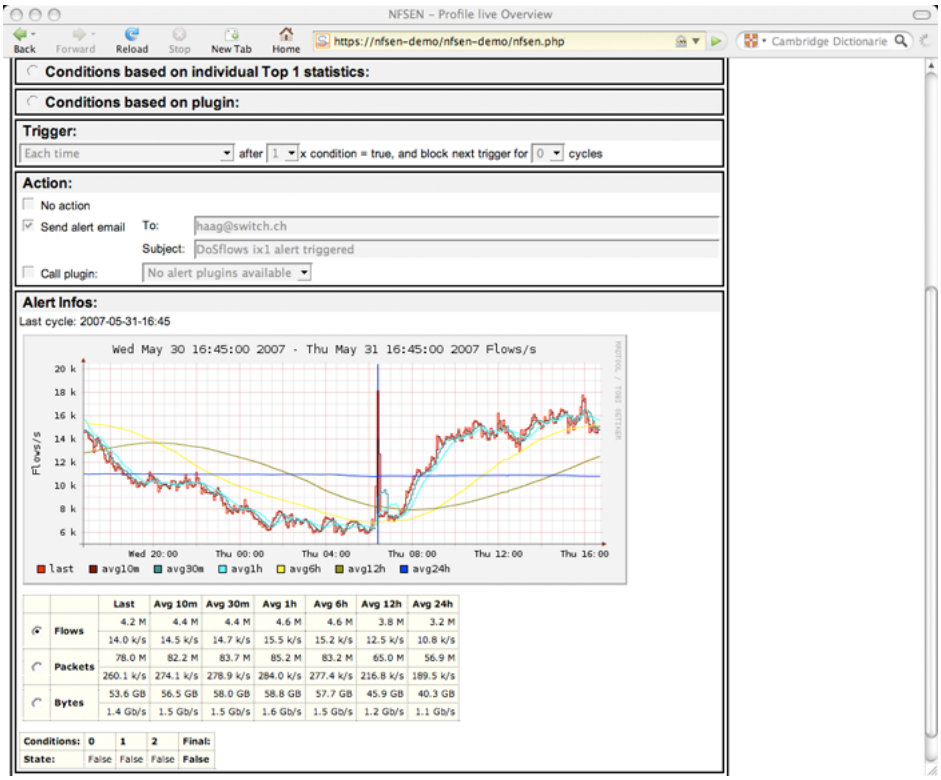


Obr. 3.6: nfsen–RRD grafy

fprobe–NetFlow sonda založená na knižnici libpcap. Zbiera dáta a zasiela ich na kolektor v sieti. Síce ide o softwarové riešenie, slúži však ako náhrada za drahé hardwarové sondy.

NfSen–grafická webová nadstavba nad Nfdump tools umožnujúca:

- Vizualizáciu NetFlow dát pomocou RRD (Round Robin Database)
- Rozširenie funkcionality pomocou pluginov (perl scripty pre backhand, PHP pre fronthand)
- Nastavovanie pravidiel pre alarmy



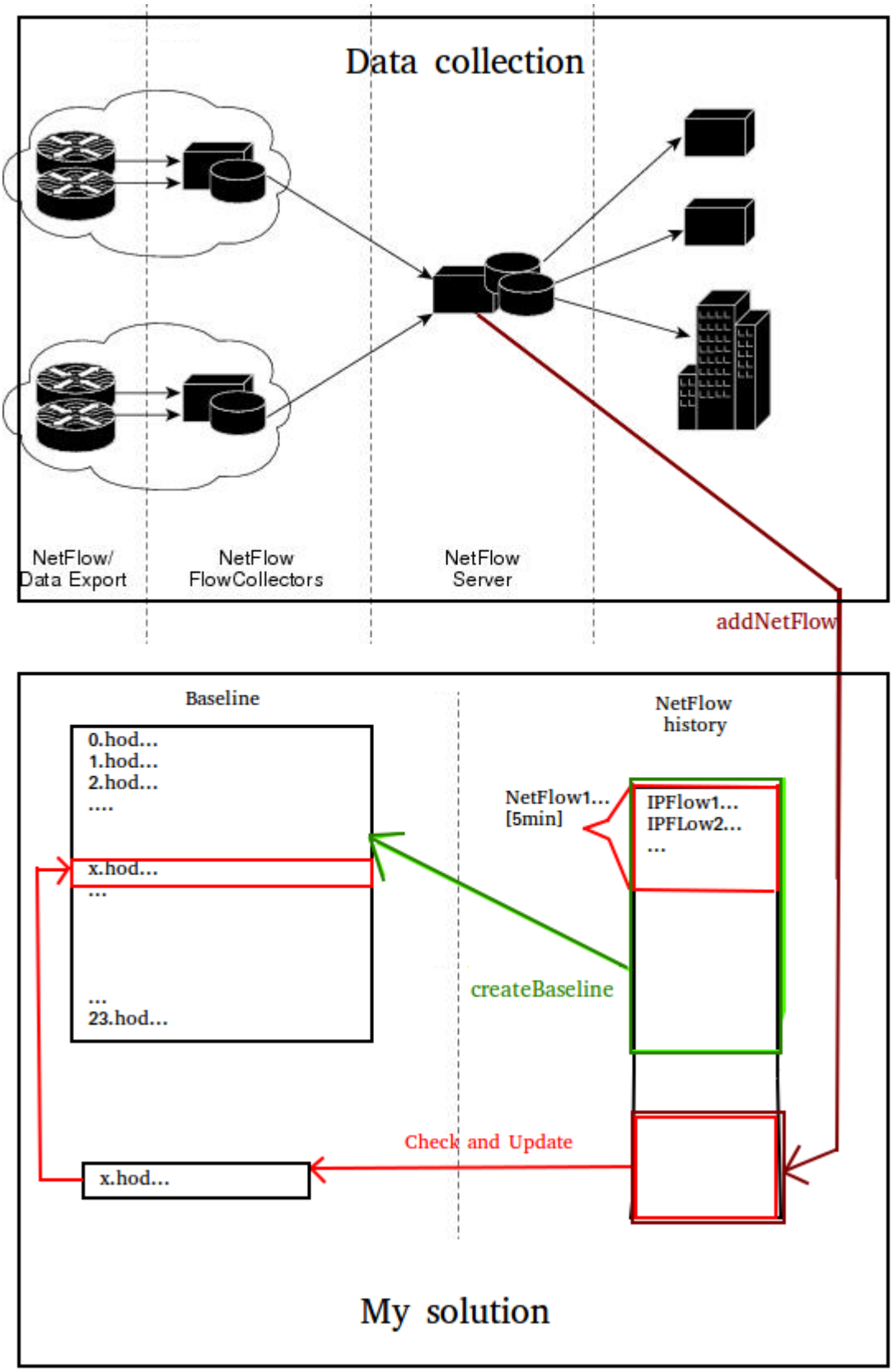
Obr. 3.7: nfsen–Alarmy

3.2 Architektúra riešenia

Naše riešenie ako analyzátor NetFlow dát si po načítaní dát z NetFlow úložiska (servera) ukladá IPFlow záznamy s rovnakým časom (zaokrúhlenie na dolnú časť päťminútového intervalu) pre jednotlivý NetFlow záznam do databázy s tabuľkou obsahujúcou históriu NetFlow.

Pri vytvorení štatistickej krivky sa vyberie z tabuľky, v ktorej je história úsek ohraničený dátumami, z ktorých chceme baseline vytvárať a pre každú hodinu v každom dni vytvorí štatistické dáta. Spôsob výberu, štruktúra a obsah samotných dát je popísaný v kapitole 3.4.

Ak sa používateľ riešenia (alebo automatický systém) rozhodne, že chce overiť nové dáta, vyberie konkrétny súbor dát (alebo sú mu automaticky pridelené), vytvorí štatistický odhad na základe vybraného algoritmu (algoritmy sú popísané v teoretickej časti) a porovná so záznamom pre prislúchajúcu hodinu. V prípade, že sú tieto odhady vyhovujúce podmienkam algoritmu, sa baseline automaticky aktualizuje. V prípade, že sú dáta nevyhovujúce sa postupuje podľa zvoleného módu a teda túto anomáliu systém ohlási a zapíše do logovacích súborov, alebo ignoruje a aktualizuje baseline.



Obr. 3.8: Architektúra nášho riešenia

3.3 Ukladanie dát

Na čítanie dát z disku využívame open-source softvér nfdump, ktorý sme opísali v teoretickej časti. Pre neskoršie spracovanie dát a štatistiky nie je formát, v ktorom ich kolektor ukladá vhodný a preto ako prvé vyberieme vhodnú štruktúru na ukladanie dát. Vybrali sme si možnosť ukladania dát do databázy, konkrétne sme používali SQL relačnú databázu HSQLDB. Táto databáza je flexibilná, rýchla, so schopnosťou čiastočného alebo úplného vykonávania v pamäti. HSQLDB je úplne zadarmo na použitie a distribúciu na základe štandardnej BSD licencie a plne kompatibilný so všetkými hlavnými open source licenciami.

Hlavnou úlohou databázy bude uchovávanie traficu pre neskoršie spracovanie. Túto tabuľku budeme dalej nazývať NetFlow históriou.
Pre naše potreby je nutné uchovávať 12 základných atribútov:

1. ID–jednoznačný identifikátor záznamu
2. startDate–dátum kedy bol záznam vytvorený. Záznam samotný bol vytvorený kolektorom, nie programom samotným.
3. startTime–čas kedy bol záznam vytvorený.
4. duration–trvanie komunikácie medzi stanicami.
5. protocol–protokol, pomocou ktorého stanice komunikovali.
6. srcIP–IP adresa stanice, z ktorej komunikacia odchádzala.
7. srcPort–port stanice, z ktorej komunikacia odchádzala.
8. dstIP–IP adresa stanice, na ktorú komunikácia smerovala.
9. dstPort–port stanice, na ktorú komunikácia smerovala.
10. packets–počet prenesených packetov počas komunikácie.
11. bytes–počet bytov prenesených počas komunikácie.
12. flows–počet IP tokov obsahjúcich informácie o komunikácii.

Je nutné si uvedomiť, že pri vkladaní záznamov do databázy sa volá update za každým riadkom, teda záznamom o jednej komunikácii medzi dvoma stanicami. Pri vkladaní dát sa počíta s väčším počtom záznamov. Ak priemerný počet záznamov

vytvorených za 5 minút je n, jedna hodina má 12 5-minútových intervalov, deň 24 hodín a počítame so záznamami za týždeň, je to počet zavolaní update na databázu $p = 12 * 24 * 7 * n = 2016 * n$
Optimalizáciou tejto operácie sme sa v tejto práci nezaoberali a je to zároveň plán do budúcna. Časovým odhadom pomocou testovacích a vzorových dát sa budeme venovať pri analýze výsledkov.

3.4 Baseline

V tejto práci sa zameriame na hlavné atribúty podľa ktorých budeme baseline a aj následné štatistiky vytvárať, popísané v podsekcii venovanej týmto atribútom. Uchovávanie baseline realizujem pomocou databázy, v ktorej uchovávam jednotlivé štatistické informácie k danej hodine cez deň.

3.4.1 Atribúty výberu

V tejto práci sa zameriavame na odhaľovanie útokov na základe odľahlých hodnôt, pričom výber dát je definovaný šiestimi základnými atribútmi:

1. Vzťah IP adresy a počtu prenesených dát
2. Vzťah portu a počtu prenesených dát
3. Vzťah protokolu a počtu prenesených dát
4. Vzťah spojenia IP, portu, protokolu a počtu prenesených dát
5. Vzťah cieľovej IP adresy a cieľových portov
6. Vzťah cieľovej IP a zdrojovej IP

Vzťahy 1-3 (počet prenesených dát)

Potreba kontroly prenesených dát vznikla za predpokladom určovania štatisticky priateľnej záťaže na jednotlivé IP, porty, či protokoly. Pri vytváraní baseline podľa tohto typu atribútu, sa v záznamoch pre daný atribút jedného štatistického výberu (v našom prípade 5-minútový interval) spočítajú hodnoty pre prenesené dáta, čo predstavuje výsledok jedného štatistického merania pre daný atribút.

Vzťah spojenia IP, portu, protokolu a počtu prenesených dát

Tento vzťah sme navrhli na kontrolu zaťaženia konkrétnej služby. Funkcionalita je rovnaká ako pri vzťahoch 1-3, ale prizerá sa pri vytváraní na zhodu všetkých troch atribútov.

Vzťah cieľovej IP adresy a cieľových portov (skenovanie portov)

Tento vzťah sme navrhli na odhaľovanie útoku nazývaného "skenovanie portov".

Skenovanie portov je jedna z najpopulárnejších techník útočníkov, určená na objavenie bežiacich servisov a ich portov. Všetky počítače spojené do lokálnej počítačovej siete (LAN) alebo Internetu majú spustených veľa servisov a majú povolených veľa portov. Skenovanie portov pomáha potenciálnemu útočníkovi nájsť otvorené porty. Skenovanie portov sa skladá zo zasielania správ postupne na všetky porty. Potom podľa druhu odpovede môžeme zistiť, ktorý port sa používa a ktorý nie. Na základe týchto informácií môže útočník zamerať svoje útoky na dané porty a servisy využitím známych chýb.

Pri vytváraní baseline podľa tohto typu atribútu, sa v záznamoch pre danú cieľovú IP adresu jedného štatistického výberu (v našom prípade 5-minútový interval) vhadzujú porty do množiny otvorených portov. Keďže sa v množine nemôže prvok vyskytovať dvakrát, na konci merania jedného výberu, predstavuje veľkosť tejto množiny výsledok jedného štatistického merania. Pred ďalším meraním sa množina premaže.

Vzťah cieľovej IP a zdrojovej IP (DDOS)

Tento vzťah sme navrhli na odhaľovanie útoku typu DDOS.

DDOS-distribuovaný DOS (odmietnutie služby). Tento útok je príznačný tým, že na jednu cieľovú adresu prichádza nezvyčajne veľa požiadaviek z geograficky rôznych zdrojových IP adries.

Pri vytváraní baseline podľa tohto typu atribútu, sa v záznamoch pre danú cieľovú IP adresu jedného štatistického výberu (v našom prípade 5-minútový interval) vytvoria 4 polia naplnené hodnotami false (pre každý blok jedno pole) a pre každú zdrojovú IP sa do poľa prislúchajúceho bloku, na miesto určujúce hodnotou tohto bloku vloží hodnota true.

Príklad:

$DstIP =$ "1.2.3.4"

$poleA[1] = true$;

$poleB[2] = true;$
$poleC[3] = true;$
$poleD[4] = true;$

Súčet všetkých true hodnôt predstavuje výsledok jedného štatistického merania. Pred ďalším meraním sa množina premaže.

3.4.2 Kvartil

Pri implementácii štatistickej krivky sme potrebovali vytvoriť triedu obsahujúcu základné štatistické informácie. Túto triedu sme nazvali práve podľa týchto hodnôt a teda Kvartil. Viac o kvartiloch píšeme v teoretickej časti.
Táto štruktúra uchováva okrem iného aj 9 základných atribútov:

1. medián–druhý kvartil, alebo tzv. stredná hodnota.
2. mean–aritmetický priemer
3. lowerQuartile–prvý (25%-ný) kvartil
4. upperQuartile–tretí (75%-ný) kvartil
5. upperWhisker–horná hranica vyrátaná pomocou vybraného algoritmu na detekciu odľahlých hodnôt.
6. num–počet všetkých IP tokov, z ktorých bol baseline tvorený.
7. sum–súčet všetkých hodnôt uchovavajúcich IP tokmi, ktoré tvoria baseline.
8. sumSquare–súčet štvorcov všetkých hodnôt uchovavajúcich IP tokmi, ktoré tvoria baseline.
9. standardDeviation–smerodajná odchýlka

3.4.3 Vytváranie baseline-u

Riešenie je implementované tak, že pri vytváraní nového baseline, by mali byť už záznamy uložené v tabuľke s históriou. Je možné vybrať si atribúty podľa ktorých sa má baseline zostaviť. Takisto je možné vybrať si rozsah dátumov, z ktorých chceme baseline vytvárať a štatistický model podľa ktorého budeme hľadať odľahlé hodnoty.

Na základe týchto atribútov sa vytvorí v našej databáze nová tabuľka, ktorá obsahuje 24 záznamov, pričom každý záznam obsahuje štatistické dáta pre danú hodinu, čo v praxi znamená, že sú v tabuľke vložené dáta uchovávané triedou Kvartil (model M1 a M2) a ema-odhadovaná hodnota pomocou exponencionálneho vyhladzovania (model M3). Takáto tabuľka teda predstavuje štatistický prehľad o tom, ako štatisticky vyzerá prevádzka pre zadané atribúty. Pri vytváraní baseline-u je dôležité spomenúť, že dáta z ktorých je vytváraný by mali zodpovedať bežnej prevádzke v sieti, ktorú chceme monitorovať a v čase keď boli odchytávané by nemal prebiehať žiaden útok v sieti.

3.5 Detekcia

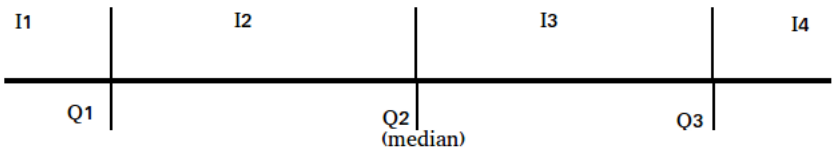
V prípade detekcie je možné fungovať v dvoch módoch a teda v log a update móde. V oboch módoch sa dáta z doposiaľ nespracovaného netflow súboru porovnajú s dátami, ktoré obsahuje baseline pre atribúty, ktoré nás zaujímajú, no v update móde, narozdiel od log módu, sa dáta v tabuľke obsahujúcej baseline prerátajú a aktualizujú bez ohľadu na to, aké dáta sa v súbore nachádzali. V log móde sa síce takisto dáta aktualizujú, ale len v prípade, že sa v dátach zo súboru nenachádza záznam, ktorý prekračuje povolenú hranicu, teda je detekovaný ako anomália. V prípade, že sa anomália detekuje, ostáva baseline neaktualizovaný a vytvorí sa záznam o anomálii v logovacom súbore aj s informáciou o aký záznam a atribúty ide.

Rozsiahlejším problémom je však správne aktualizovanie štatistických dát tak, aby nedošlo k skresleniu údajov. Pri aktualizovaní baseline dát sme však potrebovali zrekonštruovať predchádzajúce dáta, aby sme vedeli určiť nové hodnoty. Je podstatné si uvedomiť, že medzi Q_1 (25%-tným)a Q_3 (75%-tným) je polovica všetkých záznamov a počet záznamov(sum) poznáme. Z tohoto faktu sme si zistili priemernú vzdialenosť medzi dvoma záznamami $d = (Q_3 - Q_1)/(sum/2)$.
Na prerátanie kvartilov však potrebujeme vedieť, kde sa dáta ktoré aktualizujem nachádzajú. Podľa Obr.2.6 sme rozdelili dáta na štyri intervaly a určili sme správanie kvartilov pre každý z nich.

Rozoberme si teda všetky štyri možnosti:

Algorithm 2 Aktualizácia baseline

for all $Y : X$ **do**
  if $(Y > 0)\&\&(Y < Q_1)$ **then**
    $Q_1 - = d/4$
    $median - = d/2$
    $Q_3 - = 3d/4$
  end if
  if $(Y > Q_1)\&\&(Y < Q_2)$ **then**
    $Q_1 + = 3d/4$
    $median - = d/2$
    $Q3 - = 3d/4$
  end if
  if $(Y > Q_2)\&\&(Y < Q_3)$ **then**
    $Q_1 + = 3d/4$
    $median + = d/2$
    $Q3 - = 3d/4$
  end if
  if $(Y > Q_3)\&\&(Y < max)$ **then**
    $Q_1 + = 3d/4$
    $median + = d/2$
    $Q3 + = 3d/4$
  end if
end for



Obr. 3.9: Rozdelenie intervalov pre update baseline-u

Z aktualizovaných kvartilov sa ešte doráta horná hranica podľa prislušného štatistického modelu.

Takisto je potrebné aktualizovať smerodajnú odchýku

$S^2 = \frac{1}{N-1} \sum (Y_i - \overline{Y})^2$

Všetky dáta však nieje možné uchovávať. Odvádzame však tento vzorec:

$S^2 = \frac{1}{N-1} \sum Y_i^2 - 2x_i\overline{Y} + \overline{Y}^2$

Toto už vieme zapísať ako:

$S^2 = \frac{1}{N-1}(sumSquare - sum * 2N * mean + n * mean^2)$

Takže pre nový záznam Y_{new} aktualizujeme baseline:

Algorithm 3 Aktualizácia baseline

for all $Y_{new} : X$ **do**

$N++$

$sum+ = Y_{new}$

$sumSquare+ = Y_{new}^2$

end for

$standardDeviation = \frac{1}{N-1}(sumSquare - sum * 2N * mean + n * mean^2)$

Posledná hodnota, ktorú je potrebné dorátať(potrebná pre model M3) je hodnota pre exponencionálne vyhladzovanie. Túto hodnotu vyrátame nasledujúcim algoritmom:

Keďže baseline je už vytvorený, dáta ktoré kolektor ukladá prichádzajú v 5 minutových intervaloch a súbor, ktorý sa vytvorí je možné spracovať za čas omnoho kratší, tak detekcia funguje v reálnom čase.

Algorithm 4 Aktualizácia baseline

for all $Y_t : X$ **do**
  if $t == 0$ **then**
    $S_t == Y_t$
  end if
  if $t > 0$ **then**
    $S_t = \alpha * Y_{t-1} + (1 - \alpha) * S_{t-1}$
  end if
end for

Hľadanie hranice

Z baseline dát vieme pre príslušný algoritmus zistiť, či nové dáta súhlasia s baseline dátami.

- M1-zisťuje, či suma tokov $set < max$, pričom $max = Q_3 + A * (Q_3 - Q_2) + C$.
- M2-zisťuje, či $G \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(N,N-2)}}{N-2+t^2_{a/(N,N-2)}}}$
- M3-zisťuje, či suma tokov $set < S_t$ pre danú hodinu t.

3.6 Popis tried a metód

V tejto kapitole popíšeme základné triedy a ich metódy.

DatabaseFlowFile

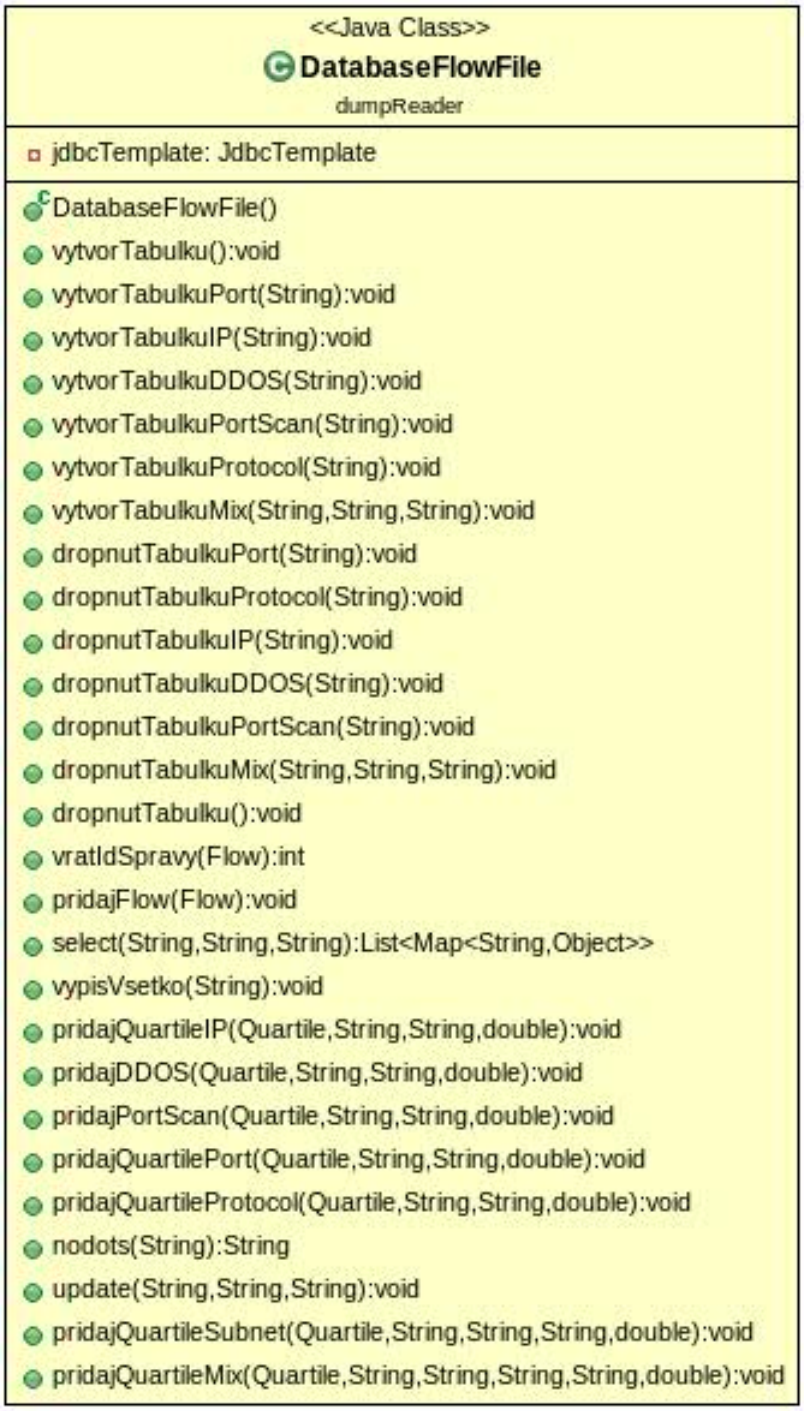
Táto metóda obsahuje všetky metódy potrebné na komunikáciu s databázou, ako vytváranie tabuliek, mazanie tabuliek, pridávanie a aktualizovanie záznamov pre každú tabuľku (Obr. 3.10).

Kvartil

Kvartil je trieda obsahujúca základné štatistické dáta a metódy na ich výpočet (Obr. 3.11).

Baseline

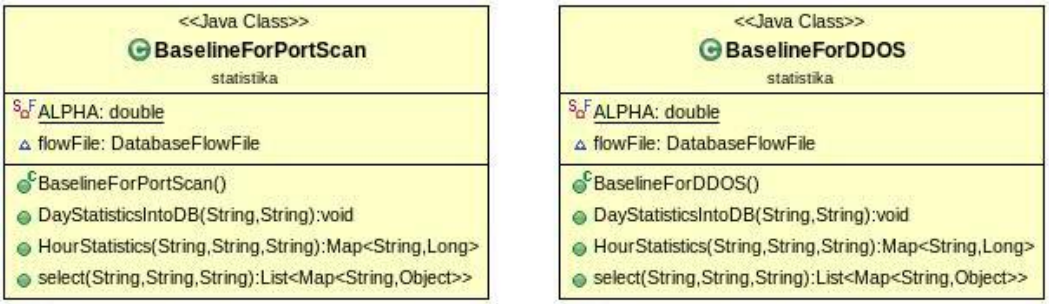
Baseline je vytváraný pomocou tried špecifikujúcich o aký cieľ skúmania ide. Konkrétne typy baseline sú uvedené v kapitole venovanej baseline (Obr. 3.12,Obr. 3.13).



Obr. 3.10: Trieda DatabaseFlowFile

Obr. 3.11: Trieda Kvartil

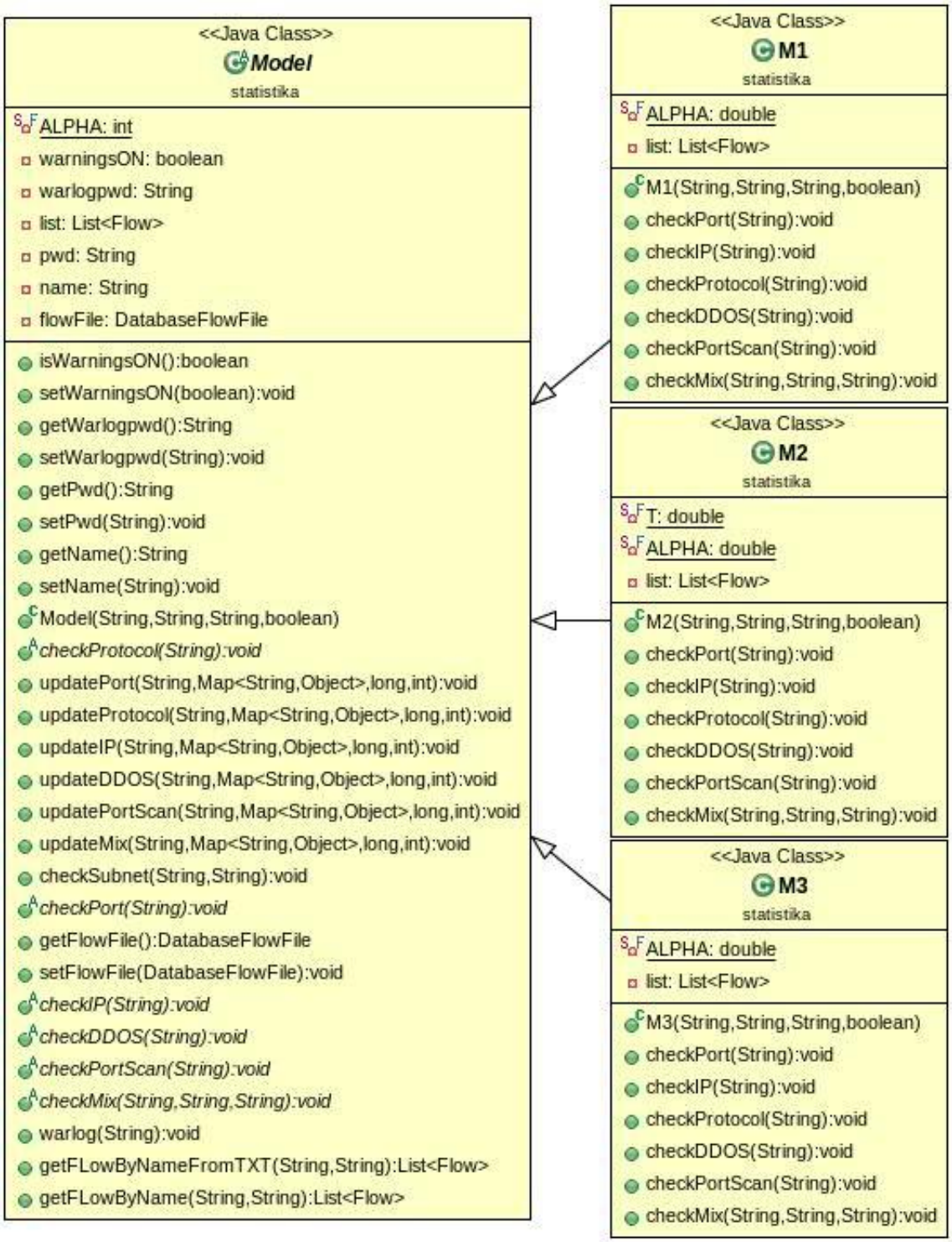
Obr. 3.12: Triedy vytvárajúce baseline pre daný atribút [IP, port, protocol, subnet, mix]



Obr. 3.13: Triedy vytvárajúce baseline pre daný atribút [útok skenovaním portov, DDOS útoky]

Modely

Trieda model definuje metódy potrebné pre načítanie nových dát a porovnanie s existujúcim baseline. Pre každý atribút výberu baseline je vytvorená osobitná metóda, ktorej spôsob porovnávania je rozšírený triedou daného modelu (M1,M2,M3) (Obr. 3.14).



Obr. 3.14: Triedy štatistických modelov na porovnávanie nových dát s vytvoreným baselineom

Kapitola 4

Testovanie a analýza výsledkov

4.1 Praktické riešenie zberu údajov

V tejto sekcii si ozrejmíme, ako postupovať pri zbere dát potrebných pre naše riešenie a navrhneme si testovacie riešenie zberu dát, kedy nie je potrebné investovať do finančne náročných hardwarových riešení.

4.1.1 Sondy-odchytávanie údajov

Na odchytávanie údajov, vytváranie štatistiky v podobe IP tokov a zasielanie (exportovanie) tejto štatistiky na uloženie a ďalšiu analýzu na kolektor s NetFlow štandardom sa využívajú takzvané sondy. Odchytávanie údajov nie je súčasť výsledného produktu, no pre vytváranie testov budeme sondu potrebovať. V produkčnom prostredí sa využívajú hlavne hardwarové sondy, ale pre naše potreby budem používať softwarovú sondu, presnejšie nástroj fprobe, ktorý opisujeme v praktickej časti. Systém sme vytvárali na operačnom systéme Linux založenom na distribúcii Ubuntu s baličkovým systémom, v ktorom sa nachádza aj fprobe. Pre spustenie fprobe je potrebné zadať do príkazového riadku

$\#/usr/sbin/fprobe\ -eth0\ localhost:555$

Takto budeme odchytávať komunikáciu na $eth0$ a preposielať na kolektor, ktorý je na localhoste na porte 555.

4.1.2 Kolektory–Ukladanie záznamov

Na ukladanie údajov sme použili balíček nfdump tools. Konkrétne nfcapd (netflow capture daemon). Pre spustenie nfcapd je potrebné zadať do príkazového riadku

$\#/usr/sbin/nfcapd\ -w\ -D\ -p\ 555\ -I\ eth0\ -S\ 1\ -l\ /var/nfcapd$

Takto budeme ukladať netflow dáta do adresára $/var/nfcapd$, pričom daemon počúva na rozhraní $eth0$ a porte 555. Prepínačom $-S\ 1$ sme zaručili, že dáta sa budú triediť do adresárov podľa dátumu.

Pre tento prepínač sú:

1. default – no hierarchy levels
2. %R/%m/%d – rok/mesiac/deň
3. %R/%m/%d/%H – rok/mesiac/deň/hodina
4. %R/%T/%t – rok/týždeň v roku/deň v týždni
5. %R/%T/%u/%H – rok/týždeň v roku/deň v týždni/hodina
6. %R/%D – rok/deň v roku
7. %R/%D/%H – rok/deň v roku/hodina
8. %R-%m-%d – rok-mesiac-deň
9. %R-%m-%d/%H – rok-mesiac-deň/hodina

A prepínač $-D$ spúšťa kolektor ako daemona na pozadí. Záznamy sú ukladané v 5 minútových intervaloch do súborov opisujúcich tento interval s názvom v tvare $nfcapd.RRRMMDDHHm$.

4.1.3 Zobrazenie záznamov

Pre jednoduché zobrazenie záznamov na disku môžeme použiť balíček nfdump tools. Konkrétne nfdump (netflow dump). Stačí zadať do príkazového riadku

$\#/usr/sbin/nfdump\ -r\ nfcapd.RRRMMDDHHmm$,

kde RRRMMDDHHmm sa nahradí časom, v ktorom bol záznam získaný zaokrúhlený na 5 minút (názov súboru v ktorom sa nachádza). Práve výstup tohto volania využívame ako vstupné dáta v našom programe a spôsob použitia uvádzame v praktickej časti.

4.2 Testovanie

Keďže sme v práci nevyužívali žiadne reálne dáta, museli sme si ich laboratórne vytvoriť. Všetky testovacie dáta sú priložené na CD. Súčasťou systému sú aj triedy na testovanie pomocou týchto dát.V prvom rade sme vytvorili metódy na vytváranie dlhodobého toku dát na základe jedného päťminútového záznamu, vďaka ktorým sme boli chopní testovať ukladanie histórie do databázy a vyhodnotiť rýchlosť tejto operácie. Pri testovaní sme namiesto štandardnej štruktúry ukladali dáta v plain texte a teda aj doimplementovali v našom systéme metódy na čítanie dát z plain textu.

Test1: Vkladanie záznamov do databázy

Objem dát: Vkladaná vzorka na jeden deň bola 70560 záznamov rovnomerne rozložených pre všetky päťminútové záznamy.
Čas behu: 2hodín 8minút 2sekúnd 7stotín, teda približne 128min. na 70560 IPFlow záznamov, čo znamená, že zápis jedného IPFlow záznamu v priemere trvá 0.11 sekundy.
Závažnosť výsledkov: Dlhší čas na vkladanie záznamov má pre naše riešenie nižšiu prioritu, lebo tieto úkony sa môžu automatizovať nezávisle od ostatných metód (napr. cez noc, alebo počas vyhodnocovania záznamov, ktoré sa už v databáze nachádzajú).
Riešenie: Keďže úzkym hrdlom je neustále dopytovanie do SQL databázy, navrhujeme zaradiť medzi plány do budúcna zaradiť prechod na inú štruktúru ukladania dát, pri ktorej využijeme paralelizmus, ako je Hadoop, alebo prípadné využitie NoSQL.

Test2: Vytváranie baseline pre port

Výsledky tohto testu sú zrovnateľné s testami na IP, protokol a mix, pretože ide o rovnaký počet operácií na záznam, rovnaké problémy a plány do budúcna. **Objem dát:** Prezeraných záznamov je v histórii 493920. Záznamov, z ktorých sa baseline bude vyrátavať je z toho 72576.
Čas behu: 8minút 31sekúnd 6stotín, teda približne 511sec. na 72576 IPFlow záznamov, čo znamená, že zápis jedného IPFlow záznamu v priemere trvá 0.7 stotín sekundy.
Závažnosť výsledkov: Dlhší čas na vytváranie baseline ma pre naše riešenie nižšiu prioritu, lebo tieto úkony sa môžu automatizovať nezávisle od ostatných metód (napr. cez noc).

Riešenie: Ako vhodný plán do budúcna sa ukazuje paralélne spracovávanie baseline pre jednotlivé hodiny, kedy by sa čas potrebný na výpočet znížil 24-krát.

Test2: Vytváranie baseline pre skenovanie portov na IP

Objem dát: Prezeraných záznamov je v histórii 493920. Záznamov, z ktorých sa baseline bude vyrátavať je z toho 231840.
Čas behu: 8minút 15sekúnd 6stotín, teda približne 495sec. na 231840 IPFlow záznamov, čo znamená, že zápis jedného IPFlow záznamu v priemere trvá 0.2 stotiny sekundy.
Závažnosť výsledkov: Rýchlosť tohto riešenia je spôsobená rozumným návrhom zberu rôznych portov, kde sa neaktualizujú dáta pre každý záznam, ale len pre konkrétnu hodinu. Zádrhel bol len v tom, že je potrebné kontrolovať, či sa port v množine nenachádza, čo však pri malých počtoch portov môžeme považovať za konštantu a teda to môžeme zanedbať.
Riešenie: Pre ešte vyššiu rýchlosť navrhujeme spracovávať jednotlivé hodiny baselineu distribuovane v rôznych vláknach.

Test3: Vytváranie baseline pre DDOS na IP

Objem dát: Prezeraných záznamov je v histórii 493920. Záznamov z ktorých sa baseline bude vyrátavať je z toho 231840.
Čas behu: 8minút 31sekúnd 5stotín, teda približne 511sec. na 231840 IPFlow záznamov, čo znamená, že zápis jedného IPFlow záznamu v priemere trvá 0.2 stotiny sekundy.
Závažnosť výsledkov: Rýchlosť tohto riešenia je spôsobená rozumným návrhom zberu rôznych IP adries, kde sa neaktualizujú dáta pre každý záznam, ale len pre konkrétnu hodinu. Ukladanie opakujúcich sa IP adries je rozumne riešené cez pole konštantnej dĺžky (4*255) a teda nevznikol žiaden problém.
Riešenie: Pre ešte vyššiu rýchlosť navrhujeme spracovávať jednotlivé hodiny baselineu distribuovane v rôznych vláknach.

Testy spoľahlivosti modelov

Pri testovaní spoľahlivosti modelov je takmer nevyhnutné testovať konkrétny model voči reálnemu toku dát. V našich podmienkach sme aspoň vytvorili trojice baseline, pozitívny a negatívny test, pre ktoré platí, že pre baseline pozitívny test baseline updatne pre danú metódu a negatívny detekuje anomáliu. Pre skenovanie portov, sme použili dáta, pri ktorých zbere bol vedený ping na jednotlivé porty. Pri všetkých ostatných testoch boli útoky vyrábané umelo pomocou viacnásobného kopírovania traficu.

Testy rýchlosti modelov

Pri testovaní rýchlosti nezáleží na tom, aký model testujeme, lebo ich zložitosť je približne rovnaká a aktualizujú sa rovnako všetky dáta a preto sme aj dostali rovnaké výsledky. Pre jeden model skontrolovanie piatich baseline (IP,port, protocol, DDOS, skenovanie portov) trval výpočet s aktualizovaním 2 sekundy, čo v prípade, že nové dáta prichádzajú každých 5 minút môžeme nazvať kontrolou v reálnom čase.

Kapitola 5

Záver

Cieľom tejto práce bolo navrhnúť a implementvať algoritmus na detekciu štatistického profilu (baseline) sieťovej prevádzky na základe NetFlow dát, ako aj navrhnúť a naimplementovať algoritmy na detekciu útokov na základe štatistických odchýlok od tohto profilu.

Na začiatku práce sme sa oboznámili s protokolom NetFlow, jeho výhodami a spôsobom použitia. Takisto sme sa oboznámili s niektorými existujúcimi riešeniami a objasnili základné štatistické pojmy potrebné pre návrh algoritmov na odhaľovanie anomálií. Poznatky, ktoré sme získali sú popísané v teoretickej časti tejto práce. V tejto časti nájdeme aj teoretický návrh troch algoritmov na výpočet anomálií a teda potencionálnych útokov.

V praktickej časti sme sa zamerali na implementáciu systému na analýzovanie NetFlow dát. Systém, ktorý sme navrhli je schopný analyzovať NetFlow dáta a vytvoriť štatistické profily (baseline) na základe rôznych atribútov, ako aj analyzovať nové dáta porovnaním s týmto profilom na základe troch navrhnutých algoritmov. Pre lepšiu predstavu čitateľa sme vytvorili grafické užívateľské prostredie na kontrolovanie nových NetFlow dát.

Na konci práce sme navrhli testovanie systému a vytvorili testovacie sady na ktorých sme otestovali spoľahlivosť, merali rýchlosť systému a podľa výsledkov meraní sme stanovili ciele a vízie do budúcna.

Na záver by sme radi skonštatovali, že sme v práci splnili stanovené cieľe a dúfame, že bude nápomocná či už ako teoretický podklad, alebo ako samotné rozhranie (API) pre detekčné systémy založené na NetFlow dátach.

Zoznam použitej literatúry

[1] GRAHAM J. G. UPON, IAN COOK: Understanding Statistics. 1. vydanie. Oxford University Press, Oxford, 1997. ISBN 0199143919.

[2] B. CLAISE, Ed.: Cisco Systems NetFlow Services Export Version 9. In RFC 5153 [onlne].
Dostupný z WWW: [http://www.ietf.org/rfc/rfc3954.txt].

[3] SIMPSON B., TOUSS F.: HyperSQL User Guide: HyperSQL Database Engine (HSQLDB) 2.3. 2013.
Dostupný z WWW: [http://hsqldb.org/doc/2.0/guide/guide.pdf].

[4] TUKEY, J. W.: Exploratory Data Analysis. Reading, MA: Addison-Wesley, p. 44, 1977.

[5] HAAG P.: User Documentation nfdump & NfSen.
Dostupný z WWW: [http://www.first.org/conference/2006/papers/haag-peter-papers.pdf].

[6] NetFlow Services Solutions Guide. Dostupný z WWW:
[http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.pdf].

[7] ZEY CH., NIST/SEMATECH e-Handbook of Statistical Methods,
Dostupný z WWW: [http://www.itl.nist.gov/div898/handbook/]