OPTIMIZATION OF ENTROPY-BASED METHOD FOR NETWORK ANOMALY DETECTION

Václav Bartoš
Master Degree Programme (2), FIT BUT
E-mail: xbarto11@stud.fit.vutbr.cz

Supervised by: Martin Žádník
E-mail: izadnik@fit.vutbr.cz

Abstract: This work proposes a speed optimization of entropy-based method for network anomaly detection. This optimization is based on usage of a hash function to reduce size of a histogram, that is needed to compute entropy. Presented results of experiments show, that detection abilities of the algorithm are affected only minimally by this approach but it provides a significant speedup.

Keywords: anomaly detection, entropy, optimization, hash function

1 ÚVOD

Na počítačových sítích stále přibývá množství různých útoků a jiných nežádoucích činností. Zajištění bezpečnosti sítě je dnes velmi složitá úloha. Jednou z metod, které v jejím zajišťování mohou pomáhat, je *detekce anomálií*.

Narozdíl od běžně používaných systémů pro detekci signatur (vzorů), jako je například program SNORT, které v síťovém provozu vyhledávají vzory specifické pro konkrétní známé útoky, systémy pro detekci anomálií vytvářejí model normálního chování sítě a každá významnější odchylka od tohoto modelu je označena jako anomálie a tedy pravděpodobně útok. Tato oblast je však stále spíše předmětem výzkumu a v praxi jsou tyto systémy zatím nasazovány jen vyjímečně.

2 DETEKCE ANOMÁLIÍ ZALOŽENÁ NA VÝPOČTU ENTROPIE

Pro detekci anomálií v síťovém provozu bylo navrženo množství metod založených na nejrůznějších principech. Tato práce se zabývá optimalizací metody, kterou navrhli Lakhina et al. v článku [1]. Tato metoda je založena na předpokladu, že útočná aktivita vyvolá v síťovém provozu změny v rozložení hodnot některých položek z hlaviček paketů, konkrétně zdrojových a cílových IP adres a portů. Tyto změny jsou dobře patrné na histogramu četnosti výskytů jednotlivých hodnot.

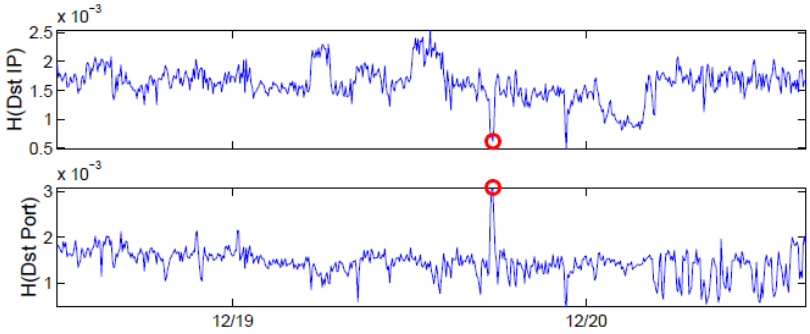
Například skenování portů jednoho počítače ovlivní distribuci cílových adres a cílových portů. Histogram cílových adres bude koncentrován na jednu konkrétní adresu, zatímco histogram cílových portů bude více rozptýlen (distribuce bude rovnoměrnější). Různé typy útoků způsobují různé kombinace koncentrace či ropztýlení distribuce některých položek z hlaviček paketů. Přehled útoků a jimi způsobovaných změn lze nalézt v originálním článku.

Pomocí analýzy distribuce položek z hlaviček paketů lze tedy poměrně spolehlivě detekovat přinejmenším některé typy útoků. Distribuce položek je ale mnohorozměrný objekt a je obtížné s ní pracovat přímo. Naštěstí to však není nutné, většinou nám stačí vědět, zda byla distribuce rozptýlena či koncentrována. Vhodnou mírou, která dokáže tyto změny jednoduše vyjádřit jednou hodnotou, je *entropie*. Ta je pro určitý vzorek dat definována takto:

$$H(X) = -\sum_{i=1}^{N} \frac{n_i}{S} \log_2 \frac{n_i}{S}$$

kde N označuje počet hodnot, kterých mohou data nabývat, n_i označuje počet výskytů hodnoty i ve vzorku, a S je celkový počet hodnot ve vzorku. Pokud je distribuce některé položky koncentrována, entropie této položky se sníží, pokud je rozptýlena, entropie se naopak zvýší.

Když data o síťovém provozu rozdělíme do časových intervalů a v každém takovém intervalu vypočítáme entropii jednotlivých položek, dostaneme průběhy entropie v čase. Různé anomálie se pak v takových průbězích projevují kombinací prudkého zvýšení či snížení entropie současně u několika položek. Příklad na obrázku 1 ukazuje výrazné krátkodobé snížení entropie cílových adres současně s výrazným zvýšením entropie cílových portů, což odpovídá skenování portů.



Obrázek 1: Časové průběhy entropie cílových adres a portů. Červeně je vyznačen okamžik, kdy probíhalo skenování portů. [1]

3 NAVRŽENÁ OPTIMALIZACE

Výpočet entropie sestává ze dvou částí. Nejprve se musí vytvořit histogram, tj. nalézt počet výskytů každé hodnoty, a z těchto dat se poté podle výše uvedeného vzorce vypočítá samotná entropie. Implementace tohoto algoritmu ukázala, že část vytváření histogramu zabírá více než 90% času celého výpočtu.

Pro uložení histogramu některých položek totiž nelze použít obyčené pole, protože např. pro histogram IP adres by bylo zapotřebí 2^{32} položek (přestože většina by obsahovala nulu), což je z hlediska paměťové náročnosti jen velmi obtížně realizovatelné (a v případě IPv6 zcela nemožné). Je proto nutné využít jinou strukturu, například nějakou implementaci asociativního pole, která je však vždy pomalejší.

Aby bylo možno použít pro ukládání histogramu pole, navrhuji dlouhé položky transformovat pomocí jednoduché hash funkce na hodnoty kratší (cca do 16 bitů), jejichž histogram ve formě pole bude potřebovat výrazně méně paměti. Při hashování pochopitelně mohou vznikat kolize a více různých vstupních hodnot tak může být započítáno do jedné položky histogramu. Jak ale ukazují výsledky experimentů uvedené dále, pokud není počet bitů výstupu hash funkce příliš malý, nemají tyto kolize na detekční schopnosti algoritmu výrazný vliv.

4 VÝSLEDKY

Výše uvedený algoritmus byl implemenován v jazyce C++ a byl otestován na datech z univerzitní sítě. Útoky byly uměle generované a vkládané do těchto dat. Data byla rozdělena do pětiminutových intervalů a v každém intervalu byla vypočítána entropie zdrojových a cílových adres a portů. Pro každou dimenzi byla vypočítána průměrná hodnota entropie, její směrodatná odchylka a velikost změny entropie způsobené skenováním portů. V tabulce 1 je pro různé bitové šířky výstupu hash funkce uveden poměr této změny ku směrodatné odchylce, což je hodnota vyjadřující, jak snadno je útok rozlišitelný od běžného kolísání entropie.

Varianta výpočtu	SrcIP	DstIP	SrcPort	DstPort
Původní verze	4,188	13,026	2,516	1,425
16-bit hash	4,382	14,060	2,259	1,640
12-bit hash	4,382	16,808	2,025	1,563
8-bit hash	4,382	26,618	1,817	1,323
4-bit hash	3,001	37,115	1,291	1,204

Tabulka 1: Poměr změny entropie způsobené skenováním portu a směrodatné odchylky entropie.

Z dat v tabulce je vidět, že při použití dostatečného počtu bitů na výstupu hash funkce je anomálie přibližně stejně rozlišitelná, jako v původní verzi bez hash funkce. Při použití nižšího počtu bitů může v některých dimenzích rozlišitelnost dokonce výrazně stoupat. K tomu ale dochází jen tehdy, pokud daný typ útoku v této dimenzi způsobuje koncetraci histogramu k jedné hodnotě. V případě rozptylování histogramu je při použití malého počtu bitů změna způsobená útokem příliš malá.

Pro detekci některých specifických útoků, projevujících se pouze koncentrací histogramu, může tedy stačit i velmi malý počet bitů. Pro detekci ostatních útoků je sice potřeba bitů více, ale stále dost málo na to, aby to znamenalo výrazné urychlení výpočtu a ušetření paměti.

Dále je vhodné zmínit, že počet bitů hash funkce může být různý pro každou dimenzi. Dostatečný počet bitů lze přibližně určit jako průměrnou hodnotu entropie v dané dimenzi.

Tabulka 2 ukazuje, jaký má navržená optimalizace a velikost výstupu hash funkce vliv na rychlost výpočtu. Je vidět, že zavedením hashování lze algoritmus mnohonásobně zrychlit, přičemž šířka výstupu hash funkce již toto zrychlení příliš neovlivní. Snaha snižovat počet bitů hash funkce má tedy význam především z hlediska spotřebované paměti.

Verze algoritmu	Vytvoření histogramu [s]	Výpočet entropie [s]	Zrychlení
Původní verze	3,0433	0,2330	–
16-bit hash	0,1895	0,0134	16,15×
12-bit hash	0,1999	0,0015	16,27×
8-bit hash	0,2011	0,0001	16,28×
4-bit hash	0,2012	0,0000	16,28×

Tabulka 2: Doba výpočtu různých variant a zrychlení oproti veerzi bez hashování.

5 ZÁVĚR

V této práci byla navržena optimalizace výpočtu entropie pro algoritmus detekce anomálií v síťovém provozu navržený v [1]. Optimalizace spočívá v použití hash funkce ke snížení počtu položek histogramu potřebného pro výpočet entropie. Bylo ukázáno, že pokud použijeme dostatečnou šířku výstupu hash funkce, ovlivňují kolize vznikající při hashování výsledky algoritmu jen minimálně. Přitom tato optimalizace dokáže zrychlit výpočet algoritmu až šestnáctinásobně.

REFERENCE

[1] Lakhina A., Crovella, M., Diot, C.: Mining Anomalies Using Traffic Feature Distributions. ACM SIGCOMM Computer Communication Review, ročník 35, č. 4, říjen 2005.