Spracovanie sieťových prevádzkových údajov netflow

Vladimír Homola

3Ib, 2020 – 2021

Abstrakt. V práci sa venujeme analýze netflow údajov, spracovaniu bezpečnostných údajov. Na tieto potreby sme navrhli riešenie vychádzajúce z algoritmu použitého v článku: *Big data analytics for network anomaly detection from netflow data*. Navrhnuté riešenie využíva rozdelenie netflow do časových intervalov (kde my uvažujeme aj obojsmernú komunikáciu, čo pôvodný algoritmus nerobil a je to podľa nás chyba), následnú agregáciu údajov v týchto intervaloch podľa zdrojových ip adries a klastrovanie, kde my sme sa zamysleli aj nad iným typom klastrovania a porovnávame výsledky iného výskumu v oblasti s našími výsledkami nad rovnakými datasetmi. Naše navrhnuté riešenie hľadá podozrivé záznamy o toku (napr. botnet) a upozorňuje na nich s vysokou mierou úspešnosti.

Kľúčové slová: netflow, spracovanie dát, detekcia útokov

1 Úvod

V posledných rokoch boli navrhované a vyvíjané rôzne prístupy monitorovania siete, každý z nich na rôzny účel. Môžeme ich rozdeliť do 2 hlavných kategórií. Prvou z nich sú pasívne prístupy monitorovania siete. Tie sledujú existujúci tok v sieti pri prechádzaní tzv. meracími bodmi, čiže skúmajú tok generovaný užívateľmi siete, napr. packet capture, flow export. Druhou sú aktívne prístupy monitorovania siete. Tie pomocou vkladania prenosu do siete vykonávajú rôzne typy meraní. Sú implementované nástrojmi ako Ping alebo Traceroute. V našej práci sa zameriavame na tie pasívne, konkrétne netflow.

V dnešnej dobe sa kybernetické útoky dejú dlhodobo a systematicky, na rozdiel od minulosti, kedy boli organizované skôr jednoducho a náhodne. Kyberbezpečnosť sa stáva viac a viac dôležitou a krajiny začali investovať nemalé peniaze na ochranu kritickej infraštruktúry a zároveň samotné kyberútoky začínaju byť viac komplexné, premyslené a deštruktívne. Preto sa v našej práci venujeme aj detekcii podozrivého správania zo sieťových prevádzkových údajov netflow.

1.1 Motivácia

Motiváciou bolo pozrieť sa bližšie na fungovanie sieťovej prevádzky a jej monitorovanie a odhaliť podozrivé správanie v sieti. Keďže v dnešnej dobe sa kybernetické útoky dejú dlhodobo a systematicky, tak chceme na základe analýzy zozbieraných dát netflow vedieť povedať niečo viac o dianí v sieti, či nastal nejaký útok

(evidencia bezpečnostných incidentov) alebo ide o bežnú prevádzku. Ak nastal nejaký útok, pokúsiť sa zistiť cieľ útoku, a z ktorých ip adries bol útok vykonávaný. Celkovo mať prehľad o tom, čo sa deje v sieti. Vďaka tomuto by sme vedeli do budúcna predpovedať pravdepodobné ciele možného útoku alebo zakročiť včas, aby útok ani nenastal.

1.2 Ciele práce

Na základe motivácie aktuálneho stavu sme si stanovili výskumnú oblasť, ktorú vieme rozdeliť na parciálne ciele, ktorým sa venujeme v práci:

1. Analýza sieťových prevádzkových údajov netflow.
2. Návrh distribúcie sieťových prevádzkových údajov netflow k centrálnemu uzlu.
3. Dátová analýza zozbieraných sieťových prevádzkových údajov netflow.

1.3 Prehľad podobných prác

Predtým, než sme sa pustili do tvorby našej práce, sme vyhľadali podobné práce. Pozreli sme sa na to, ako v nich riešili podobnú problematiku.

Detekciou anomálií v sieti zo sieťových prevádzkových údajov netflow sa zaoberajú v práci [1]. Konkrétne sa v nej hovorí o tom, že útoky sa v dnešnej dobe dejú systematicky a dlhodobo. Navyše vysoký výpočtový objem a neustále zmeny v distribúcií sieťových údajov sťažili analýzu údajov a detekciu abnormálneho správania v sieti. Z tohto dôvodu sa riešením stali big data riešenia.

Práca [2] sa zaoberá analýzou tokov a hovorí o nej, že vďaka zameraniu sa na analýzu tokov namiesto samotných paketov je škálovateľnejšia než tradičná analýza prenosu na základe paketov. Konštatuje, že monitorovanie toku sa stalo prevládajúcou metódou monitorovania prevádzky vo vysokorýchlostných sieťach. Na rozdiel od toho, čo sa často predpokladá, sú všetky fázy monitorovania toku úzko spojené. Práca obsahuje tutoriál pre všetky fázy nastavenia monitorovania toku.

Botnety sa stali vážnym problémom internetu. Konkrétne sa v práci [3] zaoberajú detekciou botnet Command and Control (C&C) serverov prostredníctvom rozsiahlej analýzy netflow. Hovoria o tom, že botnety sú aj naďalej významným problémom v rámci internetu. Dva hlavné faktory, ktoré bránia celoplošnému vývoju systému, ktorý by detekoval botnet, sú nedostupnosť surových sieťových dát, a ak sú aj dostupné, ich analýza v reálnom čase je ťažká výzva. Problém analýzy netflow je, že môže obsahovať false positive odhalenie botnetu. V práci vyvinuli spôsob detekcie v reálnom čase Disclosure, vďaka ktorému sa to dá znížiť.

2 Práca s datasetom

2.1 Charakteristika datasetu

Dataset, s ktorým sme doposiaľ pracovali je CTU-13 Dataset. Daný dataset je „labeled dataset“, to znamená, že každý záznam datasetu obsahuje popis, ktorý hovorí, či išlo o botnet, normálnu prevádzku alebo prevádzku na pozadí. Pochádza z Českej technickej

univerzity a jeho cieľom bolo vo veľkom zaznamenať reálnu botnet prevádzku zmiešanú s normálnou prevádzkou a prevádzkou na pozadí. Celkovo daný dataset obsahuje 13 scenárov s rôznymi vzorkami botnetu. Každý riadok datasetu obsahuje nasledujúce polia: čas začatia toku, trvanie toku, protokol, zdrojová ip adresa, zdrojový port, smer komunikácie, cieľová ip adresa, cieľový port, stav protokolu, zdrojový typ služby, cieľový typ služby, počet prenesených paketov, počet prenesených bajtov, počet bajtov odoslaných zo zdroja, popis toku.Vzhľad datasetu možno vidieť na Obrázku 1. Každý scenár možno spracovávať ako samostatný dataset. Jednotlivé scenáre, množstvo dát v nich, ukazuje Tabuľka 1.

```
StartTime,Dur,Proto,SrcAddr,Sport,Dir,DstAddr,Dport,State,sTos,dTos,TotPkts,TotBytes,SrcBytes,Label
2011/08/18 10:21:46.633335,1.060248,tcp,93.45.239.29,1611,   ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:19:49.027650,279.349152,tcp,62.240.166.118,1031,  <?>,147.32.84.229,13363,SRPA_PA,0,0,15,1318,955,flow=Background-TCP-Attempt
2011/08/18 10:22:07.160628,166.390015,tcp,147.32.86.148,58067,   ->,66.235.132.232,80,SR_SA,0,0,3,212,134,flow=Background-TCP-Established
2011/08/18 10:26:02.052163,1.187083,tcp,147.32.3.51,3130,   ->,147.32.84.46,10010,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:26:52.226748,0.980571,tcp,88.212.37.169,3134,   ->,147.32.84.118,6881,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:27:57.611601,1.179357,tcp,94.44.60.103,49905,   ->,147.32.84.118,6881,S_RA,0,0,4,268,148,flow=Background-TCP-Attempt
2011/08/18 10:28:15.463038,1.140237,tcp,2.159.127.100,1378,   ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:28:37.132447,12.058948,tcp,213.233.154.219,36381,   ->,147.32.84.229,13363,SRA_SA,0,0,7,508,208,flow=Background-TCP-Established
2011/08/18 10:29:03.150806,0.942243,tcp,88.212.37.169,62055,   ->,147.32.84.118,6881,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:29:43.750355,3.547213,tcp,95.210.161.212,58571,   ->,147.32.84.118,6881,S_RA,0,0,8,488,248,flow=Background-TCP-Attempt
2011/08/18 10:30:05.331403,1.459868,tcp,94.44.60.103,49988,   ->,147.32.84.118,6881,S_RA,0,0,4,268,148,flow=Background-TCP-Attempt
2011/08/18 10:30:54.064370,0.999792,tcp,85.132.162.9,30333,   ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:32:17.843382,1.015245,tcp,88.212.37.169,48256,   ->,147.32.84.118,6881,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:33:08.505665,3.001954,tcp,140.115.25.74,49843,   ->,147.32.84.229,13363,SR_SA,0,0,3,192,126,flow=Background-TCP-Established
2011/08/18 10:33:11.090578,1.520176,tcp,94.44.60.103,50180,   ->,147.32.84.118,6881,S_RA,0,0,4,268,148,flow=Background-TCP-Attempt
```

Obrázok 1. Formát súboru datasetu.

Tabuľka 1. Množstvo dát v jednotlivých botnet scenároch.

Id	Trvanie(h)	#paketov	#netflow	Veľkosť	Bot	#botov
1	6,15	71 971 482	2 824 637	52GB	Neris	1
2	4,21	71 851 300	1 808 123	60GB	Neris	1
3	66,85	167 730 395	4 710 639	121GB	Rbot	1
4	4,21	62 089 135	1 121 077	53GB	Rbot	1
5	11,63	4 481 167	129 833	37,6GB	Virut	1
6	2,18	38 764 357	558 920	30GB	Menti	1
7	0,38	7 467 139	114 078	5,8GB	Sogou	1
8	19,5	155 207 799	2 954 231	123GB	Murlo	1
9	5,18	115 414 321	2 753 885	94GB	Neris	10
10	4,75	90 389 782	1 309 792	73GB	Rbot	10
11	0,26	6 337 202	107 252	5,2GB	Rbot	3
12	1,21	13 212 268	325 472	8,3GB	NSIS.ay	3
13	16,36	50 888 256	1 925 150	34GB	Virut	1

2.2 Práca s jednotlivým scenárom datasetu a fungovanie algoritmu

Program dostane na vstup cestu k datasetu a cestu, kde má uložiť výsledky algoritmu. Následne načíta daný dataset a spýta sa používateľa, či si želá spracovať a zanalyzovať celý dataset naraz alebo ho rozdeliť na časové okná dĺžky zadanej používateľom a každé z okien spracovať a vyhodnotiť zvlášť.

Ďalej program, či už vybrané okno datasetu alebo celý dataset podľa toho ako zvolil používateľ, rozdelí do 1-minútových časových intervalov (1 minúta je dostačujúca na to, aby boli zachytené anomálie a zároveň neobsahuje príliš veľa záznamov). Následne

sú záznamy netflow agregované podľa zdrojových ip adries. Pre každú ip adresu je vypočítaný počet jedinečných zdrojových portov, počet jedinečných cieľových ip adries, počet jedinečných cieľových portov, počet záznamov netflow a počet prenesených bajtov a paketov. Získané dáta sú ďalej štandardizované podľa nasledujúceho vzorca (1), kde p je priemer, s je smerodajná odchýlka agregovaných údajov z predchádzajúceho kroku algoritmu a x sú údaje jednej konkrétnej ip adresy, ktoré sme vypočítali pri agregácií. Týmto vyrovnáme variabilitu dát a zároveň štandardizované dáta sú menej ovplyvňované odľahlými hodnotami.

$$z = \frac{x-p}{s} \tag{1}$$

Nasledujúcim bodom algoritmu je klastrovanie. Keď má algoritmus štandardizované dáta, tak agregované záznamy netflow sú klastrované využitím k-means algoritmu. Klastrovací algoritmus je nastavený na vytvorenie 2 klastrov (anomálny a neanomálny) a na maximálne 1000 iterácií, ak sa algoritmus neukončí skôr. Na začiatku sa náhodne zvolia 2 centroidy (centrá klastrov) a následne sa vypočíta matica vzdialeností jednotlivých elementov (v našom prípade je elementom zdrojová ip adresa) od jednotlivých centroidov. Potom sa daný element priradí k jednému z centroidov podľa toho, ku ktorému má menšiu vzdialenosť. Keď je každý element pridelený do klastra, vypočíta sa priemer vlastností elementov daného klastra a na základe toho sa vytvorí nový centroid s týmito vlastnosťami. Následne sú elementy opäť prerozdelené medzi klastre. Toto sa opakuje až dovtedy, kým nenastane situácia, že by nejaký element zmenil svoj klaster alebo prebehne 1000 iterácií. Keď nastane jedna z uvedených situácií, tak klastrovanie končí a ako výsledok teda dostávame anomálny a neanomálny klaster. Anomálny klaster vieme rozoznať podľa toho, že pre zdrojové ip adresy v ňom je typické, že komunikovali v priemere s malým množstvom cieľových ip adries a portov, avšak z veľa zdrojových portov a vo veľa tokoch.

Predtým, ako vyhodnotíme výsledky samotného algoritmu, sa ešte pozrieme na neanomálny klaster. Aj keď sme klastrovaním od seba oddelili anomálne a neanomálne ip adresy. Môže sa stať, že nejaké anomálie sa nachádzajú aj v neanomálnom klastri. Vytvoríme interval vzdialeností elementov od centra klastra, kde začiatok intervalu bude predstavovať minimum zo všetkých vzdialeností elementov od centra klastra a koniec intervalu bude predstavovať maximum daných vzdialeností. Tento interval ďalej rozdelíme na 5 podintervalov s rovnakou veľkosťou a následne všetky elementy klastra priradíme do jedného z vytvorených podintervalov na základe jeho vzdialenosti od centra klastra. Všetky elementy, ktoré sa nenachádzajú v prvom podintervale, sú tiež označené za anomálie.

2.3 Výsledky a vyhodnotenie algoritmu

Výsledkom algoritmu sú údaje o anomálnych ip adresách (z koľkých rôznych zdrojových portov sa komunikovalo na koľko rôznych cieľových adries, v koľkých rôznych záznamoch o toku sa vyskytovala daná ip adresa a podobne). Ďalej samotný výpis záznamov, ktoré boli označené za anomálne. Následne program vypíše počet záznamov, ktoré boli v skutočnosti botnet a počet, ktoré náš algoritmus odhalil. A nakoniec počítadlá pre:

- TP (true positive/skutočne pozitívny)
- FP (false positive/falošne pozitívny)
- TN (true negative/skutočne negatívny)
- FN (false negative/falošne negatívny)

a vypíše sa celková úspešnosť algoritmu. Na zistenie týchto údajov využívame to, že dataset CTU-13 je labeled dataset, čo bolo spomenuté na začiatku kapitoly 2. To znamená, že po behu algoritmu sa môžeme pozrieť, ktoré záznamy boli v skutočnosti botnet. Výsledky algoritmu možno vidieť na Obrázku 2. Išlo o 10. scenár datasetu, typ botnetu bol Rbot. Tu sme sa dopracovali dokonca k 100% úspešnosti. Všetky záznamy, ktoré náš algoritmus označil za botnet, botnetom aj boli a zároveň všetky záznamy, ktoré označil za normálnu prevádzku, boli aj v skutočnosti normálnou prevádzkou. Ďalej na Obrázku 3 sa nachádzajú výsledky algoritmu na 13. scenári datasetu. Tu náš algoritmus síce odhalil všetku botnet prevádzku, ale zároveň označil za botnet aj prevádzku, ktorá v skutočnosti botnetom nebola. Pri scenári č.8 na Obrázku 4 náš algoritmus chybne označil 108 záznamov ako normálnu prevádzku, aj keď bola v skutočnosti botnetom.

```
odhaleny botnet: 106506
skutocny botnet: 106506
------------------------
TP: 106506
TN: 2015694
FP: 0
FN: 0
presnost: 1.0
```

Obrázok 2. Výsledok nášho algoritmu pri spustení na scenári č.10 datasetu CTU-13.

```
odhaleny botnet: 47526
skutocny botnet: 47526
------------------------------
TP: 47526
TN: 3138834
FP: 211924
FN: 0
presnost: 0.9376379372648078
```

Obrázok 3. Výsledok nášho algoritmu pri spustení na scenári č.13 datasetu CTU-13.

```
odhaleny botnet: 6328
skutocny botnet: 6436
-----------------------------
TP: 6328
TN: 5133410
FP: 100844
FN: 108
presnost: 0.9807368876999021
```

Obrázok 4. Výsledok nášho algoritmu pri spustení na scenári č.8 datasetu CTU-13.

3 Záver

Naštudovali sme si problematiku podobných prác a ako v nich riešili podobné problémy. Implementovali sme algoritmus na detekciu botnet správania zo záznamov sieťových prevádzkových údajov netflow. Úspešnosť nášho algoritmu sa momentálne pohybuje na úrovni 90-100%. Pri opakovaných spusteniach algoritmu sme porozovali príležitostne nepresné výsledky. Tento problém je spojený s náhodným zvolením centroidov. Ak sú zvolené „zle“, to je vtedy, keď majú oba klastre na začiatku približne rovnaký počet prvkov, tak presnosť algoritmu môže výrazne klesnúť. V matematike ale existujú metódy ako vždy zvoliť centroidy správne, čomu sa chceme venovať v práci neskôr.

Do budúcna by sme chceli doladiť niektoré detaily programu a zároveň skúsiť ešte vylepšiť algoritmus, aby sme dosiahli ešte vyššiu úspešnosť. Využiť k-medoids klastrovanie a porovnať jeho výsledky s k-means klastrovaním. Nakoniec by sme chceli skúsiť spustiť náš algoritmus na dátach zozbieraných z univerzitnej siete.

Literatúra

1. D.S. Terzi, R. Terzi, S. Sagiroglu, Big data analytics for network anomaly detection from netflow data, in: International Conference on Computer Science and Engineering (UBMK), IEEE, 2017, pp. 592–597
2. R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, “Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX,” IEEE Communications Surveys and Tutorials, vol. 16, no. 4, pp. 2037 – 2064, 2014.
3. L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, “Disclosure: Detecting botnet command and control servers through large-scale netflow analysis,” in Proc. 28th Annu. Comput. Secur. Appl. Conf. (ACSAC’12), Orlando, FL, USA, Dec. 3–7, 2012, pp. 129–138.