COMPARISON BETWEEN DIVERGENCE MEASURES FOR ANOMALY DETECTION OF MOBILE AGENTS IN IP NETWORKS

Jean Tajer, Mo Adda and Benjamin Aziz

University Of Portmouth, School Of Computing, Portsmouth, United Kingdom

ABSTRACT

This paper deals with detection of SYN flooding attacks which are the most common type of attacks in a Mobile Agent World. We propose a new framework for the detection of flooding attacks by integrating Divergence measures over Sketch data structure. We compare three divergence measures (Hellinger Distance, Chi-square and Power divergence) to analyze their detection accuracy. The performance of the proposed framework is investigated in terms of detection probability and false alarm ratio. We focus on tuning the parameter of Divergence Measures to optimize the performance. We conduct performance analysis over publicly available real IP traces, in Mobile Agent Network, integrated with flooding attacks. Our experimental results show that Power Divergence outperforms Chi-square divergence and Hellinger distance in network anomalies detection in terms of detection and false alarm.

KEYWORDS

Mobile Agents, SYN flooding, Hellinger Distance, Chi-square, Power Divergence, Sketch Technique, IP Networks

1.INTRODUCTION

Multi-Agent Systems (MAS) are designed using independent, autonomous known as agents which can perform their tasks independently or collectively in different types of environments [2]. The agents can be considered as processes with the ability to perform an action on the environment on behalf of user [32]. These systems allow distribution of complex tasks amongst agents. One of the basic properties of multi-agent system is its ability of self-organization which makes it utterly desirable for autonomous and flexible system designs such as graphical applications, logistics, transportation, search engines, network management etc [33].

Mobile Agent Systems can be divided based into programming language by which they are developed and use: Java and non-Java based. Around 85% of Mobile Agent systems available today are built using Java, due to its inherent support to Mobile Agent programming [19].

Mobile Agents are becoming a focus of modern research because of their applications in distributed systems which are replacing traditional client-server architectures rapidly [34]. However, one of the key concerns in practical implementation of Mobile Agent is the lack of protection against any threats.

The rest of this paper is organized as follows. Related work is provided in Section 2. Section 3 provides the security issues that a Mobile Agent can counter while visiting another host in the network.

DOI: 10.5121/ijwmn.2017.9301

We will discuss Sketch data structure to provide grained analysis and to derive probability distributions and will introduce different divergence measures (Hellinger Distance, Chi-square and Power divergences) in order to compare their performance if a flooding attack happens on a Mobile Agent Network, in Section 4. Section 5 describes our proposed approach design. In Section 6, we present our experimental works and check the capability, reaction and performance of the mobile agents based on the developed design. Finally in Section 7, we present the conclusion and our future work.

2.RELATED WORK

From one side several researches have been proposed security solutions to detect and prevent attacks in real traffic. Most of these proposed solutions emphasize on many different detection and prevention strategies.

SYN flooding attack detection has been an interested issue for security researchers. The authors in [2] present the effects of correlation analysis on the DDoS detection. They propose a covariance analysis method for detecting SYN flooding attacks.

Existing methods for anomaly detection are based on different techniques, such as Haar-wavelet analysis [3], [4], Entropy based method [5] and Holt-Winters [6] seasonal forecasting method. Authors in [7] compare two different algorithms (CUSUM and adaptive threshold) for the detection of SYN flooding attack. They conclude that CUSUM performs better then adaptive threshold in terms of detection accuracy of low intensity attacks. However, both of these algorithms face problems of false alarm ratio under normal IP traffic variation.

Other work aggregates the whole traffic in one time series, and applies a change point detection algorithm to detect the instant of anomaly occurrence. The latter has a good performance in terms of spatial and temporal complexities, but presents the drawback of aggregating all traffic in one flow, where low intensity attacks cannot be detected. Furthermore, these methods use static threshold for detecting anomalies, which is not adequate with traffic variations, and may induce false alarm and miss detection.

Sketch data structure uses the random aggregation for more grained analysis than aggregating the whole traffic in one time series. It has been used to summarize monitored traffic in a fixed memory, and to provide scalable input for time series analysis. Authors in [8] propose the use of CUmulative SUM (CUSUM) over the sketch for network anomaly detection. Furthermore, they propose a new mechanism for Sketch inversion and malicious flows identification. We will exploit the Sketch data structure to derive probability distributions.

In addition, recent work experiments the histogram-based detector in order to detect the anomaly behaviors and changes in traffic distributions [9]. They apply Kullback-Leibler divergence between the current and previous measurement distributions.

Authors in [10] apply Hellinger distance (HD) on Sketch data structure, in order to detect divergence between current and previous distributions of the number of SIP INVITE request. In fact, HD must be near zero when probability distributions are similar, and it increases up to one whenever the distributions diverges (e.g. under Invite flooding attacks). In addition, they used the dynamic threshold proposed in [11] during their experimental analysis.

From other side, several researches have been conducted over mobile Agents.Some Articles showed what exactly it is makes Java such a powerful tool for mobile agent development, also it highlighted some shortcomings in Java language systems that have implications for the

conceptual design and use of Java-based mobile agent systems [19],[20].Some studies concentrate their work on the fault tolerance techniques in mobile agents, network management applications based on mobile agent technologies and how the fault tolerance techniques can improve their performance [25], [26].
Other articles worked on an agent-based intelligent mobile assistant for supporting users prior to and during the execution of their tasks [27].
In addition, some works have been performed to integrate the mobile agents with the e-commerce. Some technical relevant issues are well presented [28], [29], [30].

Some researches concentrated their work on security concerns (i.e masquerading, denial of service, unauthorized access and repudiation) of mobile agents and how to protect them by several techniques like for example providing logical framework designed to support large-scale heterogeneous mobile agent applications, on safe code interpretation, digital signatures, path histories, State Appraisal and Proof-Carrying Code (PCC) [21], [22],[23],[24]

Our research combines the mobile agents and the detection methods. It emphasizes on how a mobile agent can detect a flooding attacks. We develop a general framework that increases the detection accuracy and reduces the false alarm by integrating different divergence measures over Sketch technique in a Mobile Agent world.

3.MOBILE AGENTS SECURITY THREATS AND COUNTERMEASURES

Security is one of the key factors of MAS. In fact, a MA is one of the potential threats to computer systems and vice versa, from the host system to the MAS itself. In this part, we will talk about the main security issues related to MAS.

The security threats for MASs could be divided as follows:

- IP spoofing: consists of sending packets with a faked IP source address. The server should believe that the packets come from another host, probably a host that is allowed to establish connections with the attacked host, if the real one is not allowed
- Sniffing: it is the observation and analysis of network traffic in order to obtain relevant information (such as IP addresses and host functionalities) to perform other attacks.
- UDP flood attack: this kind of flooding attack consist on sending many UDP packets to different port of a target in random way. This target will check if there's any application on the relevant port, if not, he will be occupied to send ICMP replies and can't treat requests from legitimate clients.
- SYN flood attack: it consist on sending many TCP connection requests to a target. This latter will accept the establishment of the connection and notify the client. Except that, this one will never use them. Thereby, the server will be drown by unused connections and, eventually, will not reply to legitimate users requests.

There are many security services that can be used for securing the agents systems, for example; authentication, integrity, confidentiality and authorization.

In case of the authentication, the host needs to know the sender of the delivered agent. The agent authentication process includes verifying the entity that programmed the agent and also verifying the entity that dispatched it to the host. Basically, the agent and the host need to know with whom they are talking and dealing with, here the public-key encryption or passwords can be used,.
For integrity, checking the integrity of the agents is a technique that makes sure no one has made any changes to the agents, the agents travelling form on host to another, and communicates and exchanges their data with other hosts and other agents. In this case, we need to make sure that the

agents have not been tampered with in relation to their state, code or data. Moreover, the agents could carry different types of data, for example some private data. These data should only be readable from a specific host or agents. This technique is very important to avoid an eavesdropping threat. The last service which helps to protect the agents and the hosts is authorization; the incoming agents should have a specific right to access the host information, so different agents have different authority, to protect the hosts and also to protect themselves.

4.THEORITICAL BACKGROUND

4.1. Sketch technique

In this section, we review the K-ary Sketch data structure. Using Sketch data structure makes our framework flexible and scalable for grained analysis. No matter how many flows exist in the traffic, Sketch generates fixed-number of time series [3], [4] for anomaly detection. Sketch provides more grained analysis than aggregating whole traffic in one time series.

The Sketch data structure is used for dimensionality reduction. It is based on random aggregation of traffic attribute (e.g. number of packets) in different hash tables. A Sketch S is a 2D array of H × K cell (as shown in Figure 1), where K is the size of the hash table, and H is the number of mutual independent hash functions (universal hash functions). Each item is identified by a key κn and associated with a reward value νn. For each new arriving item (κn, νn), the associated value will be added to the cell S[i][j], where i is an index used to represent the hash function associated with ith hash table ($0 \leq i \leq d - 1$), and j is the hash value (j = hi(κn)) of the key by the ith hash function.

Data items, whose keys are hashed to the same value, will be aggregated in the same cell in the hash table, and their values will be added up to existing counter in the hash table. Each hash table (or each row) is used to derive probability distribution as the ratio of the counter in each cell to the sum of whole cell in the line. The derived probability distributions (we get K probability set, one per line) are used as inputs for divergence measures

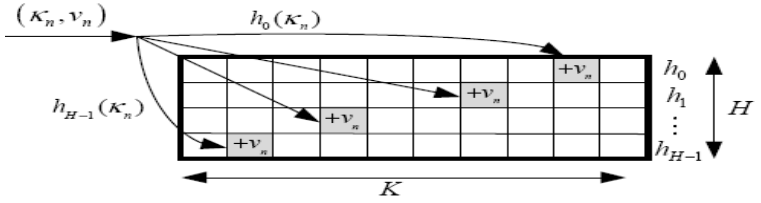

Figure 1: Sketch Data structure

4.2 Divergence Measures

These measures are used to detect the DDoS attacks based on the deviation of traffic distribution. In fact, the idea is to compare the prior distribution derived from Sketch counters in previous time slot, with the currently obtained distribution. One can use this change to detect flooding attack, because the counter of one cell will increase significantly with the number of sent requests, and the probability distribution deviates at the start and stop instants of the flooding attack.

4.2.1. Hellinger Distance (HD)

Hellinger Distance (HD) is used to measure the divergence between two sets of probability values.
For two discrete probability distributions P = (P0, P1….,Pk−1) and Q = (Q0, Q1,…,Qk−1), with Pi ≥ 0, Qi ≥ 0 and

$$\sum_{i=0}^{k-1} pi = \sum_{i=0}^{k-1} qi = 1$$

The HD between current distribution P and prior distribution Q is defined as:

$$HD(P,Q) = \frac{1}{2}\sum_{i=0}^{k-1}(\sqrt{pi} - \sqrt{qi})^2 \quad (1)$$

Where HD satisfies the inequality 0 ≤ HD (P, Q) ≤ 1, and HD (P, Q) = 0 if P = Q. HD is a symmetric distance (e.g. HD (P, Q) = HD (Q, P)), and induces two spikes, one at the beginning of change, and the second at the end of the change, [18].

4.2.2. Chi-square divergence

χ2 divergence is used to measure distance between two discrete probability distributions (P and Q). For 2 probability sets P = (p1, p2, p3. . . pn) and Q = (q1, q2, q3. . . qn), with Pi ≥ 0, Qi ≥ 0 & $\sum_{i=0}^{k-1} pi = \sum_{i=0}^{k-1} qi = 1$

The Pearson χ2 divergence between P and Q is given by:

$$x^2(P||Q) = \sum_{i=1}^{n} \frac{(Pi-Qi)^2}{Qi} \quad (2)$$

Where Q is the estimated probability distribution and P is the measured probability distribution, and χ2 (P||Q) is the distance between distributions P and Q.

For hypothesis testing, such as H0 (normal traffic hypothesis) and H1 (traffic with anomalies), χ2 values can run from zero into infinity. χ2 will be zero if P and Q are identical (Pi = Qi) under hypothesis H0, and χ2 increases as the distributions become dissimilar, and eventually so high (infinity) when the two distributions are independent (P ≠ Q)under hypothesis H1. It is important to note that χ2 divergence is nonnegative and the division 0/0 is treated as 0, and the division by zero is replaced by a very small value ε .

The χ2 divergence between 2 probability distributions P and Q must be near zero under normal traffic, with a large deviation (one spike) when distributions change occurs. χ2 is asymmetric (χ2 (P||Q) ≠ χ2 (Q||P)), and its symmetric version raises two spikes. One spike at the beginning and the second at the end of the attack.

$$x^2(P||Q) + x^2(Q||P) = \sum_{i=1}^{n} \frac{(Pi-Qi)^2 + (Pi+Qi)}{Pi*Qi} \quad (3)$$

We intend to use Pearson chi-square divergence (asymmetric) to detect anomaly through the detection of deviations from normal traffic profile, and we will modify the input time series to constrain χ2 to raise alarms (spikes) for the whole duration of attack. In [30], authors prove that χ2 divergence behaves better than all classical divergences (Hellinger distance, Kullback-Leibler, Likelihood, etc, [6].

4.2.3. Power divergence (PD)

The Power Divergence has been first defined in [7] and equivalent variants (up to a scale factor β) of this divergence are discussed in [35], [36]. The divergence measure is therefore the decision measure that generalizes the Kullback-Leibler measure and Hellinger distance to a broad class of divergence of order β. In fact, the Power Divergence is a measure of distance between two probability measure of order β given as follows :

$$PD\,(P||Q) = \frac{\sum_{i=1}^{W} Pi\,(Pi/Qi)^{\beta-1}-1}{\beta(\beta-1)} \qquad (4)$$

where P is the posterior probability distribution and Q is the prior probability distribution. This divergence presents some interesting special cases. For β = 0.5, this divergence is 4 x HD (P || Q) [36], and for β = 2 it is equal to 0.5 x X2 (P || Q) divergence. Obviously, this power divergence outperforms then the x^2 and HD measures (some results will be provided later in the paper).

In fact, by changing the values of β, one can optimize the detection of attacks compared to the x^2 and HD measures.

In our experiments described later, we will show numerically that for different values of β, the detection efficiency changes. The optimal value of _β can then be obtained.

P and Q are derived from the Sketch data structure in two consecutive discrete intervals. Firstly, the shared counters of the sketch are continuously updated from ongoing traffic during a time interval T. At the end of each interval, the probability Pi,j is calculated as the ratio of each counter to the sum of the whole counters in one hash table:

$$Pi,j = \frac{s[i][j]}{\sum_{i=1}^{W} s[i][j]} \qquad (5)$$

We obtain d probability distributions in each interval (P1 ……Pd), where Pi is the distribution (P1, ….Pd) resulted from the i^{th} hash table. Qi is the probability distributions resulted from previous interval. The probability distributions of Qi is calculated in the same manner as Pi (Eq. 5).

When the Power Divergence is larger than dynamically updated threshold, we raise an alarm. However, Power Divergence induces only two spikes (at the start and at the end of attack). As we want to continuously raise alarms for whole duration of the attack, the distribution Qi will stop sliding by keeping its value until the end of the attack. However, with the variations of normal traffic, and the similarity of DDoS attacks with flash crowd, we suppose that flooding attacks will span for many intervals, in contrast to flash crowd and normal variation. Thus reduce the false alarms. Therefore, we will trigger an alarm only if the deviation lasts more than Δ intervals.

5.PROPOSED APPROACH

The proposed approach for anomaly detection in Mobile Agent networks is based on Sketch and divergence measures (Hellinger Distance. Power Divergence and Chi-square)

The detection system records the number of monitored point (e.g. #packets, #SYN, #flows, etc.) in the Sketch for each discrete time interval T. Random aggregation of traffic flows in Sketch is

the first step of our processing, followed by time series forecasting with divergence measures (Figure. 2).

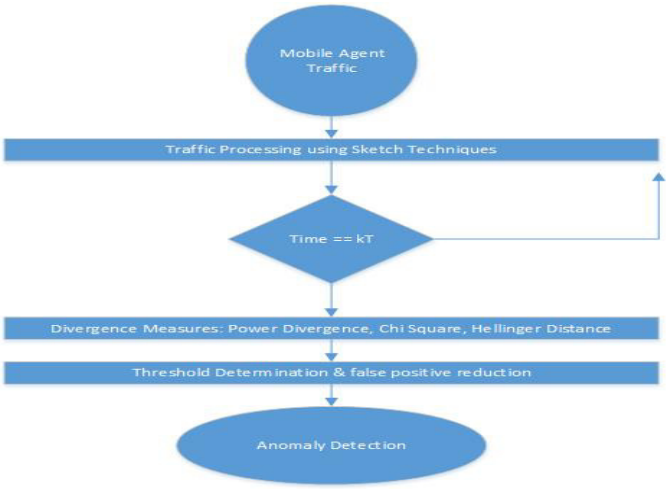

Figure 2 : Architecture of the proposed approach for network anomaly detection.

During each interval, the destination IP address (DIP), for each packet containing a SYN segment, is hashed by H hash functions. The resulted hash value by the ith function (j = hi(DIP)) is used as index of the associated counter Si,j with DIP. Each arriving SYN segment increments the associated counter.

Our analysis will be focused on TCP SYN flooding by counting the number of SYN. At the end of each epoch T, we derive probability distributions from Sketch. First, we get the sum of the counter in each line, and the probability pi,j in each cell is calculated as the ratio of each counter to the total number of SYN:

$$Pi,j = Si,j.Counter \sum_{j=0}^{k-1} Si,j.Counter \quad (6)$$

Each cell Si,j becomes a data structure, that contains: current counter, current and previous probabilities. Therefore, each line (or hash table) provides two probability distributions: the first one is from previous interval and used as reference distribution Qi. The second one is from current interval Pi, and used to measure the divergence from the reference distribution, in order to detect anomalies. Divergence measures between the current (Pi) and reference probability (Qi) distributions is calculated for each line in the Sketch, at the end of each time interval (i.e. at n.T). During malicious activities, the divergence measure D(Pi||Qi) produces spikes, and when more than L (L < H) divergences resulted from different hash tables exceed a dynamic threshold, an alarm is raised.

To detect deviations in the time series resulted from divergence measures, we derive a subsequent time series containing the values of D(Pi||Qi) without spikes. In this last time series (without large values), we define a dynamic bound of μi+ασi. Significant deviations are larger than the dynamic bound:

$$D\,(Pi||Qi) > \mu i + \alpha\sigma i \quad (7)$$

Where D(Pi||Qi) is the divergence measure in the time interval n.T for the ith line in the Sketch, and μi & σi are the mean and the standard deviation respectively of smoothed time series that

doesn't contain spikes (D^(Pi||Qi)). μi and σi are updated dynamically using the Exponentially Weighted Moving Average (EWMA):

$$\mu i = \beta\mu(i-1) + (1-\beta)\, D^{\wedge}(Pi||Qi) \quad (8)$$

$$\sigma i^2 = \beta\sigma(i-1)^2 + (1-\beta)\left(D^{\wedge}(Pi||Qi) - \mu i\right)^2 \quad (9)$$

The threshold is updated dynamically with the value of μi and σi as shown in above equations. α is a parameter used for calibrating the sensitivity of the detection algorithm to variations. It is also used to reduce the false alarm rate. Under normal traffic, divergence D (Pi||Qi) falls inside the bound of μi + 2σi. When D (Pi||Qi) exceeds the dynamically updated threshold over L lines, an alarm is triggered.

6.EXPERIMENTAL WORKS

In this section, we present the performance analysis results for integrating divergence measures over Sketch, for detecting SYN flooding attacks in a mobile agent network. As we want to compare 3 divergence measures (HD, PD & χ2) over Sketch for the detection of flooding attacks, we will implement a mobile agent network.

For the sake of simplicity, we focus our analysis on the detection of TCP SYN flooding attacks, as it is the widely used attack for DDoS in these days.

6.1 DATASET

The following techniques and tools are used: Two workstations with 8 GB and 768 MB of RAM respectively, which run Windows Server 2003 and a number of Mobile Agents are used.
We have considered the above describe mobile agents will have to execute the similar path. To measure the capability of the proposal towards eavesdropping threat, a test environment is set up using the above mentioned computers as shown in Figure 3. Computer A is considered to act as trusted server (TS) and computer B runs many host nodes simulated through various port numbers as well as the home node in a virtualized mode. Ethereal will be running regularly over computer A. its job is to collect packets in the mobile agent network and store them for a period of 4h00 from 18/02/2017 07h30 to 11h30. These traces are used to test the efficiency of divergence measures. IP addresses in the traces are scrambled by a modified version of tcpdriv tool, but correlation between addresses are conserved. We analyze these 8h30 traces using Sketch data structure, with a key of the Sketch (κn = DIP), and a reward vn = 1 for SYN request only, and vn = 0 otherwise. We set the Sketch width K to 1024, and the number of hash H to 5.

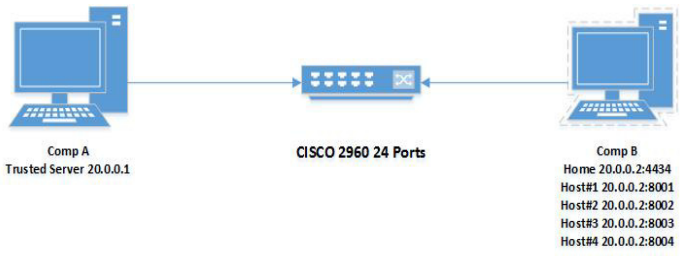

Figure 3. Experimental Lab

Afterward, we inject 12 real distributed SYN flooding attacks with different intensity inside this trace. These attacks are inserted each 30 minutes (on instants t=30, 61, 90, 127,157,187, etc.) and

span for 10 minutes. These different intensity attacks are shown in Figure 4. The first attack begins with a value of 900 SYN/min and decreases until 280 SYN/min.

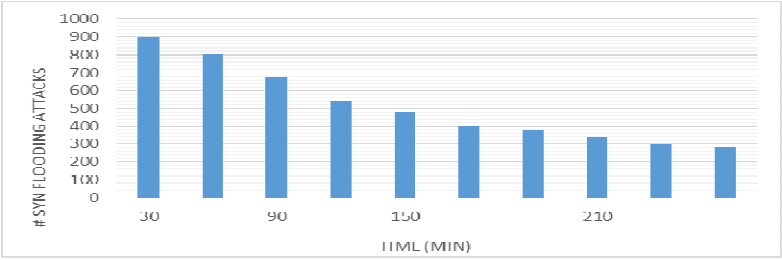

Figure 4. SYN flooding Attacks

Figure 5 & Figure 6 show the variation of total number of mobile agents' packets before and after the injection of SYN flooding attacks. By comparing these variations, we might not notice the differences between both figures without deep inspection. Inserted attacks don't induce heavy deviations in the time series of the total number of SYN requests. This can be explained by the fact that the intensity of SYN flooding attacks is not large compared to the intensity of the total number of SYN segments. In such cases, the detection of attacks is very challenging, because no heavy changes in the time series describing the variations of the total number of SYN, and the intensity of the SYN flooding attacks is buried by the large number of SYN (as shown in Figure 4) before attacks injection.

6.2 Evaluation Strategy

In this section, we present the evaluation results of the application of these divergence measures on the mobile agent IP traces.

First, we begin our analysis by applying HD & $\chi 2$ divergence over the traces (before attacks injection). We set the dynamic threshold as given in Equation. 5. We will begin our analysis by applying the HD and Chi-Square over the mobile agent IP traces (before injection SYN flooding attacks). Figure 7 & Figure 8 show the variation of these 2 divergence algorithms as well as the dynamic threshold (dashed line) before the injection of attacks. When the value of divergence measures is larger than threshold in at least 3 hash tables in the Sketch, an alarm is triggered. We see that both algorithms were able to detect anomalies at different time (t=90, 127,157,180 etc.). These anomalies are temporary and they don't persist more than many minutes. However, there are more anomalies that can be detected by using the source IP address as the key of the Sketch, but we will restrict our analysis to SYN flooding attacks. In fact, after the manual verification of traces, we found that HD triggers 4 false alarms, and the $\chi 2$ divergence achieves very high detection accuracy with 1 false alarm.

Indeed, we continue our analysis by applying the HD and Chi-Square over the mobile agent IP traces (after injection SYN flooding attacks). We noticed that in case of Hellinger Distance using a dynamic threshold, we obtain 4 false alarms with a detection of 100% (Figure 9). However, in the case of Chi-Square, we did not obtain any false alarm (Figure 10). We found through our conducted experiments that Chi-square divergence performs better than HD in terms of reducing false alarm, with less effort for tuning the dynamic threshold. The intensity of raised spikes in Chi-square increases with the intensity of attacks and dynamic threshold becomes useless.

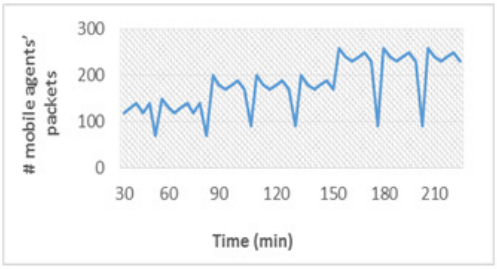

Figure 5. Total number of mobile agents' packets

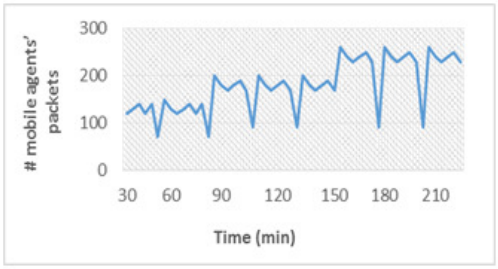

Figure 6. Total number of mobile agents' packets after SYN flooding attacks injection

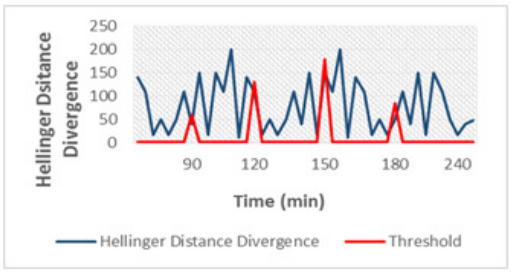

Figure 7. Hellinger Distance before attacks

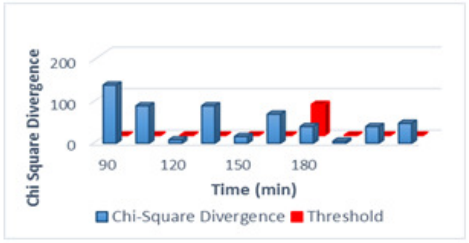

Figure 8. Chi-square before attacks

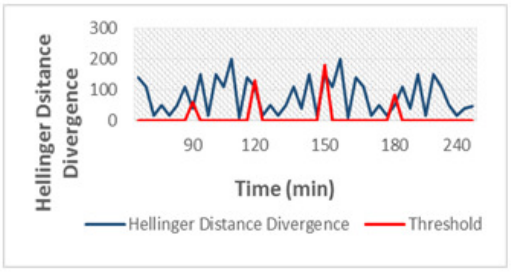

Figure 9. Hellinger Distance after attacks

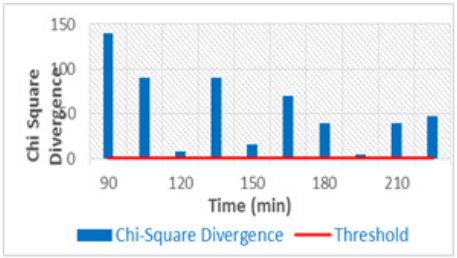

Figure 10. Chi-square after attacks

In this part, we will focus our study on the Power Divergence. Due to space limitation, we provide in this section, the results for only two values of β :0.5 and 1.5.

The value of β = 0.5 makes the Power Divergence (PD) proportional to Hellinger Distance (HD). Figure 11 shows the variation of Power Divergence for β = 0.5 with the dynamic threshold given in Equation 5. Power Divergence is able to detect all the SYN flooding attacks but with 4 false alarms.

For the value of β= 1.5, Figure 12 shows the variation of Power. We can notice that via this value of β, all the attacks have been detected (100%) with only 1 false alarm.

The intensity of spike is proportional to the intensity of the attack. We conclude that the value of β= 1.5 outperforms β = 0.5 in terms of true detection and false alarm rate.

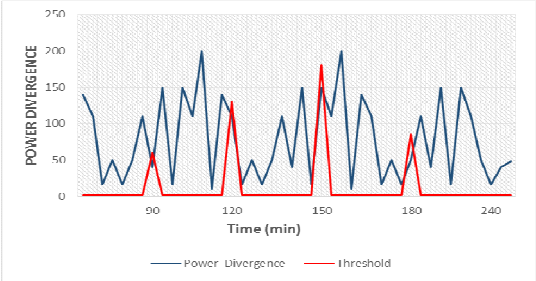

Figure 11. Power Divergence for β=0.5

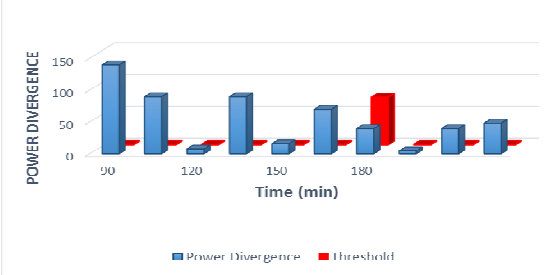

Figure 12. Power Divergence for β= 1.5

7. CONCLUSIONS

In this paper, we analyzed the accuracy of 3 divergence measures (HD, Power Divergence & Chi-square divergence) over Sketch data structure for network anomaly detection. We compared their performances in terms of true positive and false alarm ratio, over real mobile agents IP traces with injected real distributed SYN flooding attacks at known instants.

Afterward, we used dynamic threshold for achieving the best tradeoff between false alarm and true detection. We found that HD performs a good detection, but with higher false alarm ratio than Chi-square divergence. We can conclude that Chi-square conducts better detection than HD for mobile agents' network. Furthermore, the intensity of triggered spikes by Chi-square divergence increases significantly with the intensity of attacks. It is important to note that these divergence measures with Sketch are computationally efficient for handling traffic on mobile agents' traffic.

We showed that Power Divergence presents some interesting special cases. For β = 0.5, this divergence is 4 x HD (P ‖ Q) [36], and for β = 2 it is equal to 0.5 x X2 (P ‖ Q) divergence. Obviously, this power divergence outperforms then the x^2 and HD measures. In fact, by changing the values of β, one can optimize the detection of attacks compared to the x^2 and HD measures. Our experimental results show the capacity of PD in the detection of low intensity attacks. We show the result of PD with β= 0.5 & β= 1.5. We have shown that for β = 1.5, PD outperforms the HD (β = 0.5 which is the HD in this special case). Furthermore, the detection accuracy of PD increases when increasing the value of β.

In our future work, we will focus on providing additional information to pinpoint malicious flows, in order to trigger automatic reaction against ongoing attacks. We also intend to provide a method for reducing the amount of monitoring data on high speed networks, and to analyze the impact of sampling on the precision of this divergence measure.

REFERENCES

[1] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denialof-Service Activity" in Proceedings of USENIX Security Symposium (SSYM'01), 2001, pp. 9–22.

[2] HU,Jiang-Ping,Zhi-XinLIU,Jin-HuanWANG,Lin WANG, Xiao-Ming HU. "Estimation, Intervention and Interaction of Multi-agent Systems." Acta Automatica Sinica 39, no. 11 (2013): 1796-1804.

[3] O. Salem, S. Vaton, and A. Gravey, "A novel approach for anomaly detection over high –speed networks," in Proceedings of the 3rd European Conference on Computer Network Defense (ECND'07), vol. 30, 2009, pp. 49–68.

[4] G. Cormode and S. Muthukrishnan, "An improved data stream summary: The count-min sketch and its applications," J. Algorithms, vol. 55, pp. 29–38, 2004.

[5] J. Tang, Y. Cheng, and C. Zhou, "Sketch-based sip flooding detection using hellinger distance," in Proceedings of the 28th IEEE conference on Global telecommunications (GLOBECOM'09), 2009, pp. 3380–3385.

[6] M. Broniatowski and S. Leorato, "An estimation method for the neyman chi-square divergence with application to test of hypotheses," J. Multivar. Anal., pp. 1409–1436, July 2006.

[7] J. Havrda and F. Chavrat, "Quantification method of classification processes: The concept of structural α-entropy," Kybernetika, vol. 3, pp. 30–35, 1967.

[8] P. N. Rathie and P. Kannappan, "A directed-divergence function of type β," Inform. Contr., vol. 20, pp. 38–45, 1972.

[9] D. Haussler and M. Opper, "Mutual information, metric entropy, and cumulative relative entropy risk," Ann. Statist., vol. 25, pp. 2451–2492, 1997.

[10] "MAWI working group traffic archive," http://mawi.wide.ad.jp/mawi/.

[11] M.Bishop, "Introduction to security network", Addison Wesley, 1 edition, 26 October 2004

[12] VOIP Security and Privacy Threat Taxonomy, public release, 24 October 2005

[13] Mohamed Nassar,Saverio Niccolini,Radu State, Holistic "VOIP Instrusion Detection and Prevention System", ACM SIGCOMM, New York, July 2007.

[14] Mohamed Nassar, Radu State, and Olivier Festor. "Voip Honeypot Architecture". In: Integrated Network Management (IM 2007), pages 109-118. IEEE, Munich, May 2007

[15] V. Jacobson, "Congestion avoidance and control," SIGCOMM Comput. Commun. Rev., vol. 25, pp. 157–187, January 1995.

[16] Tascos Dagiuklzd, Jiri Markl,Michal Rokos, low cost tools for secure and highly available voip communication services, snocer 2

[17] http://www.webbasedconferencing.org/blog/vishing-spiting-eavesdropping-security-threats-to-voip-primer

[18] Hemant Sengar,Duminda Wijesekera, Sushil Jjodia," Detecting VOIP Flodds Using the Hellinger Distance", IEEE, Vol.19, June 2008

[19] Danny B. Lange, Mitsuru Oshima. "Mobile Agents with Java: The Aglet API", September 1998, Volume 1, Issue 3, pp 111–121

[20] Sun: Java 2 SDK security documentation. (2003).

[21] Guido J.van 't Noordende, Frances M. T. Brazier, Andrew S. Tanenbaum. "Security in a Mobile Agent System", 2004, IEEE Symposium on Multi-Agent Security and Survivability

[22] Michelle S. Wangham, Joni da Silva Fraga, Rafael R. Obelheiro. "A Security Scheme for Agent Platforms in Large-Scale Systems", 2013, IFIP International Conference on Communications and Multimedia Security Mobile, pp 104-116

[23] Gray, R., Kotz, D., Cybenko, G., Rus, "Security in a multiplelanguage, mobile agent systems". LNCS 1419. Springer-Verlag (1998)

[24] Karnik, N. "Security in Mobile Agent Systems". PhD thesis, University of Minnesota (1998)

[25] Maria Zubair, Umar Manzoor. "Mobile Agent based Network Management Applications and Fault-Tolerance Mechanisms", The Sixth International Conference on Innovative Computing Technology (INTECH 2016)

[26] Mouhammd Alkasassbeh, Mo Add. "Network fault detection with Wiener filter-based agent", Journal of Network and Computer Applications 32(4) (4):824-833 • July 2009

[27] Talal Rahwan, Tarek Rahwan, Iyad Rahwan, and Ronald Ashri. "Agent-based Support for Mobile Users using AgentSpeak(L)", Agent-Oriented Information Systems Volume 3030 of the series Lecture Notes in Computer Science pp 45-60

[28] Tu, Griffel and Lamersdof. "Integration of intelligent and mobile agent for E-commerce"

[29] Ryszard Kowalczyk, Mihaela Ulieru and Rainer Unland. “Integrating Mobile and Intelligent Agents in Advanced e-Commerce: A Survey)”, Agent-Oriented Information Systems Volume 3030 of the series Lecture Notes in Computer Science pp 45-60
[30] Jansen W. and Karygiannis “T. Mobile Agent Security”, National Institute of Standards and Technology, Gaithersburg, MD 220899.
[31] HU, Jiang-Ping, Zhi-Xin LIU, Jin-Huan WANG "Estimation, Intervention and Interaction of Multi-agent Systems." Acta Automatica Sinica 39, no. 11 (2013): 1796-1804.
[32] Umar Manzoor, Samia Nefti, Yacine Rezgui “Categorization of malicious behaviors using ontology-based cognitive agents”, Data & Knowledge Engineering, Volume 85, May 2013, Pages 40-56.
[33] Umar Manzoor, Samia Nefti, “iDetect: Content Based Monitoring of Complex Networks using Mobile Agents”, Applied Soft Computing, Volume 12, Issue 5, May 2012, Pages 1607-1619.
[34] Chen, Bo, Harry H. Cheng, and Joe Palen. "Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems." Transportation Research Part C: Emerging Technologies 17, no. 1 (2009): 1-10.
[35] P. N. Rathie and P. Kannappan, “A directed-divergence function of type _,” Inform. Contr., vol. 20, Pages 38–45, 1972.
[36] D. Haussler and M. Opper, “Mutual Information, Metric Entropy, and Cumulative Relative Entropy Risk,” Ann. Statist., vol. 25, Pages2451–2492, 1997.

AUTHORS

Jean TAJER is working as Estimation Unit Head – Low Current at Nesma Trading (KSA). He is a PHD student at University of Portsmouth (UK). His research interests are focused on areas related to security, detection of DDOS attacks over a mobile agents network, Sketch techniques, Divergence measures. He gained my MSC in Communication Network Planning and Management from University of Portsmouth in 2007. Another Master had been gained from University of Paris Sud in 2008. He worked previously at Spie Communication (France) as team leader in Network and Unified Collaboration. He gained several ceritificates from Cisco, HPE, Avaya, Juniper. up828996@myport.ac.uk
https://www.linkedin.com/in/jean-tajer-b1957522/

Mo ADDA is a Principal Lecturer at the University of Portsmouth since 2002. He obtained a PhD in distributed systems and parallel processing from the University of Surrey. As a Senior Lecturer, he taught programming, computer architecture and networking for 10 years at the University of Richmond. From 1999-2002, He worked as a senior software engineer developing software and managing projects on simulation and modelling. He have been researching parallel and distributed systems since 1987. His research interests include multithreaded architectures, mobile networks and business process modelling, parallel and distributed processing, wireless networks and sensor networks, network security and mobile intelligent agent technology. mo.adda@port.ac.uk
https://www.linkedin.com/in/mo-adda-1a589516/

Benjamin Aziz is a senior lecturer at the School of Computing, University of Portsmouth. He gained a PhD in Computer Science from Dublin City University in 2003 and since, He has held several post-doctoral research posts in University College Cork, Imperial College London and Rutherford Appleton Laboratory in Oxford. My research in the field of computer and information security spans more than 15 years. In particular, his research interests are focused on areas related to formal analysis of security properties, engineering secure large-scale distributed systems, security requirements at the engineering level, trust management and digital forensic analysis and formalisation. Over the years, he has published over 70 articles, papers, reports and book chapters in these areas. He is a member of several international working groups including the Cloud Computing Security Alliance, ERCIM Formal Methods for Industrial and Critical Systems, ERCIM Security and Trust Management and IFIP WG 11.3 on Data and Application Security and Privacy. He coordinates the Computer Security and Digital Forensics Research Group in the School of Computing. Finally, he is also an Associate Editor-in-Chief of the International Journal of Security (IJS) and an Associate Editor of Wiley’s Security and Communications Networks (SCN). benjamin.aziz@port.ac.uk
https://www.linkedin.com/in/benjamin-aziz-17ba804/