Scan Surveillance in Internet Networks
(Work in Progress)

Khadija Ramah Houerbi[1], Kavé Salamatian[2], and Farouk Kamoun[1]

[1] National School of Computer Science, University of Manouba, Tunisia
khadija.houerbi@cristal.rnu.tn, farouk.kamoun@ensi.rnu.tn
[2] Lancaster University, Lancaster, UK
kave.salamatian@comp.lancs.ac.uk

Abstract. In recent years, many measurement studies have shown the ubiquity of scanning activities in the Internet and the growing sophistication of probing techniques that became more stealthy by stretching slowly over time or using spoofed source IP addresses. Scans are mainly generated by attackers trying to map the configuration of a target network and by computer worms trying to spread over the Internet. Although, the problem of scan detection has been given a lot of attention by network security researchers, current state-of-the-art methods still suffer from high percentage of false alarms or low ratio of scan detection. In this paper, we propose to detect changes in scanning patterns, by monitor variation of the distribution of scan features in a space spanned by IP source address, IP destination address, source port number, and destination port number. This gives insight on characteristics of scanning activities and exposes the presence of emerging scanning attacks and worms. For that, we propose to use an information theoretic-based approach to detect changes in distributions.

Keywords: scan monitoring, anomaly detection, Information Theory, Networks.

1 Introduction

A major challenge for security managers is to develop network intrusion detection systems that can timely and accurately detect network attacks. A precursor to many attacks on networks is often a reconnaissance operation more commonly referred to port scans. Scans have many legitimate uses [1], including, for system administrators, to verify the security of a network, to find web servers to index by some search engines and some application (e.g. SSH, some peer to peer as edonkey and windows applications). However, intruders frequently perform port scanning as a mean to gather valuable information on target victims to achieve destructive attacks. Furthermore, computer worms also use port scanning to propagate; infected hosts usually perform scanning to search for new vulnerable hosts. This scanning activity takes the form of probe packets targeted at specific TCP or UDP ports on multiple target hosts. If a target host responds, then the scanning host will initiate the process of uploading malware onto that host.

L. Fratta et al. (Eds.): NETWORKING 2009, LNCS 5550, pp. 614–625, 2009.

Many measurement studies [2–4], have shown that port scans represent a sizable portion of today's Internet traffic, that exhibits steady growth, mainly since 2001, and a changing nature over time and between sites. In [2], the authors examined the history of scanning activities over 12.5 years (between 1994 and 2006) as observed at the border of the Lawrence Berkeley National Laboratory (LBNL). They showed that since 2001 scanning became an ubiquitous phenomenon (mainly due to the wide spread of Internet worms like CodeRed I and II, Nimda, Slammer, and Blaster). Following a different approach, the authors in [3], measured and characterized traffic amassed by network telescopes in two academic networks and termed it "Background radiation". This traffic, sent to unused IP addresses, is anomalous by nature, and reflects the presence of many unwanted activities ranging from scans, backscatter from a flooding attack victimizing someone else, exploit attempts and mis-configurations. The authors of [4] analyzed a set of logs from the global "DShield" repository that spanned a 4 month-long period during 2001-2002. They showed that while common scanning types (eg. vertical and horizontal) are indeed prevalent, other strategies such as coordinated and stealthy scans are also widely used.

In the next section, we will survey state-of the art of scan techniques and scan detection approaches. We will show that most of scan detection methods need connection reassembly and are limited to one scan type. Moreover, nearly all proposed methods have difficulties catching low rate scanners and often suffer from significant levels of false positives making them inadequate for automatic scan suppression.

Subsequently, we will present in section three, the approach proposed in this paper. Although, the proposed approach is not a scan detection method it aims to expose the presence of emerging scanning attacks and worms. It consists of collecting and analyzing scan traffic to track significant changes in its distributional aspects. Detecting such changes is important because it can be relevant of serious security problems such as rapid scanning worms or reconnaissance scan preceding destructive attacks. The section four exposes our experimental results. For that we used both real measurement datasets collected on Tunisian National University Network and some artificial traces obtained by mixing a real trace with experimental Nmap scan traffic. Finally, we conclude and present some perspectives in section five.

2 Background and Related Work

2.1 Scan Techniques

Port scanning can be defined as a technique for discovering open "doors" or ports on a host or a set of hosts. It aims to discover host's vulnerabilities by sending a probe to a port and listening for an answer. As most of Internet services are on TCP, scanners are mainly interested to probing TCP ports to determine their states. There are at least a dozen scan techniques implemented in Nmap utility which we can group in three major types: SYN scans, non-SYN scans, and idle scans. SYN scan techniques work against any compliant TCP stack and

allow clear, reliable differentiation between the open, closed, and filtered states of a TCP port. They can be performed quickly, scanning thousands of ports per second on a fast network. The Half-open scanning technique (an example of SYN scan technique) is the most popular one. It consists of sending a single SYN packet to a specific port on a target host and then waiting for a response. A SYN/ACK response indicates that the probed port is listening (open), while a RST (reset) response signifies a closed port. If no response is received (or an ICMP unreachable error is received), the port is marked as filtered. Non-SYN scans are more subtle techniques that exploit loophole in the RFC 793 to differentiate between open and closed ports . However, many systems (Microsoft Windows, many Cisco devices, BSDI, and IBM OS/400) do not follow RFC 793 to the letter that makes these techniques less effective. Finally, Idle scan is an advanced scan method that allows for blind TCP port scanning of target hosts (meaning no packets are sent to the target host from the real attacker's IP address). It exploits predictable IP fragmentation ID sequence generation on a zombie intermediate host to glean information about the open ports on the target. IDS systems will display the scan as coming from the zombie machine used. Besides being extraordinarily stealthy (due to its blind nature), it permits mapping out IP-based trust relationships between machines.

2.2 Scan Detection Methods

As a result of importance of scan traffic, various scan detection methods have been proposed to detect and prevent scan activities. These methods can be classified into counting methods [1, 5, 6], and non counting ones [7–9].

Snort [6], a popular intrusion detection system, implements a simple counting method for scan detection. It marks a source IP address as a scanner if it tries to connect to more than a given number of distinct IP addresses or a fixed number of TCP/UDP ports within a time window. However the counting algorithm can erroneously mark legitimate hosts as scanners in particular when proxies or address translation mechanisms are used. For example in [1], the authors established that over a Lawrence Berkley Laboratory (LBL) traces, with Snort's default settings, 38.5% of host detected as scanner are in fact false positives. Other counting methods have built upon the observation that, unlike benign remote users, scanners are more likely to initiate failed connections to local IP addresses. Bro [5] scan detection algorithm uses this and counts only failed connection for services specified in a configurable list. For the others, it counts all connections.However the issue with Bro is similar to Snort, even if misclassification is less likely to happen because of counting only failed attempts. Threshold Random Walk (TRW) algorithm, proposed in [1], propose to use the observed disparity between the proportion of successfully established connections between legitimate remote hosts and malicious ones.TRW uses sequential hypothesis testing to distinguish between benign remote hosts who occasionally send misaddressed traffic and remote scanners who are often likely to probe non active IP addresses or ports. However, despite its advantages, one can still easily evade TRW. In [10], the authors proposed a distributed scan method, named

z-scan that is using a limited number of zombies. This method is extremely effective against TRW.

Non-counting approaches have been proposed [7, 8, 11], to address problems with counting methods. The authors of [8] proposed a probabilistic approach to scan detection. It computes for each address an access probability distribution, calculated across all remote source IP addresses that are contacted by this host. Then, it compares the probability with that of scanners that are assumed as accessing each destination address with an equal probability. The authors of [11] proposed an entropy-based approach to fast scanning worms detection. The proposed approach works on flow-level and computes entropy contents of traffic parameters such as IP addresses. Significant changes in entropy indicate worm propagation. In [7], the authors formalize the problem of scan detection as a classification problem. A data mining classifier algorithm labels each pair (source IP, destination port) as scanner or non-scanner. Despite its performance, this data mining method heavily depends on an accurately labeled training trace and it is not suitable for real-time scan detection; it was used with time windows of 20 minutes.

3 Scan Surveillance

We have seen in the previous section that scan detection poses two tricky challenges. First, probing methods are increasingly varied and refined both for greater efficiency and to operate at lower profile to evade intrusion detection systems. Second, complete session re-assembly of traffic data, needed by many proposed scan detection methods, is impractical for on-line deployment at high-speed vantage points. Maintaining real-time assembly of many connections for many hosts requires much computing resources making the cost of such methods unaffordable for many ISPs. Moreover, as stated in [3], scan activity can be seen as an unavoidable background radiation that hits permanently our network. This means that the presence of scanners is a nuisance that is not *per se* problematic, but a change in the pattern of activity of scanners is an important sign that something is happening, *e*.g.,a new worm propagating or a cibled attack to a network. In place of trying to detect each individual scanner, we should rather monitor the scanning activity to detect change in it. When we detect a change, we can there after apply sophisticated and complex forensic techniques to detect the scanner(s) that have led to a change in scanning activity. In this paper, we will concentrate on the detection of changes in scan activity.

Our approach of monitoring scan activity has analogous foundation to what proposed in [8, 11], in the sense that we that we characterize the scanning activity by its distribution in the space spanned by IP source address (@IPsrc), IP destination address (@IPdst), source port number (# src) and destination port number (# dst) (@IPsrc,@IPdst,# src,# dst). The authors of [8] use the distribution of @IPdst or # dst as a discriminant feature between a scanner and a benign node. The authors of [11] derive the entropy from the distribution of @IPsrc is used as the feature to monitor to detect worm propagation. In this paper we are going further, and we see the joint distribution over the

quadruple (@IPsrc,@IPdst,# src,# dst) as the main feature to use to detect changes in scanning activity. Our scanning monitoring algorithm therefore consists of deriving an aggregated reference distribution on an observation window of w packets. This reference is thereafter compared with the distribution inferred over a running window of the same size. We detect a change in scanning pattern if the deviation from the reference window is larger than a detection threshold. In summary we are following the below three steps;

1. Aggregating suspect scan traffic collected on a chosen vantage point in separate windows of w packets.
2. Computing, for each window, an empirical joint distributions of the features of interest in the scanning traffic.
3. Analyzing these empirical distributions and tracking deviation from a reference distribution to expose the presence of serious security problems that need more investigation.

We saw in previous section that detecting individual scans is a challenging task. Fortunately our objective is to detect change in scanning pattern at any observation point, *i*.e.,we should be able to monitor scanning traffic using only locally available information. So we will not use any scan detection algorithm to collect scan traffic. We propose to collect on every monitored point all SYN packets that are not followed by SYN/ACK responses within a 60 seconds interval. Collecting such traffic, although it may include benign traffic as well as the real scan traffic, has three basic benefits: first, its collection is more ressource-friendly as we are ignoring all non-SYN packets and we do not need complete session re-assembly; second, such traffic is mainly composed by scans as usually, we can neglect mis-configuration traffic and transitory network congestion or failures. Finally, isolated SYN packets form most of scan traffic. Non-SYN scans suffer from TCP/IP implementation idiosyncrasies, which limits their efficiency and therefore their use.

3.1 Feature Selection

Our aim here is to track changes in the distribution of scan activity in the space spanned by (@IPsrc,@IPdst,# src,# dst). This choice comes from the observation that different port scanning strategies affects feature distributions in diverse manners. A horizontal scan affects the # dst distribution of the scan traffic, so that it becomes skewed and concentrated on the vulnerable port being scanned. Moreover, if the distribution of the source IP address field is also skewed than means that the a single or a few numbers of IP addresses conduct the attack (it is probably a botnet master, looking for vulnerable machines to send them a bot), otherwise it is most likely a worm scan. In the same way, vertical scans might be detectable as a change in the distributional aspect of source and destination IP addresses, which becomes more concentrated: on the attacker IP address for the @IPsrc distribution and on the victim for @IPdst. However, coordinated scans are less obvious to detect since the attacker can probe many ports on different target IP addresses using many source IP addresses; therefore making all feature

distributions appears dispersed. For this case we expect to have to use the full set (@IPsrc,@IPdst,# src,# dst) to be able to detect the changes. Thus, we will use the following traffic features to detect scanning attacks and worms:

- (@IPsrc,# dst): to detect emerging horizontal scans from botnet masters and individual attackers.
- (# dst): to detect new worm scans.
- (@IPsrc, @IPdst): to detect new or repetitive vertical scans.
- (@IPsrc,@IPdst,# src,# dst): to detect coordinated scans.

3.2 Distribution Inference

In most general settings, computing the traffic features joint distributions for each time window consists of having a vector of up to 2^{96} entries [1], where each entry represent a possible assignment of the features values. Every time the feature value assigned to an entry is observed it is incremented. Hopefully, the number of distinct TCP port pairs is near 25000 where number of distinct IP address pairs in a 24 hours scan trace (see Table 1) is about 2 millions. Although these numbers are lower than the 2^{64} and 2^{32} possible ones, they remain important, and it is still unfeasible to estimate precisely the distribution. We have therefore, to resort to estimate an aggregated histogram. One solution to build it is to apply a mask that aggregates into a single bin all features values sharing the same value of mask. Even if this solution is easy to implement it suffers from a lack of flexibility, aggregation level is not easy to control and most important it is too deterministic, *i*.e.,an attacker can guess the mask applied and make use of it to hide its scan traffic. To address these two problems we propose to apply a random hash function to each feature and to aggregate value based on the resulting hash values. This approach is easy to implement, leads to flexible aggregation, and last, it is immune to the attacker guess. To have more security one can even change frequently the random hash.

One might use the aggregated histogram to estimate precisely a parametric distribution form [12]. However in this work, we want to stay non-parametric so we will directly use the histogram without refining it into an estimated distribution.

3.3 Change Detection

We have postulated earlier in section 3 that we can infer change in scanning pattern by observing changes in the joint distribution of features. We need to define a way to measure the discrepancy between two distributions. This problem is indeed central in statistics. Two main approach classes can be defined to deal with this problem. A first class of approach is parametric. They consist of measuring the distance of two distributions through the change in parameters of a parametric distribution. Parametric inference and statistical tests associated with it have widely used these approaches.

[1] (@IPsrc,@IPdst,# src,# dst) is represented with 96 bits.

A second class is non-parametric and does not use any particular type of distribution. Entropy based approach [9, 11] belongs to this category. Entropy is the measure of missing information when we do not know the value of a random variable realization. The intuition beyond using entropy in anomaly detection is that we expect that a change in the distribution lead to proportional change in entropy. Unfortunately, a closer analysis of entropy shows that it is not a good indicator of distribution variation. To see this let us assume a two valued random variable with a distribution $(p, 1-p)$ and let us assume that this distribution varies to $(p+\Delta, 1-p-\Delta)$. A sensitivity analysis of the entropy $H(p)$ to variation of p can be done through the derivative:

$$\frac{\partial H}{\partial p} = \log\left(\frac{p}{1-p}\right)$$

One can therefore expect that the effect of a small variation Δ on the distribution on the entropy would be approximatively $\Delta \log\left(\frac{p}{1-p}\right)$, *i*.e.,the sensitivity of the entropy variation depends strongly on the value of p and not on the variation of the distribution; for small p the entropy will vary largely with small variation of p and for larger values the entropy will not vary as much. The sensitivity analysis shows that even very small variations of a bin probability (that could eventually result from small random fluctuations) can lead to large variation of the entropy. This elucidates why from a theoretical standpoint entropy is not a good metric to use for tracking change in distributions as small variation in distribution might lead to large variations in entropy and large variation in the distribution might not lead to important entropy variation.

In place of entropy, we propose to use the relative entropy, also named Kullback-Leibler divergence (KLD) [13] , to track changes in distributions obtained over hashed distributions. The Kullback-Leibler divergence is an information theoretic measure of the difference between two probability distributions defined on the same finite set $\mathcal{H}$: a distribution P and a reference probability distribution Q. It is given by equation (1).

$$D(P\|Q) = \sum_{x\in\mathcal{H}} P(x)\log\left(\frac{P(x)}{Q(x)}\right) \tag{1}$$

The KLD is not defined when $\exists x \in \mathcal{H}, Q(x) = 0, P(x) \neq 0$. By applying a sensitivity analysis similar to what done for entropy to a two valued discrete distribution we obtain:

$$\frac{\partial D}{\partial p} = \log\left(\frac{p}{q}\right) + \log\left(\frac{1-q}{1-p}\right)$$

This shows that in contrast with entropy the sensitivity of KLD does not depend on the value of p alone but on the ratio between p and the reference distribution Q, *i*.e.,KLD is highly sensitive if we have a large variation compared to the reference distribution, and become less sensitive if this variation is smaller. This is perfectly what we expect from a distribution variation metric. This validates

the use of KLD in place of entropy, and it explains also some of the poor results obtained when using entropy to monitor distributional changes.

Furthermore, we can formulate anomaly detection using feature distribution of the random variable X as follows. Let us assume that the scan traffic follows a stable distribution P and we are observing n observations of scan packets. Let us Q_n be the empirical feature distribution. Anomaly detection reduces to decide the following composite hypothesis test:

- H_1 : The observed scan traffic features follow the stable distribution P, *i*.e.,there is no change.
- H_2 : The observed scan traffic features follow an alternative distribution P' *i*.e.,there is a change.

It can be shown that the classical likelihood ratio test can be replaced by the below test on KLD:

$$|D(Q_n\|P) - D(Q_n\|P')| > \frac{1}{n}\log T$$

where T is the decision threshold. The Stein lemma [13] states that if one fixes the probability of false alarm (the probability that no change happened but we still decided that a change occurs) to a value and want to minimize the probability of misdetection (the probability that there are changes and we cannot detect them), this minimal misdetection probability will depend on $D(P\|P')$. However in practice we do not know the type of changes that will happen *a priori*, so the alternative distribution P' is unknown. We have therefore, to replace the change detection test by the below one:

$$|D(Q_n\|P)| > \frac{1}{n}\log T$$

However, this test is not optimal anymore and we cannot derive the misdetection errors analytically, and we have to resort to Receiver Operation Characteristic (ROC) curves. These curves show the tradeoff achieved between false alarm and mis-detection rate for a given change detection scheme.

4 Experimental Validation

The trace used in this paper consists of all packet headers that have passed during April 4th to 5th , 2006 through the 100 Mb/s link connecting Tunisian National University Network (TNUN) to the Internet. Table 1 summarize that trace and reports the total number of packets and flows. It also shows the share of transport protocols above IP protocol.

We have assumed that all SYN packets that are not followed by SYN/ACK responses within a 60 seconds interval are scan traffic. This leads to 10 millions suspicious packets that have 18388 sources and are destinated to 56585 destinations, leading to 2 millions address pairs. The traffic targeted up to 25 000 different port numbers. The suspicious packets account for 3% of total number of packets and 42% of the total number of connections.

Table 1. Summary of the packet trace used for validation

Period	Packets number in millions	Flow number in millions	Protocol share(%)		
			TCP	UDP	ICMP
April 4-5 , 2006	356	10.5	97.5	1.8	0.7

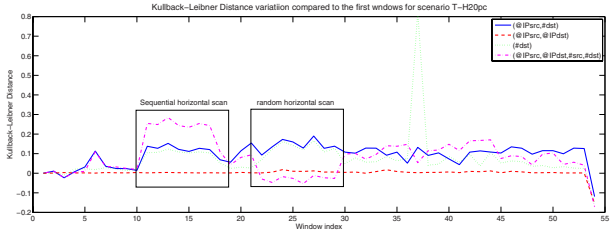

Fig. 1. Evolution of Kullback distance computed for four sets of features with a horizontal scan inserted

4.1 Validation on Artificially Changed Scan Traffic Traces

To validate the method we first create artificial changes in scanning profile and see whether we can detect these changes using the proposed methods. For this purpose, we have used Nmap tool to generate 5 scan traces: 2 horizontal scan targeted to port 80 on randomly chosen IPs (a sequential scan on all IPs of an address mask, and a random scan), and 3 vertical scans (a sequential, a random, and a targeted) to many ports on a single IP. These scan packets are injected in a real trace by mixing according to a certain percentage (referred as scan intensity) them with the originally observed real trace (T). We provide in table 2 the details of the resulting traces. We used for each set of features a hash function on 5 bits, *i*.e.,the feature distribution is obtained over 32 bins. We obtain over each fixed window of 1000 packets an empirical histogram. We used the histogram over the first window as the reference histogram for detecting changes in scanning pattern.

In Fig. 1, we present the KLD variation obtained for the scenario T-H20pc with two horizontal scans injected. The figure shows that (@IPsrc, # dst), (# dst) and (@IPsrc,@IPdst, # src, # dst), shows an noticiable change of the KLD. However unexpectedly that the distribution on (@IPsrc, @IPdst) do not show a major change. This could be explained because of the use of random hash function that distributes the value in the hash space, so that the horizontal scan

Table 2. Artificial scan traces summary

Trace	Description	Intensity
T	Original scan trace	0%
T-V20pc	Three vertical scans injected in window 100, 260, and 430	20%
T-H20pc	Two horizontal scans injected in window 100 and 260	20%

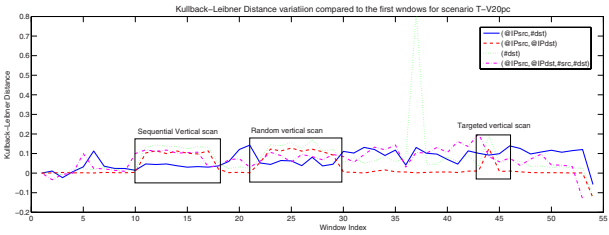

Fig. 2. Evolution of Kullback distance computed for four features compared to the first window for a one hour scan trace

is not affecting too much the distribution. Interestingly the KLD for (@IPsrc, # dst) and (# dst) show almost the same values during the horizontal scan injection. This could be explained by the fact that the variation in feature space is essentially related to port variations that are happening during horizontal scans. Moreover KLD variations are more emphasized on the (@IPsrc,@IPdst, # src, # dst) than on other features. This validates the use the approach proposed in the paper for detecting changes in horizontal scanning pattern. In Fig. 2, we present the KLD variation obtained for the scenario T-V20pc with three vertical scan injected. The figure shows that (@IPsrc, # dst) is not well reacting to the introduction of vertical scans, when (@IPsrc, @IPdst), as well as (@IPsrc,@IPdst, # src, # dst), are well reacting to the injected scan level changes. This validates the use the above three set of features for detecting vertical scans. In vertical scan this is mainly destination address and destination port that are changed, one can expect to not see it on the feature (@IPsrc, # dst). It is noteworthy that the feature (@IPsrc,@IPdst, # src, # dst) seems to be reactive to the two types of scans: horizontal and vertical. However, using each individual feature enable also to know if the change is resulting from vertical or horizontal scanning changes, where using (@IPsrc,@IPdst, # src, # dst) does not give this information.

4.2 Mining Scan Traffic Changes in Real Data

We present in the Fig. 3, the KLD variations for selected features over the whole scan traffic trace. The figure first validates the assumption that there is a stable value for KLD. This can be seen by observing that most of time the KLD distance is close to zero and has a flat structure. However some clear peaks are visible. The KLD computed over (@IPsrc, # dst) pair presents many peaks that are relative to horizontal scans. The figure also shows clearly some vertical scan that are visible by sharp peaks on the (@IPsrc, # dst) graph. However, the intensity variation is larger for horizontal scan than vertical one. This is in accordance with other studies [2–4] findings that affirmed the predominance of horizontal scans compared to other scanning strategie. Interestingly one can observe a increase and a deacrease in the KLD obtained over (@IPsrc,# dst) feature in windows 700 to 800. This increase represents a significant change in horizontal scanning behaviour. It has happened from 10 PM to 6 AM the next

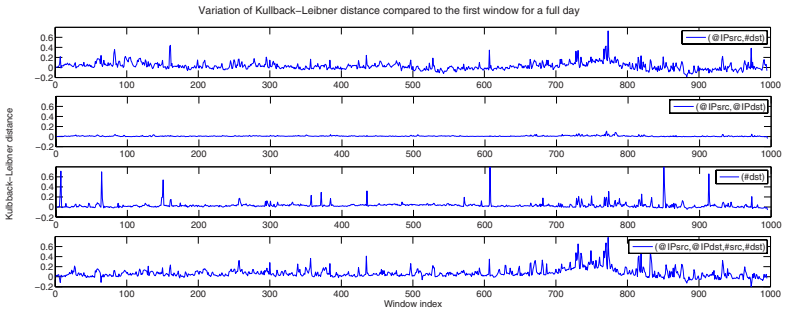

Fig. 3. Evolution of Kullback distance computed for four features to the first window for a one hour scan trace

day. This could be related to a significant scan activity targeting TNUN. Here also (@IPsrc,@IPdst, # src, # dst) seems to be a good tradeoff between the all the feature. It regroups peaks in (@IPsrc, @IPdst) as well as (@IPsrc, # dst) curves and it can also detect the change happening between time 700 and 800.

5 Conclusion

We presented in this paper a scan surveillance approach based on Kulback-distance-based approach. The proposed approach is completely non-parametric and does not need any hypothesis on the nature of scan traffic. We provided preliminary evidence that the method is effective. However, a complete analysis will need a complete study with a larger set of traces and particularly the observation of a real change in scanning pattern (for example in the advent of a new worm propagating). We are actually instrumenting equipment in network to do this real time analysis.

References

1. Jung, J., Paxson, V., Berger, A., Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing. In: IEEE Symposium on Security and Privacy, pp. 211–225 (2004)
2. Allman, M., Paxson, V., Terrell, J.: A Brief History of Scanning. In: ACM SIGCOMM/USENIX Internet Measurement Conference (October 2007)
3. Pang, R., Yegneswaran, V., Barford, P., Paxson, V., Peterson, L.: Characteristics of internet background radiation. In: 4th ACM SIGCOMM IMC conference, pp. 27–40. ACM, New York (2004)
4. Yegneswaran, V., Barford, P., Ullrich, J.: Internet intrusions: global characteristics and prevalence. In: ACM SIGMETRICS, pp. 138–147. ACM, New York (2003)
5. Paxson, V.: Bro: a system for detecting network intruders in real-time. In: 7th conference on USENIX Security Symposium, pp. 2435–2463. USENIX Association, Berkeley (1998)

6. Roesch, M.: Snort–Lightweight Intrusion Detection for Networks. In: 13th Conference on Systems Administration LISA 1999, pp. 229–238 (1999)
7. Simon, G., Xiong, H., Eilertson, E., Kumar, V.: Scan Detection: A Data Mining Approach. In: Proceedings of the Sixth SIAM International Conference on Data Mining, pp. 118–129 (2006)
8. Leckie, C., Kotagiri, R.: A probabilistic approach to detecting network scans. In: Network Operations and Management Symposium, NOMS 2002. 2002 IEEE/IFIP, pp. 359–372 (2002)
9. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: SIGCOMM conference, pp. 217–228. ACM, New York (2005)
10. Kang, M., Caballero, J., Song, D.: Distributed Evasive Scan Techniques and Countermeasures. In: Hämmerli, B.M., Sommer, R. (eds.) DIMVA 2007. LNCS, vol. 4579, pp. 157–174. Springer, Heidelberg (2007)
11. Wagner, A., Plattner, B.: Entropy based worm and anomaly detection in fast IP networks. In: 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, pp. 172–177 (2005)
12. Akodjenou-Jeannin, M., Salamatian, K., Gallinari, P.: Flexible Grid-Based Clustering. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS, vol. 4702, pp. 350–357. Springer, Heidelberg (2007)
13. Cover, T.M., Thomas, J.A.: Elements of information theory. John Wiley and Sons, Inc., Chichester (1991)