COMPUTERS & SECURITY 43 (2014) 35–48

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

On fingerprinting probing activities

Elias Bou-Harb [a,b,*], Mourad Debbabi [a,b], Chadi Assi [a,b]

[a] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada
[b] National Cyber-Forensics and Training Alliance (NCFTA), Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 2 July 2013
Received in revised form
26 November 2013
Accepted 15 February 2014

Keywords:
Fingerprinting probing activities
Probing analysis
Statistical approach
Unsupervised data clustering
Network scanning

ABSTRACT

Motivated by recent cyber attacks that were facilitated through probing, limited cyber security intelligence and the lack of accuracy that is provided by scanning detection systems, this paper presents a new approach to fingerprint probing activity. It investigates whether the perceived traffic refers to probing activities and which exact scanning technique is being employed to perform the probing. Further, this work strives to examine probing traffic dimensions to infer the 'machinery' of the scan; whether the probing is random or follows a certain predefined pattern; which probing strategy is being employed; and whether the probing activity is generated from a software tool or from a worm/bot. The approach leverages a number of statistical techniques, probabilistic distribution methods and observations in an attempt to understand and analyze probing activities. To prevent evasion, the approach formulates this matter as a change point detection problem that yielded motivating results. Evaluations performed using 55 GB of real darknet traffic shows that the extracted inferences exhibit promising accuracy and can generate significant insights that could be used for mitigation purposes.

1. Introduction

Recent events demonstrated that cyberspace could be subjected to amplified, debilitating and disrupting cyber attacks leading to drastic impacts on provided network and Internet wide services. For instance, it was disclosed that hackers had orchestrated multiple breaches of Sony's PlayStation Network taking it offline for 24 days and costing the company an estimated $171 million (Techcrunch). Citibank revealed that it had detected a data breach that publicly exposed 360,000 North American credit card holders' account details, including their account numbers, names, and email addresses (Forbes). Moreover, hackers have targeted Twitter gaining substantial access to around 250,000 user accounts (Hackers Targeted). The latter cyber attack came days after US newspapers, the New York Times and the Wall Street Journal, revealed that their respective websites had been the target of a well coordinated hacking effort (Major US Newspapers). Further, Google became the target of a phishing campaign that aimed at gaining access to the accounts of senior officials in the U.S., Korea and other governments (Computer World). Moreover, a hacker team dubbed as LulzSec took down CIA.gov with a targeted denial of service attack (The Next Web). Last but not least, Facebook officials announced that they have discovered that computers belonging to several of its engineers had been hacked using a zero-day Java attack (ArsRechnica). In general, cyberspace could facilitate advanced persistent threats (Daly, 2009), distributed denial of service attacks (Chung, January 2012), zero day exploits (Bilge and Dumitras, 2012) and cyber

* *Corresponding author*. Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada H3G1M8. Tel.: +1 5146495049.
E-mail address: e_bouh@encs.concordia.ca (E. Bou-Harb).
http://dx.doi.org/10.1016/j.cose.2014.02.005
0167-4048/

terrorism/warfare (Symantec). Despite efforts to protect the cyberspace, the latest reports from senior government officials (National Security Council) highlighted that only limited progress has been made in improving the cyber security of crucial networks.

Probing, the task of scanning enterprise networks or Internet wide services, searching for vulnerabilities or ways to infiltrate IT assets, is a significant cyber security concern (Bou-Harb et al., 2013a). The latter is due to the fact that probing is commonly the primary stage of an intrusion attempt that enables an attacker to remotely locate, target, and subsequently exploit vulnerable systems. It is basically a core technique and a facilitating factor of the above mentioned cyber attacks. Additionally, it was concluded that a momentous 50% of attacks against cyber systems are preceded by some form of network scanning activity (Panjwani et al., 2005). Furthermore, there has been recently a flourishing of a new cyber phenomenon dubbed as cyber scanning campaigns (Dainotti et al., Nov 2012; Internet Census, 2012; Reports of a Distributed). These are scanning approaches that are extremely distributed, possess composite stealth capabilities and high coordination and orchestration. Rather than focusing on specific hosts or networks, these campaigns aim at probing and consequently exploiting Internet's wide services and infrastructures.

Motivated by the above, in addition to the lack of accuracy that is provided by current scanning detection systems (Zambon & Bolzoni), this paper attempts to answer the following questions:

- Does the perceived traffic refer to probing activities?
- Which specific scanning technique is being employed to perform the probing?
- Is the probing random or does it follow a certain predefined pattern?
- What is the adopted probing strategy?
- Who is generating such activity?

Stimulated by the requirement to answer the above and hence generate insights that could be used for mitigation purposes, we frame the paper's contributions as follows:

- Proposing a new method to fingerprint probing activity (i.e., detect activity and identify the technique). The method does not rely on identifying the scanning source and is independent from the scanning strategy (remote to local, local to remote, local to local), the scanning aim (wide range or target specific) and the scanning method (single source or distributed). Further, the proposed method does not rely on a certain predefined alert threshold, the transport protocol used (TCP or UDP) or the number of probed destinations and ports. The method uniquely leverages the Detrended Fluctuation Analysis (DFA) technique.
- Validating our method by comparing its clustering capabilities with the well established machine learning clustering approaches, namely, the k-means and Expectation Maximization (EM) methods.
- Employing various statistical test techniques, including Bhattacharyya distance, Mann–Kendall and WaldWolfowitz, to analyze probing traffic dimensions, namely, employed technique, monotonicity and randomness.
- Adopting a change point detection mechanism based on a time series Bayesian inference to prevent approach evasion that could be attempted by a malicious attacker or due to network traffic fluctuations.
- Evaluating the proposed approach using 55 GB of real darknet traffic.

The remainder of this paper is organized as follows. Section 2 elaborates on our proposed approach by attempting to answer the above mentioned questions. Specifically, it describes the rationale and methodology, presents the observation validation techniques, and pinpoints the possible obtained cyber security inferences. Section 3 presents the empirical evaluation and discusses the results. The evasion prevention approach is highlighted in Section 4. A discussion of the limitations of the proposed approach is stated in Section 5. Related work is reviewed in Section 6. Finally, Section 7 summarizes the paper and pinpoints the future work.

2. Proposed approach

2.1. *Does the perceived traffic refer to probing activities?*

Organizations are interested in generating insights and inferences concerning any probing activities that they might receive. It is significant for them to have the capability to fingerprint probing events. Thus, they would benefit from knowing if the perceived traffic is related to scanning or not and if it is, exactly which scanning technique has been employed. This section aims at tackling these two issues.

The rationale of the proposed method states that regardless of the source, strategy and aim of the probing, the reconnaissance activity should have been generated using a certain literature-known scanning technique (i.e., TCP SYN, UDP, ACK, etc. (Bhuyan et al., 2010)). We observe that a number of those probing techniques demonstrate a similar temporal correlation and similarity when generating their corresponding probing traffic. In other words, the observation states that we can cluster the scanning techniques based on their traffic correlation statuses. Subsequently, we can differentiate between probing and other malicious traffic (i.e., Denial of Service (DoS)) based on the possessed traffic correlation status. We can as well attribute the probing traffic to a certain cluster of scanning techniques (i.e., the probing activity, after confirmed as probing, can be identified as being generated by a certain cluster of techniques that possess similar traffic correlation status). To identify exactly which scanning technique has been employed in the probing, we statistically estimate the relative closeness of the probing traffic in comparison with the techniques found in that cluster.

To enable the capturing of traffic signals correlation statuses, the proposed method employs the Detrended Fluctuation Analysis (DFA) technique. DFA was first proposed in (Peng et al., 1994) and has since been used in many research areas to study signals correlation. Very limited work in the areas of cyber security and malicious traffic detection has utilized DFA

(Harder et al., 2006; Fukuda et al., 2008), and to the best of our knowledge, no work has leveraged the DFA technique to tackle the problem of fingerprinting probing traffic.

The DFA method of characterizing a non-stationary time series is based on the root mean square analysis of a random walk. DFA is advantageous in comparison with other methods such as spectral analysis (Priestley, 1981) and Hurst analysis (Matos et al., 2008) since it permits the detection of long range correlations embedded in a seemingly non-stationary time series. It avoids as well the spurious detection of apparent long-–range correlations that are an artifact of non-stationarity. Another advantage of DFA is that it produces results that are independent of the effect of the trend (Hu et al., 2001).

Given a traffic time series, the following steps need to be applied to implement DFA:

- Integrate the time series; The time series of length N is integrated by applying

$$y(k) = \sum_{i=1}^{k} [B(i) - B_{ave}] \quad (1)$$

- where $B(i)$ is the ith interval and B_{ave} is the average interval.
- Divide the time series into "boxes" (i.e., bin size) of length n.
- In each box, perform a least-squares polynomial fit of order p. The y coordinate of the straight line segments is denoted by $y_n(k)$.
- In each box, detrend the integrated time series, $y(k)$, by subtracting the local trend, $y_n(k)$. The root-mean-square fluctuation of this integrated and detrended time series is calculated by

$$F(n) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} [y(k) - y_n(k)]^2} \quad (2)$$

- Repeat this procedure for different box sizes (i.e., time scales) n

The output of the DFA procedure is a relationship $F(n)$, the average fluctuation as a function of box size, and the box size n. Typically, $F(n)$ will increase with box size n. A linear relationship on a log–log graph indicates the presence of scaling; statistical self-affinity expressed as $F(n) \sim n^{\alpha}$. Under such conditions, the fluctuations can be characterized by a scaling exponent α, which is the slope of the line relating log $F(n)$ to log (n). The scaling exponent α can take the following values, disclosing the *correlation status* of the traffic time series.

- $\alpha < 0.5$: anti-correlated.
- $\alpha \approx 0.5$: uncorrelated or white noise.
- $\alpha > 0.5$: correlated.
- $\alpha \approx 1$: 1/*f*-noise or pink noise.
- $\alpha > 1$: non-stationary, random walk like, unbounded
- $\alpha \approx 1.5$: Brownian noise.

We proceed by fingerprinting the scanning techniques. For the scope of the current work, we have selected 10 cyber scanning techniques. To accomplish the task, we created an experimental environment that includes two virtual machines, a scanning and a target machine. Note that the virtual environment was hosted by VMware software while the machines were running Ubuntu Linux 10 with 2 GB of RAM and 2.1 GHz dual core CPUs. The machines are isolated from any external networks to prevent any noise in the generated signal. The target machine does not operate any special service. We have also setup a TCPDUMP (Jacobson et al., 1989) sink on the target to collect the network traffic data originating from the scanning machine. It is worthy to note that we do not record the traffic response from the target as our intention is to capture the scanning techniques' traffic regardless of the offered services by the probed machine. To execute the scanning activity, we have utilized Nmap (Lyon, 2009), an open source utility for network scanning and discovery. We ran Nmap 10 times, once for each scanning technique, and collected the generated traffic in 10 packet capture (pcap) (Jacobson et al., 1989) traffic files. The selected scanning techniques and the corresponding required Nmap command flags to execute each of the techniques are summarized in Table 1. For detailed information about the concerned scanning techniques, we refer the reader to (Bhuyan et al., 2010; Lyon, 2009). The generated packets' distribution of the 10 scanning techniques is illustrated in Fig. 1. Subsequently, we applied the DFA technique on each of the scanning techniques' traffic signals. To achieve that, we have utilized the DFA MATLAB code found in (Little et al.) and used **1 ms** as the bin size **for all** the 10 scanning techniques. The outcome of applying the DFA on the previous scanning traffic time series distributions is shown in Fig. 2 and the output of the scaling exponents α is summarized in Table 2. From Table 2 and the information relating the scaling exponent α to the correlation status, we can produce Table 3 that discloses that a number of scanning techniques in fact demonstrated a similar temporal correlation and similarity when generating their corresponding probing traffic.

It is significant to pinpoint that such results are independent from the used scanning tool. In this work, we have used Nmap since it is the most widely adopted and well established scanning tool. Moreover, it provided a simple

Table 1 – Selected cyber scanning techniques and Nmap command flags.

Cyber scanning technique	Nmap command flags
TCP SYN Scan	-sS
TCP connect() Scan	-sT
FIN Scan	-sF
Xmas Scan	-sX
Null Scan	-sN
UDP Scan	-sU
IP Protocol Scan	-sO
ACK Scan	-sA
Window Scan	-sW
RPC Scan	-sR

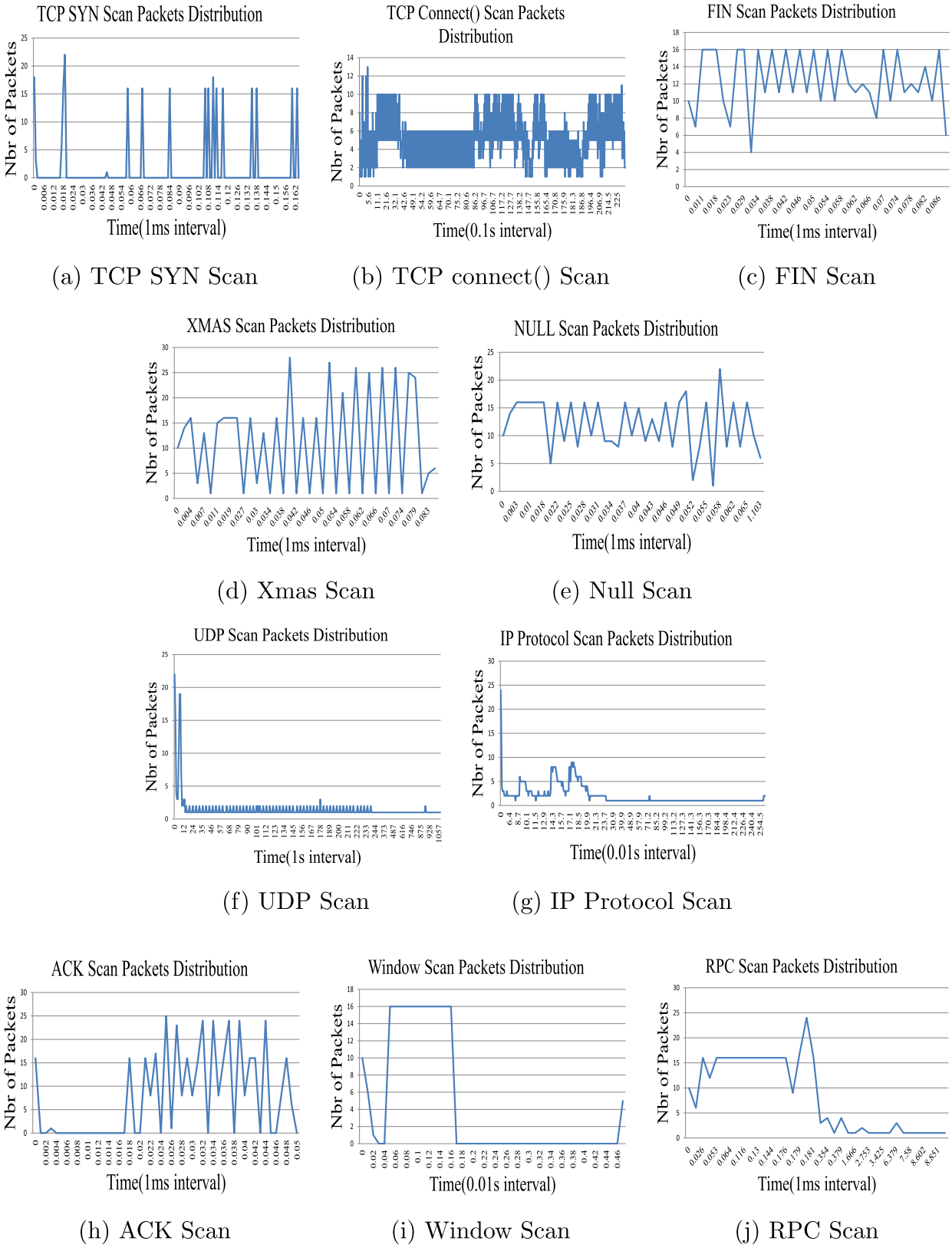

Fig. 1 – Packets' distribution generated by the scanning techniques.

mechanism to generate the scanning traffic. We argue that same scanning techniques will generate a somehow similar traffic distribution regardless of the tool used and hence will output similar DFA results. To support this statement, we executed an experiment using Fing, another network scanning and discovery tool. We also selected the TCP SYN Scan since it is the most popular scanning technique (Staniford et al., 2002a). We repeated the same experimentation as

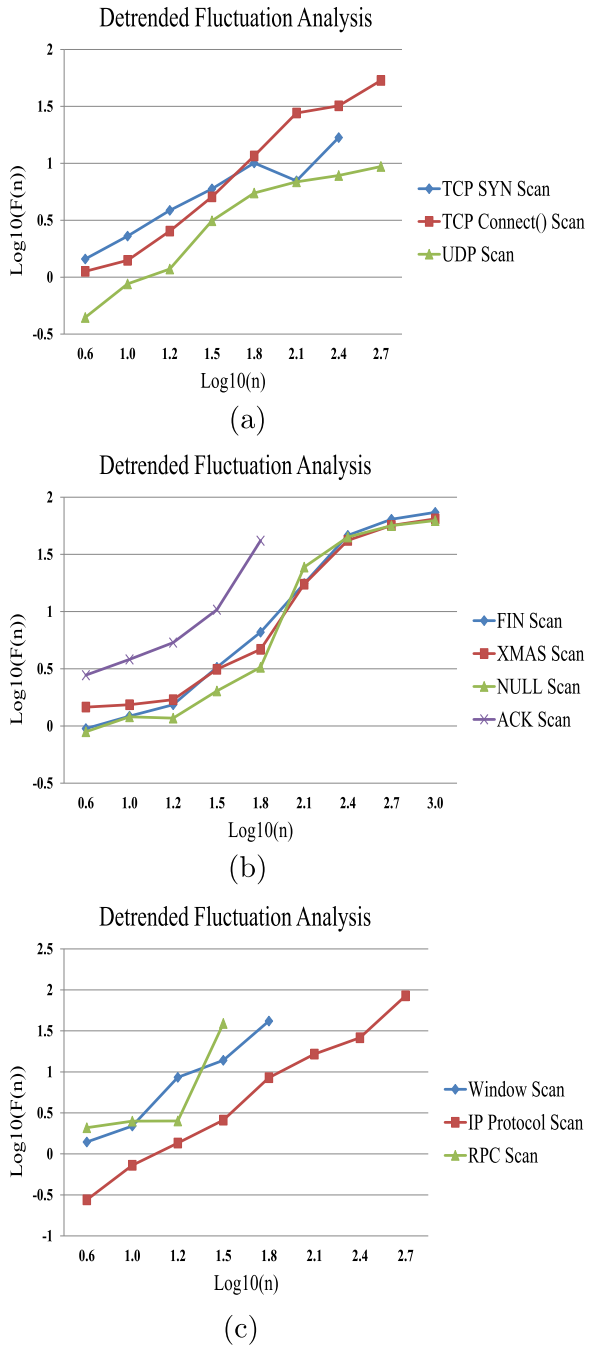

Fig. 2 – Applying DFA on the scanning techniques traffic signals.

we did with the other scanning techniques. The output of the DFA scaling exponent α was =0.55. Therefore, the correlation status was shown to be 'correlated' which is

[1] We further validated our observation by executing the same experiment using three scanning tools, namely, Unicornscan, AATools Port Scanner and Superscan on other scanning techniques. Moreover, the presented empirical evaluation in Section 3 will indeed demonstrate the effectiveness of the proposed approach.

Table 2 – Summary of the DFA scaling exponent α.

Cyber scanning technique	Scaling exponent
TCP SYN Scan	0.57
TCP connect() Scan	0.87
FIN Scan	0.31
Xmas Scan	0.30
Null Scan	0.37
UDP Scan	0.66
IP Protocol Scan	1.13
ACK Scan	0.44
Window Scan	1.24
RPC Scan	1.31

coherent with the DFA results that we have previously obtained with the TCP SYN Scan when we used Nmap.[1] We can generalize such result to other techniques as well; since DFA operates on packets distribution in a time series, where similar scanning techniques, when following the protocol and technique standards, will possess similar distributions when probing their target, then we can expect similar DFA results regardless of the tool used.

It is also noteworthy to mention that attackers/scanners will not be able to avoid detection by modifying the correlation status of a probing signal because of two reasons. First, since they are going to employ one of the techniques of Table 2, which presents a comprehensive list of scanning techniques, regardless of whether they use a tool or anything else, they will indeed generate a probing signal that will be fingerprinted by the proposed approach. Second, the fact the we aim to leverage a darknet space to perform the inference in which the scanners have no knowledge about its existence, will cause the scanners to be detected. We argue that scanners will not go into the trouble of developing entirely new probing techniques that do not rely on the techniques of Table 2 and at the same time possess the capability to detect darknets, where both tasks are known to be hard (Woodhead, 2012), if not impossible, and impractical from a scanner perspective.

We proceed by presenting Fig. 3, which depicts the system process that is adopted by the proposed approach to permit the inference of the probing activities as well as the employed probing techniques. One of the issues that arises is where to apply DFA given a traffic time series that needs to be tested; this problem could be re-stated as follows: given a traffic time series S_t, find a starting position X and an ending position $X + \sigma$ in S_t where we can apply DFA on. Assuming a 'random' or a 'predefined' window location in the time series S_t to apply DFA on will be erroneous as this will result in wrong inferences. For example, if the traffic time series that needs to be tested is of length 5 min, applying DFA on the entire distribution could indicate a result (suppose it was inferred that it is not scanning) while the actual scanning starts on the 3rd minute; the entire distribution's correlation status appears to be close to noise (i.e., α value $\approx$1 and hence not scanning) while from the 3rd minute up to the 5th the signal is correlated (i.e., scanning). To tackle this, we present Algorithm 1 which discloses an approach to approximate when to apply DFA, to correctly infer whether or not the traffic refers to probing activities.

input : A time series S_t of the distribution under testing; the set of time series Sc_p of the distributions of the scanning techniques.
output: X, reflecting the starting location on where to apply DFA in S_t

m=length(S_t);
for every Sc_p **do**
n=length(Sc_p);
for i=*1* $\rightarrow (m-n)$ **do**
  s[i]=compare[$S_t(1+i, \cdots, n+i)$, $Sc_p(1, \cdots, n)$];
end
S[p]=min(s[]);
end
X=min(S[]);
return (X);

compare(A, B)
for i=*1* $\rightarrow n$ **do**
  $K[i]= \|\mathbf{E}\| = \mathrm{d}(A(i), B(i))$;
  sum+=$K[i]$;
  return (sum);
end

Algorithm 1. Approximating the starting location on where to apply DFA in S_t.

As shown in Fig. 3, Algorithm 1 takes as input the time series S_t of the distribution under testing and all the other distributions of the scanning techniques Sc_p of Fig. 1. For every distribution related to the scanning techniques, it calculates the euclidean distance **E** between the points in Sc_p and S_t. Subsequently, the scanning technique distribution is moved one point in a sliding window fashion against S_t. For each sliding window, it records the distance between Sc_p and S_t. After finishing all the sliding window procedures, the algorithm stores the minimum distance between both sliding windows in all the iterations. The algorithm finally selects X as the minimum of all the distances in all sliding windows after all the techniques Sc_p have passed on S_t. This will approximate the starting position on where to apply DFA in S_t. Note that, σ in the ending position $X+\sigma$, that was previously mentioned, is the length of the scanning technique Sc_p where X was derived from. It is significant to note that we use the scanning techniques Sc_p of Fig. 1 as a way to infer, in an apriori fashion, where to start applying DFA and hence where we estimate that the scanning is initiated; we are not completely matching the techniques with the input time series S_t. In fact this is not the intention of Algorithm 1 and hence we do not expect the techniques to completely overlap the input time

Table 3 – Cyber scanning techniques and corresponding correlation statuses.

Correlation status	Cyber scanning techniques
Anti-correlated	FIN Scan
	Xmas Scan
	Null Scan
	ACK Scan
Correlated	TCP SYN Scan
	TCP connect() Scan
	UDP Scan
Non-stationary	IP Protocol Scan
	Window Scan
	RPC Scan

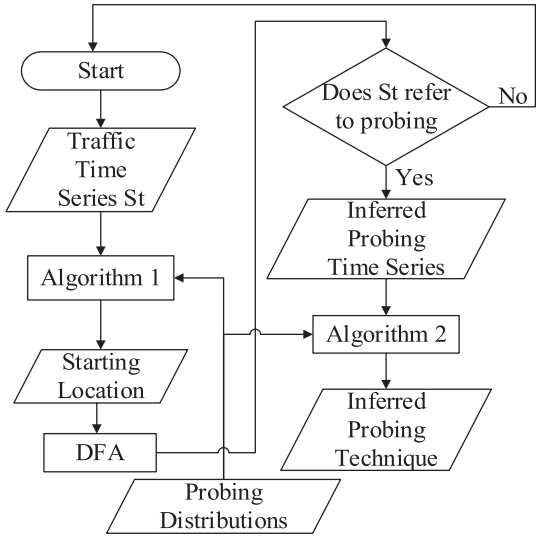

Fig. 3 – Employed system process.

series S_t. Thus, any variation of scanning techniques' distributions is tolerated and manageable, as long their correlation status, as we expect and observe, are kept stable.

After applying DFA, given the output information of Algorithm 1, we end up with a certain correlation status. We expect that the correlation status indicates a probing activity (recall Table 3). However, we may encounter the case where the correlation status does not indicate probing (i.e., uncorrelated, 1/*f*-noise or Brownian noise). If the activity refers to probing, then the output correlation status will lead us to a certain cluster of scanning techniques (of Table 3) that this probing activity has been generated from. To exactly identify which scanning technique has been used to probe, we present Algorithm 2, which discloses an approach to achieve that.

Algorithm 2. Identifying the scanning technique.

As depicted in Fig. 3, Algorithm 2 takes as input a time series S_b of the probing distribution that DFA was previously applied on; S_b refers to the time series extracted from X to $X + \sigma$. For each of the scanning techniques Sc_{bi} in the cluster Sc_b that is related to the previous output correlation status, we statistically measure the relative closeness of S_b to each scanning technique in that cluster using Bhattacharyya distance (Kailath, 1967), **Bha**. The latter statistic test is an established and an effective metric to determine the overlap of two sample distributions (Kailath, 1967). Intuitively, the algorithm selects the technique's distribution that is closer to S_b, Sc_{bi}. Sc_{bi} will be identified as the scanning technique that S_b was employing.

2.2. *Observation validation*

From what has been presented in Table 3, we deduced that the techniques could be clustered into 3 groups based on their probing traffic correlation statuses. Thus, the null hypothesis states that the scanning techniques' traffic originates from 3 independent clusters. However, by default, the DFA approach, especially in the context of probing activity, is not an established method to identify clusters. We now aim to validate the proposed method by comparing it with the well established machine learning clustering approaches, namely, the k-means and the Expectation Maximization (EM) techniques. These techniques and the validation results are consequently discussed.

When the data observations are not pre-labeled into defined numerical or categorical classes, as in our case, two standard widely deployed algorithms for data clustering using unsupervised learning could be employed. These are the k-means (MacQueen, 1967) and the EM (Dempster et al., 1977) algorithms. The k-means algorithm finds k clusters by choosing n data points at random as initial cluster centers. Each data point is then assigned to the cluster with the center that is closest to that point. Each cluster center is then replaced by the mean of all the data points that have been assigned to that cluster. Note that, the k-means algorithm operates by minimizing the sum of squared Euclidean distances between data records in a cluster and the clusters mean vector. This process is iterated until no data point is reassigned to a different cluster. On the other hand, the EM algorithm views the data clustering problem as a framework for data density estimation using a probability density function. An effective representation of the probability density function is the *mixture model*, which asserts that the data is a combination of k individual component densities corresponding to k clusters. The EM problem can be summarized as follows: given a set of data observations, identify a set of k populations in the data and provide a density distribution model of each of the populations. Readers who are interested in the details of the EM are referred to Dempster et al., 1977.

Recall that the aim is to validate the soundness of our proposed method by comparing it with the discussed machine learning clustering approaches. We proceed by going back to our scanning traffic pcap files that we have previously collected. Subsequently, we extracted from them a total of 29 data link, network and transport layer packet features as summarized in Table 4. This feature extraction procedure was achieved using the open source jNetPcap API. We consequently compiled the extracted features into a unified data file of 7138 data instances. To apply the k-means and the EM algorithm on our data file, we have respectively used MATLAB's default clustering functionality and the WEKA data mining tool (Hall et al., 2009). The output of those procedures is depicted in Fig. 4.

Fig. 4 clearly shows the formation of 3 clusters. This result provides evidence that the traffic originates from 3 different

input : A time series S_b of the probing distribution that DFA was previously applied on; a cluster of time series Sc_b of the distributions of the scanning techniques related to the correlation status.
output: Sc_{bi}, reflecting one scanning technique that is estimated to be generating the probing activity found in S_b.

for every Sc_{bi} **do**
$Bha_{bi} = \|\mathbf{Bha}\| = \mathrm{d}(\mathbf{Sc_{bi}}, \mathbf{S_b})$;
end
$d_i = Min(Bha_{bi})$;
return $(Sc_{bi}|i \text{ of } d_i)$;

Table 4 – Features description.

Features		
Data link features	1	Delta time with previous capture packet
	2	Packet Length
	3	Frame Length
	4	Capture Length
	5	The flag 'frame' is marked
Network layer features	6	IP Header length
	7	IP Flags.
	8	IP Flags: reversed bit
	9	IP Flags: do not fragment bit
	10	IP Flags: more fragments bit
	11	IP Fragment offset
	12	IP Time to live
	13	IP Protocol
Transport layer features	14	TCP Segment length
	15	TCP Sequence number
	16	TCP Next sequence number
	17	TCP Acknowledgment number
	18	TCP Header length
	19	TCP Flags
	20	TCP Flags: congestion window reduced
	21	TCP Flags: ECN-Echo
	22	TCP Flags: Urgent
	23	TCP Flags: Acknowledgment
	24	TCP Flags: Push
	25	TCP Flags: Reset
	26	TCP Flags: Syn
	27	TCP Flags: Fin
	28	TCP Window size
	29	UDP Length

classes. To further test the validity of this result, we produced a silhouette graph of the k-means clusters as shown in Fig. 4b. Typically, a silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters. A value of 1 indicates that the points are very distant from neighboring clusters, a value of 0 informs that the points are not distant from other clusters while a negative value indicates that the points are erroneously placed in that cluster. From Fig. 4b, it is shown that a significant amount of points in all the 3 classes have a large silhouette value, greater than 0.6 (k-Means Clustering), indicating that the clusters are separated from neighboring clusters. This provides incentives to validate the quality of the formed k-means clusters. Further, the output of the EM clustering procedure on the same data file is shown in Fig. 4c. Similar to the k-means results, we can notice the formation of 3 distinct classes. These results relatively validate the proposed method by revealing that the scanning traffic originates from 3 distinct classes; we accept the null hypothesis that stated that the scanning techniques' traffic originates from 3 independent clusters.

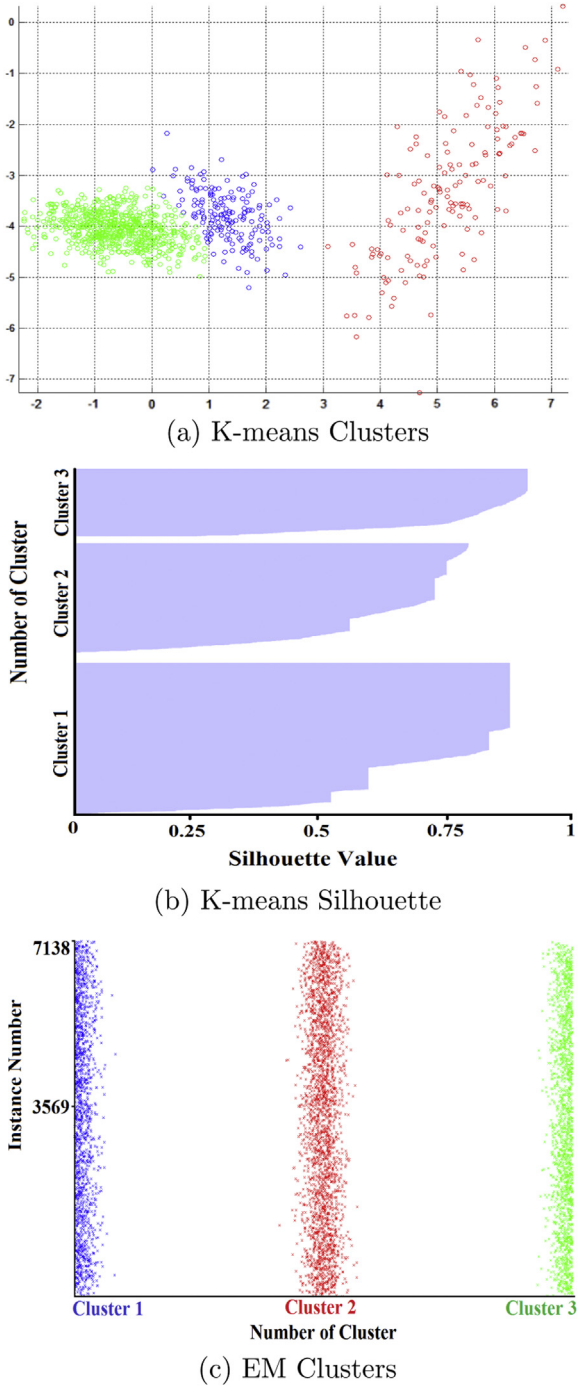

Fig. 4 – Method validation through unsupervised learning.

2.3. *Is the probing random?*

When probing activities occur, it would be interesting to possess insights related to how it is being generated; does the scanning occur in a random manner or does it follow a certain predefined pattern. Patterns existence provide cyber security intelligence about bots orchestration behavior. The latter is specifically important when trying to obtain inferences related to cyber scanning campaigns, such as the one reported in (Dainotti et al., 2012). To answer this question, we proceed as follows. For each distinct pair of hosts retrieved from the probing traffic, we test for randomness in the generated traffic using the Wald–Wolfowitz (also known as the runs) statistic test (Friedman and Rafsky, 1979). If the result is positive, we record it for that specific session and apply the test on the

remaining probing traffic. If the outcome is negative, we infer that the generated traffic follows a certain pattern. To capture the specific employed pattern, in this work, we model the probing traffic as a Poisson process[2] and retrieve the maximum likelihood estimate intervals (at a 95% confidence level) for the Poisson parameter λ that corresponds to that traffic. The choice to model the traffic as a Poisson distribution is motivated by (Li et al., 2008), where the authors observed that probe arrivals is coherent with that distribution.

2.4. How are the targets being scanned?

As revealed in (Dainotti et al., 2012; Internet Census, 2012), coordinated probing bots employ various strategies when probing their targets. These strategies could include IP-sequential (Bartlett et al., 2007), reverse IP-sequential (Heidemann et al., 2008), uniform permutation (Staniford et al., 2002b) or other types of permutations. In an attempt to capture the probing strategies, we execute the following. For each probing source in the probing traffic, we extract its corresponding distribution of target IPs. To differentiate between sequential and permutation probing, we apply the Mann–Kendall statistic test (Kendall, 1948), a non-parametric hypothesis testing approach, to check for monotonicity in those distributions. The rational behind the monotonicity test is that sequential probing will indeed induce a monotonic signal in the distribution of target IPs while permutation probing will not. Further, in this work, we set the significance level to 0.5% since a higher value could introduce false positives. To differentiate between (forward) IP-sequential and reverse IP-sequential, for those distributions that tested positive for monotonicity, we also record the slope of the distribution; a positive slope defines a forward IP-sequential strategy while a negative one implies a reverse IP-sequential strategy. For those distributions that tested negative for monotonicity (i.e., not a sequential strategy), we leverage the chi-square goodness-of-fit statistic test. The latter insight will inform us whether or not the employed strategy is a uniform permutation; if the test fails, then the employed strategy will be deemed as a permutation; uniform permutation otherwise.

2.5. Who is generating the probing activity?

When an enterprise fingerprint probing (i.e., detect activity and identify the technique), it is of value as well to infer about the 'machinery' behind the source of the probing. Specifically, it would be significant to pinpoint whether the scanning is generated by a scanning tool or a worm/botnet. One use case for this, is for instance, when an Internet Service Provider (ISP) notices that one of its customers is generating probing where that probing is generated by a worm/botnet. Keeping in mind that most probing activity is generated from non-spoofed Internet Protocol (IP) addresses (so the actual attacker/scanner can essentially get back the probing results), then the ISP can react on that and provide the suitable anti-malware solution to that specific customer.

[2] The modeling approach is not very significant but rather the consistency of adopting one approach on all the probing sources.

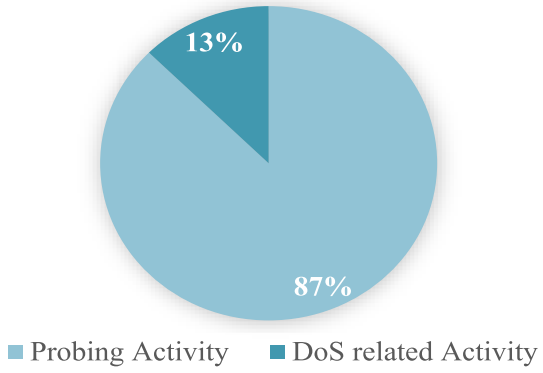

Fig. 5 – Sessions distribution.

From the two previous questions, we can infer those probing events that are random and monotonic. It is known that monotonic probing is a behavior of probing tools in which the latter sequentially scan their targets (IPs and ports) (Jajodia, 2012). Furthermore, for random events, the monotonic trend checking can help filter out traffic caused by the non-bot scanners (Li et al., 2011). Thus, we deem a probing source as leveraging a probing tool if their traffic is randomly generated and if they adopt a sequential probing strategy (i.e., including reverse IP-sequential); a worm/bot otherwise. Although in the current work we do not directly differentiate between scanning generated by worms or bots, however, future work would couple such information with real malware data (that we possess as well) to provide more accurate inferences, including the type of malware or the botnet orchestration pattern.

Note that the 'machinery' of the scan that was tackled in Sections 2.3, 2.4 and 2.5 aim to provide inferences and insights related to probing activities targeting a certain organization. Although the latter could be interesting to the organizations being scanned, we envision in future work that such proposed approach could be leveraged to automatically infer large-scale probing campaigns, such as the one analyzed in (Dainotti et al., 2012). More specifically, we intend to employ that latter coupled with other inferences that we are currently developing to cluster the probing sources that possess similar probing behaviors.

3. Empirical evaluation

We possess real darknet data that we are receiving on a daily basis from a trusted third party. Such traffic originates from the Internet and is destined to numerous/13 network sensors. The darknet sensors cover more than 12 countries and monitor around half a million dark IPs. The data mostly consists of unsolicited TCP, UDP and ICMP traffic. It might contain as well some DNS traffic. In a nutshell, darknet traffic is Internet traffic destined to routable but unused Internet addresses (i.e., dark sensors). Since these addresses are unallocated, any traffic targeting them is suspicious. Darknet analysis has shown to be an effective method to generate cyber threat intelligence (Bailey et al., 2005; Yegneswaran

et al., 2004). We use one week of data (55 GB), extracted from multiple/24 networks, that was collected during the duration of February 24th to March 3rd 2013, to empirically evaluate our approach. Darknet traffic is typically composed of three types of traffic, namely, scanning, backscattered and misconfiguration (Wustrow et al., 2010). Scanning arises from bots and worms while backscattered traffic commonly refers to unsolicited traffic that is the result of responses to denial of service attacks with spoofed source IP addresses. On the other hand, misconfiguration traffic is due to network/routing or hardware/software faults causing such traffic to be sent to the darknet sensors.

The first aim is to filter out misconfiguration data. We use a simple metric that records the average number of sources per destination darknet address. This metric should be significantly larger for misconfiguration than scanning traffic. However, although it differentiates misconfiguration from scanning traffic, it could include as well backscattered traffic as they also can possess a large average number of sources per destination (i.e, in case of a DoS). To cope with this issue, we observe, per the technique in (Wustrow et al., 2010), flags in packet headers, such as TCP SYN + ACK, RST, RST + ACK, ACK, etc., that resemble backscattered traffic (Wustrow et al., 2010). Subsequently, we filter out flows that lack that observation, deeming them as misconfiguration.

Second, we aggregate the connections into sessions using an approach similar to the first step algorithm by Kannan et al. (Kannan et al., 2006). We consider all those connections within T_{aggreg} of each other as part of the same session for a given pair of hosts. We used the same threshold, $T_{aggreg} = 100$ s, and found that this seems to correctly group the majority of connections between any given pair of hosts. For each day in our data set, we extracted 100 sessions for a total of 700 sessions.

We setup an experimental environment using Java and implemented Algorithms 1 and 2. For all the statistical test techniques, including DFA, Bhattacharyya distance, Mann-Kendall and Wald-Wolfowitz, we employed their MATLAB implementations (Little et al.; Cao; Fatichi; MathWorks).

We first applied the approach to attempt to differentiate between scanning and backscattered traffic (i.e., DoS related activity). Recall, that we have identified scanning as having the correlation statuses of Table 3. Fig. 5 represents how the 700 sessions were distributed and fingerprinted. It is shown that probing activity corresponds to 87% (612) of all the sessions. This scanning to backscattered traffic ratio is somehow coherent with other darknet studies (Wustrow et al., 2010). Note that in Fig. 5, DoS related activity was fingerprinted as such since it was shown from its DFA results that 13% (88) of the sessions possessed correlation statuses corresponding to either uncorrelated, 1/*f*-noise or Brownian noise. The fact that DoS related traffic demonstrated noise or Brownian noise is compliant with what was found in (Harder et al., 2006) when the authors performed DFA analysis on DoS traffic. To further validate such inference, we implemented the DoS detection algorithm by Moore et al. (Moore et al., 2006) and applied it on the 88 sessions. 77 sessions out of the 88 were detected as DoS related. Thus, with our approach, we have erroneously fingerprinted 11 sessions as DoS related (assuming the mentioned DoS detection algorithm did not produce any false positive). To understand why that occurred, we inspected the 11 sessions. 7 sessions out of the 11 possessed a DFA scaling exponent α ranging from 1.51 to 1.59, and accordingly were fingerprinted as Brownian noise (i.e., DoS related). However, after inspecting their traffic packets, they were shown to be a rare type of RPC scanning traffic. This suggests that one should not consider large α values as valid results or at least keep those incidents in a 'quarantine' for further automated post-processing. The remaining 4 sessions that were also erroneously fingerprinted seem to be misconfiguration that apparently, were not previously filtered as expected.

To evaluate the scanning fingerprinting capabilities of our approach, we experimented with Snort's sfPortscan preprocessor using the same 612 sessions that were previously fingerprinted as probing. sfPortscan (Roelker et al., 2004), a preprocessor plugin for the open source network intrusion and detection system Snort, provides the capability to detect TCP, UDP, and ICMP scanning. The sfPortscan preprocessor detects scans by counting RST packets from each perceived target during a predetermined timeout interval. Before declaring a scan, 5 events (i.e., RST packets) are required from a given target within a window. The sliding timeout window varies from 60 to 600 s by sensitivity level; at the highest level, an alert will be generated if the 5 events are observed within 600 s. We have chosen to compare our approach with Snort's sfPortscan preprocessor since Snort is one of the most broadly deployed intrusion detection/prevention technology worldwide and has become a de-facto standard.

We relied on sfPortscan's output as a baseline for our comparison. Snort's sfPortscan detected 590 scans. After a semi-automated analysis and comparison that was based on the logged scanning traffic flows (i.e., source and destination IP and port, protocol, and timestamp), we identified that all the 612 scans that our approach fingerprinted as probing activity include sfPortscan's 590 scans. Therefore, relative to this technique and experimenting with this specific data set, we confirm that our approach yielded no false negative. Moreover, according to the results, our proposed approach generated 22 sessions that are considered as false positive. It is worthy to pinpoint that our approach can detect certain types of scans that were not included at the time of the experiment, and by default, in Snort's sfPortscan definitions. These include scans from a single host to a single port on a single host, slow scans and a specific host scanning multiple ports on multiple hosts. In general, we claim that a certain limited, acceptable and a manageable number of false positives might occur

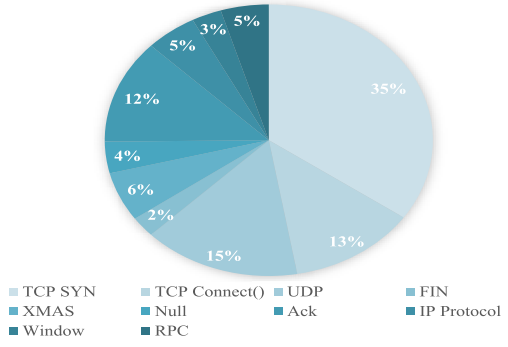

Fig. 6 – Probing techniques distribution.

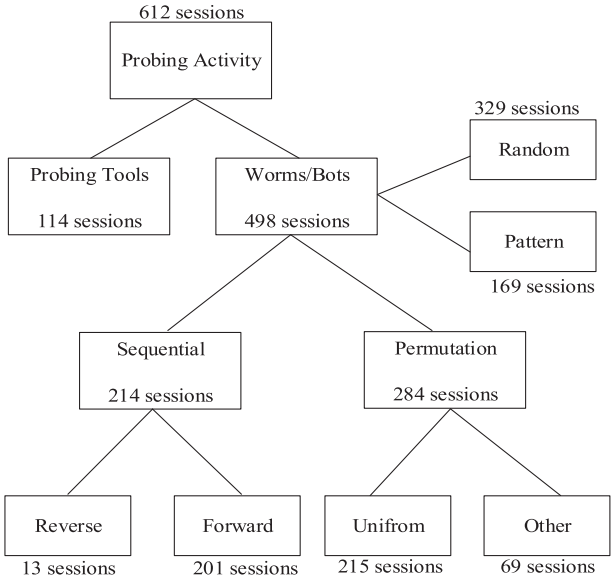

Fig. 7 – Probing activity dimensions analysis.

(taking into consideration the system that we compare our approach with). We need as well to consider Snort's sfPortscan false negatives and the different types of probing that our approach is able to fingerprint.

We next applied the proposed approach to identify which techniques were leveraged in the previous fingerprinted probing activity. Fig. 6 reveals that TCP SYN scanning leads with 35% (212) of all the sessions, followed by UDP, TCP connect() and ACK scanning. FIN, Xmas Tree, and Null scanning are typically considered as members of the 'stealth' scans because they send a single frame to a TCP port without any TCP handshaking or any additional packet transfers. They are relatively effective in evading firewall detection and they are often employed. The fact that the latter techniques were found to be among the least leveraged in the previous fingerprinted probing activity in our data set is quite puzzling.

We proceed by attempting to answer the questions that were raised in Sections 2.3, 2.4 and 2.5. We applied the proposed approach which yielded the output of Fig. 7. The results disclose that around 81% of the probing activity is being generated by worms or bots. Only 21% are being generated by probing tools. These percentages infer that leveraging real malware data, in a future study, could reveal substantial cyber security insights. The results also elaborate on the manner in which worms/bots generate their probing traffic. It is demonstrated that 66% of that probing traffic follows a random approach, while the remaining 34% follow a certain pattern when scanning their targets. Concerning the employed probing strategy, it is shown that 57% of the probing sources leveraged a permutation while the remaining adopted a sequential strategy when probing their targets. Of those that employed a permutation, 76% used a uniform permutation while 24% adopted other types of permutations. The majority (≈95%) of those that employed a sequential strategy were found to adopt a forward IP-sequential strategy while only 5% adopted a reverse IP-sequential strategy. The latter insights allows us to 1) track the probing activity that possess similar scanning patterns and strategies and perhaps attribute it to the same campaign, 2) apply similar mitigation steps to probing activity with similar patterns and techniques and 3) provide inferences, although not decisive, that the probing is being generated by an orchestrated botnet. Note that we also generate supplementary material related to the above mentioned probing activity (i.e., worm, bots, probing patterns) including geo-location information per real source, organization, ISP, city, region and country. However, we refrain from publishing those due to sensitivity/legal issues.

4. Evasion prevention

To fingerprint probing activity (i.e., detect activity and identify the technique), our approach, as stated in Section 2 and

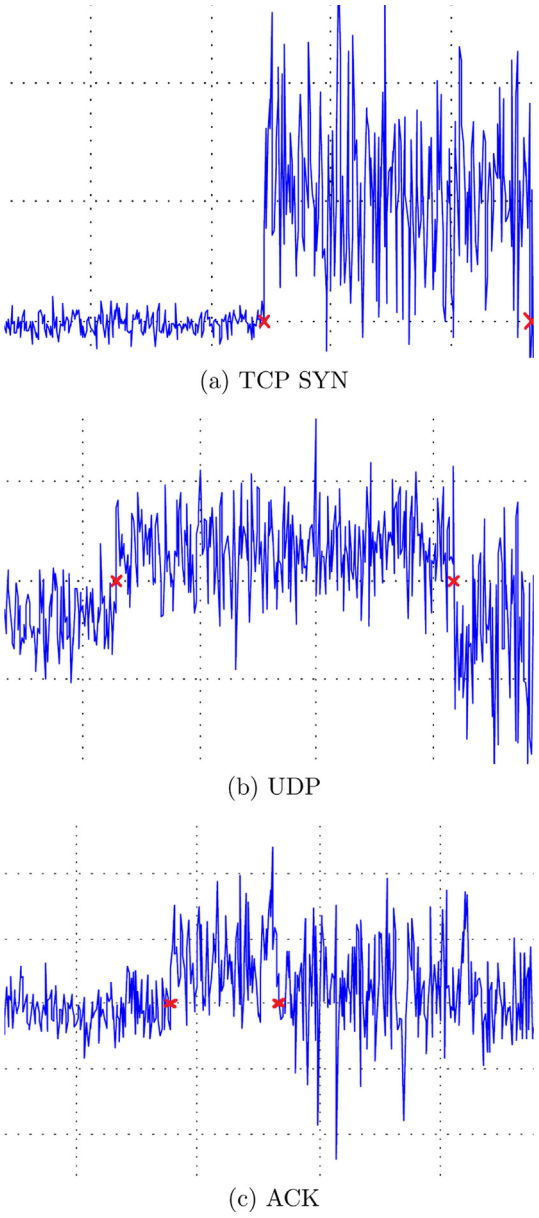

Fig. 8 – Approach evasion prevention using a change point detection technique.

evaluated in Section 3, leverages the DFA time series technique and operates on traffic distributions. However, it is realistic to pinpoint that the approach could be evaded in two ways. First, by a malicious attacker who deliberately injects a number of packets while performing his probing activity. Second, due to network fluctuations (i.e., delay), the distribution could be distorted. In both cases, our approach might erroneously miss the probing, attribute the probing to the wrong technique cluster or fail to identify the exact technique. We project that by formulating this problem as a time series change point detection problem, we can detect if and where the distribution has been susceptible to any sudden modifications. The time series change point detection problem has been excessively reviewed in the literature (Guralnik and Srivastava, 1999; Adams and MacKay, 2007). For the sake of this work, as a proof of concept, we selected the work from Adams and MacKay (2007) to experiment the effectiveness of such approach. We decided to leverage this specific work since it adopts a statistical approach which is coherent with the theme of our approach, is highly respected in its domain, and is directly applicable to our work that is related to time series distributions. Further, concerning its performance, in the worst-case, it is linear in space and time complexity according to the number of data points observed before the sudden fluctuation, which renders it practically viable. We employed the authors' MATLAB implementation, experimented with three different types of scanning traffic, namely, TCP SYN, UDP and ACK, and emulated a malicious attacker by injecting packets in the probing distributions using 'packit' (Bounds), a packet analysis and injection tool.

Fig. 8 shows how the TCP SYN, UDP and ACK scan distributions is being generated with a low frequency distribution. Suddenly, the malicious attacker injects random packets. The marked 'X's' on the Figure demonstrate how the change point detection technique was successfully able to detect the change. In the context of our work, to prevent distribution modifications, we can simply remove the distributions between the first marked 'X' and the second marked 'X' to retrieve the original distributions, namely the probing distributions. Although we admit that further experimentation should be undertaken to thoroughly authenticate the effectiveness of such change point detection techniques in providing evasion prevention to our approach, the obtained preliminary results seem to be promising and thus motivating.

5. Approach limitations

We acknowledge a number of limitations in our proposed approach. First, although in this work, the approach exhibited promising accuracy when evaluated using darknet (i.e., malicious) data, we have not tested the approach using normal two way traffic. Normal network traffic (i.e., benign http and ftp traffic for example) is known to be self-similar (i.e., possesses long traffic correlations). This directly affects the accuracy of our approach. We believe this point is manageable by applying pre-processing filtering mechanisms to filter out the benign traffic before applying our proposed approach. The payload analysis techniques in (Yegneswaran et al., 2005) seem viable to accomplish that. To verify this, we have executed the following experiment. We obtained a mixture of normal/malicious traffic data set from DARPA.[3] Subsequently, we executed the approach on that dataset after employing the payload filtering technique (Yegneswaran et al., 2005). The approach's accuracy scored around 83% in comparison with DARPA's ground truth information. We believe that such percentage of accuracy is motivating, keeping in mind that our approach is primarily targeted towards darknet analysis. Second, in its current state, our approach does not fingerprint ICMP scanning. The latter constitutes a significant portion of today's probing activity. We can overcome this limitation by analyzing/fingerprinting the distributions related to such traffic and performing experimentation validation as we did to other scanning techniques. Third, the approach does not differentiate between worms and bots probing. If accomplished, it will yield significant cyber security intelligence for attribution and mitigation purposes. This task is currently work in progress. Finally, bots probing orchestration is not yet confirmed. Coupled with probing patterns that we have generated in this work, it could be used for tracking cyber scanning campaigns and for effective mitigation.

6. Related work

In this section, we discuss few works related to probing detection/analysis using statistical approaches. Zhang et al. (2008) proposed a scan detection method based on a distributed cooperative model. Their technique is composed of feature-based detection, scenario-based detection and statistic-based detection. Their proposed architecture is decomposed into 5 layers (sensors, event generators, event detection agents, a fusion center and a control center) that collaborate to achieve the intended task. The technique's statistic-based detection employs predefined thresholds that allows the detection of both scan and denial of service attacks. A positive aspect of this work is that the proposed technique is well suited to distributed large-scale environments. However, the presented work was based on an illustrated described scenario and the authors did not discuss its applicability on real data samples. Bhuyan et al. (2012) presented the adaptive outlier based approach for coordinated scan detection (AOCD). First, the authors used the principal component analysis feature reduction technique to identify the relevant feature set. Second, they employed a variant of the fuzzy c-means clustering algorithm to cluster information. The authors tested their algorithm using different real-life datasets and compared the results against other available literature techniques. Their approach assumes that the target of the scanning is a set of contiguous addresses, which is not always the case. In another work, Baldoni et al. (2012) proposed a collaborative architecture where each target network deploys local sensors that send alarms to a collaborative layer. This, in turn, correlates this data with the aim of (1) identifying coordinated cyber scanning activity while (2) reducing false positive alarms and (3) correctly separating groups of attackers that act concurrently on overlapping targets. The soundness

[3] http://tinyurl.com/lzbdd7h.

of the proposed approach was tested on real network traces. Their proposed system is designed to leverage information coming from various network domains to detect distributed scanning. Hence, the collaborative layer appears to be ineffective when the adversary is acting only against one network domain. In a more general work, Dainotti et al. (2012) presented the measurement and analysis of a 12-day world-wide cyber scanning campaign targeting VoIP (SIP) servers. The authors used darknet/telescope data collected at the UCSD network telescope to exclusively focus on the analysis and reporting of that SIP scanning incident.

Our work that is motivated by the positive DFA results from our previous work (Bou-Harb et al., 2013b) is different from the above as it does not rely on identifying the scanning source and is independent from the scanning strategy. Further, the proposed approach does not rely on a certain predefined alert threshold, the transport protocol used or the number of probed destinations and ports. Moreover, we attempted to go further than detection by analyzing probing traffic dimensions, namely, employed technique, monotonicity and randomness.

7. Conclusion

This paper presents a new method to fingerprint probing activity. It aims at detecting the cyber scanning activity and identifying the exact technique that was employed in the activity. Further, it analyzes certain probing traffic dimensions such as monotonicity and randomness to generate inferences related to the 'machinery' of the scan (i.e., probing tools Vs worms/bots), the approach of the scanning (i.e., randomness Vs. probing patterns) and the employed probing strategy (i.e., sequential Vs. permutation). The paper leverages and employs several statistical techniques to achieve the required tasks. Empirical evaluations performed using real darknet traffic showed that the extracted inferences exhibit promising accuracy. As for future work, we are in the process of working on coupling part of the inferences that we have generated in this work with real malware data to provide more accurate and impactful inferences, including the type of probing malware and the botnet orchestration pattern.

Acknowledgment

The authors are grateful for Concordia University and the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this work. The first author is supported by the Alexander Graham Bell Canada Graduate Scholarship (CGS) from NSERC.

REFERENCES

Adams Ryan Prescott, MacKay David JC. Bayesian online changepoint detection; 2007. Cambridge, UK.

ArsRechnica. Facebook computers compromised. http://tinyurl.com/cwmvxrv.

Bailey Michael, Cooke Evan, Jahanian Farnam, Nazario Jose, Watson David. The internet motion sensor: a distributed blackhole monitoring system. In: Proceedings of the 12th ISOC Symposium on Network and Distributed Systems Security (SNDSS); 2005. pp. 167–79.

Baldoni R, Di Luna G, Querzoni L. Collaborative detection of coordinated port scans. Technical report, http://www.dis.uniroma1.it/~midlab; 2012.

Bartlett Genevieve, Heidemann John, Papadopoulos Christos. Understanding passive and active service discovery. In: Proceedings of the 7th ACM SIGCOMM conference on Internet Measurement. ACM; 2007. pp. 57–70.

Bhuyan MH, Bhattacharyya DK, Kalita JK. Surveying port scans and their detection methodologies. Comput J 2010;54(10):1565–81.

Bhuyan MH, Bhattacharyya DK, Kalita JK. Aocd: an adaptive outlier based coordinated scan detection approach. Int J Netw Secur 2012;14(6):339–51.

Bilge Leyla, Dumitras Tudor. Before we knew it: an empirical study of zero-day attacks in the real world. In: Proceedings of the 2012 ACM conference on Computer and Communications Security, CCS '12. New York, NY, USA: ACM; 2012. pp. 833–44.

Bou-Harb E, Debbabi M, Assi C. Cyber scanning: a comprehensive survey. Commun Surv Tutorials, IEEE 2013a;(99):1–24.

Bou-Harb E, Debbabi M, Assi C. On detecting and clustering distributed cyber scanning. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International; 2013. pp. 926–33.

Bounds Darren. Packit – packet analysis and injection tool. http://linux.die.net/man/8/packit.

Cao Yi. Bhattacharyya Distance measure for pattern recognition. http://tinyurl.com/bveualz.

Chung Yoo. Distributed denial of service is a scalability problem. SIGCOMM Comput. Commun Rev January 2012;42(1):69–71.

Computer World. Google: Phishers stole e-mail from U.S. officials, others. http://tinyurl.com/cav4rwj.

Dainotti A, King A, Claffy K, Papale F, Pescap A. Analysis of a "/0" Stealth Scan from a Botnet. In: Internet Measurement Conference (IMC); Nov 2012.

Daly MK. Advanced persistent threat. Usenix; Nov, 4, 2009.

Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. J Royal Stat Soc Ser B Methodol; 1977:1–38.

Fatichi Simone. Mann–Kendall Test. http://tinyurl.com/cstvpwa.

Fing, the ultimate network toolkit. http://www.overlooksoft.com/fing.

Forbes. Citibank reveals one percent of credit card accounts exposed in hacker intrusion. http://tinyurl.com/7jxgxqz.

Friedman Jerome H, Rafsky Lawrence C. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. Ann Stat; 1979:697–717.

Fukuda K, Hirotsu T, Akashi O, Sugawara T. Correlation among piecewise unwanted traffic time series. In: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008; 2008. pp. 1–5.

Guralnik Valery, Srivastava Jaideep. Event detection from time series data. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 1999. pp. 33–42.

The Wall Street Journal. Hackers targeted Twitter user data. http://tinyurl.com/a9tkbn5.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka data mining software: an update. ACM SIGKDD Explor Newsl 2009;11(1):10–8.

Harder U, Johnson MW, Bradley JT, Knottenbelt WJ. Observing internet worm and virus attacks with a small network telescope. Electron Notes Theor Comput Sci 2006;151(3):47–59.

Heidemann John, Pradkin Yuri, Govindan Ramesh, Papadopoulos Christos, Bartlett Genevieve, Bannister Joseph. Census and survey of the visible internet. In: Proceedings of the 8th ACM SIGCOMM conference on Internet Measurement. ACM; 2008. pp. 169–82.
Hu K, Ivanov PC, Chen Z, Carpena P, Stanley HE. Effect of trends on detrended fluctuation analysis. Phys Rev E 2001;64(1):011114.
Internet Census. Port scanning/0 using insecure embedded devices http://tinyurl.com/c8af8lt; 2012.
Jacobson V, Leres C, McCanne S. The tcpdump manual page. Berkeley, CA: Lawrence Berkeley Laboratory; 1989.
Jajodia Sushil. Cyber situational awareness: Issues and research; 2012.
Kailath Thomas. The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans Commun Technol 1967;15(1):52–60.
Kannan Jayanthkumar, Jung Jaeyeon, Paxson Vern, Koksal Can Emre. Semi-automated discovery of application session structure. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. ACM; 2006. pp. 119–32.
Kendall Maurice George. Rank correlation methods; 1948.
k-Means Clustering. http://www.mathworks.com/help/stats/k-means-clustering.html.
Li Zhichun, Goyal Anup, Chen Yan. Honeynet-based botnet scan traffic analysis. In: Botnet detection. Springer; 2008. pp. 25–44.
Li Zhichun, Goyal Anup, Chen Yan, Paxson Vern. Towards situational awareness of large-scale botnet probing events. IEEE Trans Inform Forensics Secur 2011;6(1):175–88.
Little M, McSharry P, Moroz I, Roberts S. Nonlinear, biophysically-informed speech pathology detection. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 proceedings, vol. 2. p. II.
Lyon GF. Nmap network scanning: the official nmap project guide to network discovery and security scanning author: Gordon fyodor l; 2009.
MacQueen James. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on Mathematical Statistics and Probability, vol. 1; 1967. p. 14. California, USA.
ABC news. Major US newspapers allege Chinese hack attack. http://tinyurl.com/bekao8k.
MathWorks. Run test for randomness. http://tinyurl.com/d6gtykz.
Matos JAO, Gama S, Ruskin HJ, Sharkasi AA, Crane M. Time and scale hurst exponent analysis for financial markets. Phys Stat Mech Appl 2008;387(15):3910–5.
Moore David, Shannon Colleen, Brown Douglas J, Voelker Geoffrey M, Savage Stefan. Inferring internet denial-of-service activity. ACM Trans Comput Syst (TOCS) 2006;24(2):115–39.
National Security Council. http://www.whitehouse.gov/cybersecurity.
Panjwani S, Tan S, Jarrin KM, Cukier M. An experimental evaluation to determine if port scans are precursors to an attack. In: Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on; June–1 July 2005. pp. 602–11.
Peng C-K, Buldyrev SV, Havlin S, Simons M, Stanley HE, Goldberger AL. Mosaic organization of DNA nucleotides. Physp Rev E Feb 1994;49:1685–9.
Priestley MB. Spectral analysis and time series; 1981.
Internet Storm Center. Reports of a distributed injection scan. http://isc.sans.edu/diary.html?storyid=14251.
Roelker Daniel, Norton Marc, Hewlett Jeremy. sfportscan http://projects.cs.luc.edu/comp412/dredd/docs/software/readmes/sfportscan; 2004.
jNetPcap. Sli Technologies. http://jnetpcap.com/userguide.
Snort. Available at: http://www.snort.org.
Staniford S, Hoagland JA, McAlerney JM. Practical automated detection of stealthy portscans. J Comput Secur 2002a;10(1/2):105–36.
Staniford Stuart, Paxson Vern, Weaver Nicholas. How to own the internet in your spare time. In: Proceedings of the 11th USENIX Security Symposium, vol. 8; 2002. pp. 149–67.
Symantec. W32.Stuxnet Dossier. http://tinyurl.com/36y7jzb.
Techcrunch. Hack attack: Sony confirms playstation network outage caused by 'external intrusion'. http://tinyurl.com/6cbcldv.
The Next Web. Senate website, CIA.gov reportedly hacked. http://tinyurl.com/7y8dv5b.
Woodhead Simon. Monitoring bad traffic with darknets. Netw Secur 2012;2012(1):10–4.
Wustrow Eric, Karir Manish, Bailey Michael, Jahanian Farnam, Huston Geoff. Internet background radiation revisited. In: Proceedings of the 10th annual conference on Internet Measurement. ACM; 2010. pp. 62–74.
Yegneswaran Vinod, Barford Paul, Plonka Dave. On the design and use of internet sinks for network abuse monitoring. In: Recent advances in intrusion detection; 2004.
Yegneswaran Vinod, Barford Paul, Paxson Vern. Using honeynets for internet situational awareness. In: Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets IV); 2005. pp. 17–22.
Zambon Emmanuele, Bolzoni Damiano. Network intrusion detection systems. http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Zambon.pdf.
Zhang W, Teng S, Fu X. Scan attack detection based on distributed cooperative model. In: Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on. IEEE; 2008. pp. 743–8.

Elias Bou-Harb is a network security researcher pursuing his Ph.D. in Computer Science at Concordia University, Montreal, Canada. Previously, he has completed his M.A.Sc. degree in Information Systems Security at the Concordia Institute for Information Systems Engineering. He is also a member of the National Cyber Forensic and Training Alliance (NCFTA), Canada. His research interests focus on the broad area of cyber security, including operational cyber security for critical infrastructure, LTE 4G mobile network security, VoIP attacks and countermeasures and cyber scanning campaigns. He is supported by the prestegious Alexander Graham Bell Canada Graduate Scholarship (CGS) from the Natural Sciences and Engineering Research Council of Canada (NSERC).