Computer Networks 94 (2016) 327–343

Contents lists available at ScienceDirect

Computer Networks

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

A novel cyber security capability: Inferring Internet-scale infections by correlating malware and probing activities

Elias Bou-Harb*, Mourad Debbabi, Chadi Assi

The National Cyber Forensics and Training Alliance (NCFTA) Canada & Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 17 December 2014
Revised 24 October 2015
Accepted 5 November 2015
Available online 10 November 2015

Keywords:
Probing
Malware
Darknet preprocessing
Big data correlation
Cyber security
Cyber intelligence

ABSTRACT

This paper presents a new approach to infer worldwide malware-infected machines by solely analyzing their generated probing activities. In contrary to other adopted methods, the proposed approach does not rely on symptoms of infection to detect compromised machines. This allows the inference of malware infection at very early stages of contamination. The approach aims at detecting whether the machines are infected or not as well as pinpointing the exact malware type/family. The latter insights allow network security operators of diverse organizations, Internet service providers and backbone networks to promptly detect their clients' compromised machines in addition to effectively providing them with tailored anti-malware/patch solutions. To achieve the intended goals, the proposed approach exploits the darknet Internet space and initially filters out misconfiguration traffic targeting such space using a probabilistic model. Subsequently, the approach employs statistical methods to infer large-scale probing activities as perceived by the dark space. Consequently, such activities are correlated with malware samples by leveraging fuzzy hashing and entropy based techniques. The proposed approach is empirically evaluated using a recent 60 GB of real darknet traffic and 65 thousand real malware samples. The results concur that the rationale of exploiting probing activities for worldwide early malware infection detection is indeed very promising. Further, the results, which were validated using publically available data resources, demonstrate that the extracted inferences exhibit noteworthy accuracy and can generate significant cyber security insights that could be used for effective mitigation.

1. Introduction

Today, the safety and security of our society is significantly dependent on having a secure infrastructure. This infrastructure is largely controlled and operated using cyberspace. Although tremendous efforts have been carried out to protect the cyberspace from diverse debilitating, intimidating and disrupting cyber threats, such space continues to host highly sophisticated malicious entities. The latter could be ominously leveraged to cause drastic Internet-wide and enterprise impacts by means of large-scale probing campaigns [1], distributed denial of service attacks [2], advanced persistent threats [3] and spamming botnets [4]. According to Panda Security, a staggering 33% of worldwide Internet machines are infected by malware [5]. Moreover, McAfee records over 100 thousand new malware samples every day; a momentous 69 threats every minute or around one new threat every second [6].

Network security operators of private and governmental organizations, Internet Service Provides (ISPs) and content delivery networks as well as backbone networks face, on a daily basis, the crucial challenge of dealing with their clients' malware-infected machines. The latter not only hinders the clients' overall experience and productivity but also

* Corresponding author. Tel.: +1 5146495049.
E-mail address: e_bouh@encs.concordia.ca (E. Bou-Harb).

http://dx.doi.org/10.1016/j.comnet.2015.11.004
1389-1286/

jeopardizes the entire cyber security of the provider (i.e., causing vulnerabilities or opening backdoors in the internal network). Further, it significantly degrades the provided quality of service since the compromised machines will most often cause excessive increase in bandwidth that could be rendered by extreme Peer to Peer (P2P) usage, spamming, command-and-control communications and malicious Internet downloads. Additionally, if providers' networks were used to trigger, for instance, a malware-orchestrated spamming campaign, then such providers could as well encounter serious legal issues for misusing their infrastructure (i.e., for example, under the Canadian House Government Bill C-28 Act [7]). Consequently, this will immensely adversely affect the operators' business, reliability and reputation.

Thus, network security operators are interested in possessing a cyber security capability that generates inferences and insights related to their clients' malware-infected machines. It is significant for them to be able to pinpoint such machines in addition to extract intelligence related to the exact malware type/family. The latter will facilitate the distribution of suitable and tailored anti-malware solutions to those compromised clients.

Indeed, this cyber security capability should possess the following requirements. First, it should be prompt; it must possess the ability to detect the infection as early as possible in an attempt to thwart the creation of botnets and to limit the sustained possible collateral damage and any symptoms of infection. Second, it should be cost-effective; the approach should not overburden the provider with implementation scenarios and their corresponding supplementary costs. In fact, this last point is extremely imperative and decisive; ISPs are frequently accused of ignoring their clients' malware infections because the task to detect and disinfect them is tedious, prolonged and undoubtedly expensive [8,9]. This paper, which extends our previous work [10], elaborates on such a cyber security capability that satisfies the mentioned requirements. Specifically, we frame the paper's contributions as follows:

- Proposing a new probabilistic model to preprocess telescope/darknet data to prepare it for effective use. The aim is to fingerprint darknet misconfiguration traffic and subsequently filter it out. The model is advantageous as it does not rely on arbitrary cut-off thresholds, provide separate likelihood models to distinguish between misconfiguration and other malicious darknet traffic and is independent from the nature of the source of the traffic.
- Proposing a new approach to infer Internet-scale malware-compromised machines. The approach aims at detecting such machines as well as identifying their exact infection type. The approach achieves its aims without recording or analyzing the symptoms of infection (i.e., spamming, excessive bandwidth usage, etc.), which renders it efficient from both space and processing perspectives. Further, it exploits probing activities to attain early detection of contamination incidents in addition to requiring no implementation at the providers' premises, eliminating the cost burden.
- Leveraging the darknet Internet space, around half a million routable but unallocated IP addresses, which permits the observation and identification of worldwide probing activities and thus malware-infected machines, without requiring any providers' aid or information.
- Correlating malware and probing network activities to achieve the intended goals by employing numerous statistical, fuzzy hashing and entropy based techniques.
- Evaluating the proposed approach using a recent 60 GB of real darknet traffic and 65 thousand real malware samples.

The remainder of this paper is organized as follows. In the next section, we review the related work. In Section 3, we elaborate on the proposed approach. Specifically, we explain its rationale, describe its components and present the leveraged mechanism, methods and techniques. We empirically evaluate the approach in Section 4. In Section 5, we pinpoint some limitations of the proposed approach. Finally, concluding remarks and some future work are disclosed in Section 6.

2. Related work

In this section, we review some literature work related to malware and probing correlation analysis. Further, we briefly highlight two approaches that are adopted in the industry for the purpose of detecting malware-infected machines. Additionally, we pinpoint several proposed methods for inferring worm infections. Finally, we present several research efforts in the areas of active detection of malicious machines as well as malware signature generation.

Nakao et al. [11] were among the first to exploit the idea of correlating malware and probing activities to detect zero-day attacks. The authors leveraged the nicter framework [12] to study the inter-relations between those two activities. They developed scan profiles by observing the dark space and correlated them with malware profiles that had been generated in a controlled environment. Their work seems limited in a number of points. First, the authors did not validate the accuracy of the extracted probing activities from the dark space. Second, the extracted profiles were based on few textual network and transport-layer features, where the actual correlation engine's mechanism was obscured. Third, their experiments were based on only one malware sample. In contrary, in this work, we first apply a validated statistical approach to accurately extract probing activities from darknet traffic. Second, in a first attempt ever, we correlate probing and malware activities by applying fuzzy hashing and information theoretical based techniques on the entire network traffic that was generated by those activities. Third, our experiments involve around 65 thousand malware samples. Fourth, the aim of this work differs as it is rendered by the capability to provide network security operators, worldwide, with the ability to rapidly and cost-effectively detect their clients' infections, without requiring the providers to maintain an implementation nor provide any aid or disclose any sensitive network related information. In an another closely related work, Song et al. [13] carried out correlation analysis between 10 spamming botnets and malware-infected hosts as observed by honeypots. They disclosed that the majority of the spamming botnets have been infected by at least four different malware. The authors as well developed methods to identify which exact malware type/family has been the cause of contamination. Our work differs from this work as we are

correlating probing activities rather than spamming for early infection detection. Further, we are leveraging the darknet space instead of honeypots to extract Internet-scale cyber security intelligence. In a slightly different work, Eto et el. [14] proposed a malware distinction method based on scan patterns by employing spectrum analysis. The authors stated that by observing certain probing patterns, one can recognize the similarities and dissimilarities between different types of malware. The authors noted that the latter could be used as a fingerprint to effectively infer infection. The authors however, did not perform any correlation but rather limited their work to observation and analysis.

The industry has also developed approaches to identify malware compromised machines. For instance, True Internet, one of Thailand's largest ISPs, had adopted a behavioral approach to infer its clients infected machines. Their approach monitors the symptoms of infection, including but not limited to, spamming, excessive P2P usage and Denial of Service (DoS) attempts, and subsequently triggers an alert towards a controller, which then automatically quarantine the client. Although such approaches might be effective, they are typically late in detection, which might cause serious vulnerabilities within the provider. Moreover, they are usually not cost-effective as the provider ought to purchase and maintain other detection systems. Another example would be NetCologne, an ISP and cable provider in Germany, that took a different approach to automate how it deals with subscribers that are infected with malware. NetCologne setup and maintained a honeypot; infected machines often attempt to attack other computers on the same network and hence the honeypot is an accessible target that allows the identification of compromised machines. While this approach seems practical in detecting infected machines, it is neither cost-effective since the provider needs to implement and maintain the honeypot nor it is able to identify the exact malware type/family that had contaminated those machines[1]. Moreover, honeypot evasion approaches are known to be effective [15] and are often adopted by sophisticated malware.

In the context of inferring worm infections, Gu et al. [16] presented the design and implementation of BotHunter, a perimeter monitoring system for real-time detection of Internet malware infections. The core of BotHunter is rendered by a correlation engine that performs alert consolidation and evidence trail gathering for investigating numerous infections. In a slightly different work, Whyte et al. [17] correlated Domain Name System (DNS) queries with outgoing connections from an enterprise network to detect worms propagation attempts. Through empirical evaluations, their proposed approach yielded efficient and accurate results and enabled automatic containment of worm propagation at the network egress points. In an alternative work, schechter et al. [18] presented a hybrid approach to detect scanning worms. Their approach, which is based on sequential hypothesis testing and rate limiting algorithms, demonstrated low false positive rates when evaluated in a real operational network environment. A possible drawback of the previous approaches is that they require to be deployed and maintained at the perimeter of the enterprise network to be effective.

In the area of active detection of malicious machines, AutoProbe was recently proposed by Xu et al. [19] to automatically generate effective and efficient fingerprints of remote malicious servers. The core aim of AutoProbe is to employ those generated fingerprints to actively scan networks, including, the Internet, in order to detect malicious servers. By employing malware execution, probe generation, classification function construction, and probing activities, AutoProbe was shown to generate highly accurate network fingerprints for malicious servers' probing. Another major contribution of AutoProbe is that it is able to generate fingerprints when no live command & control server is known. In a similar vibe, CyberProbe was proposed by Nappa et al. [20] to detect Internet-scale malicious servers and compromised hosts. CyberProbe operates by initially actively sending probes to remote hosts and subsequently examining their responses, to determine whether the remote hosts are malicious or not. Large scale evaluations have been conducted by the authors that demonstrated that CyberProbe is indeed fast, cheap, easy to deploy, and effective in achieving Internet-scale detection. Our proposed work differs from the latter two approaches in three core points. First, we do not employ active scanning in an attempt to reach out to malicious servers. Rather, we passively infer probes from Internet-scale malicious hosts as perceived by the dark IP space. Thus, our approach could be considered as less intrusive to Internet hosts. Further, active probing as employed in AutoProbe and CyberProbe would be illegal in a number of jurisdictions, including some in North America, for example, under the Canadian Criminal Code of Canada. Second, our correlation mechanism exploits the extracted probing activities from the dark IP space to fuse them with malware probing activities to attain *prompt* detection. Third, the correlation mechanism itself is based on various entropy measures and statistical methods, which have never been attempted before on two different types of network data to detect Internet-scale infections. Last but not least, in the context of malware signature generation, Rafique and Caballero. [21] proposed FIRMA, a tool that aims at (1) clustering malware binaries into families and (2) generating network signatures for each family, regardless of the type of traffic that the malware uses. Empirical evaluations using a significant number of malware binaries demonstrated that FIRMA is highly efficient and possesses notable clustering and signature generation accuracy. In a slightly similar work, Perdisci et al. [22] tackled the same problem of generating malware network signatures by exclusively focusing on malware that only use the HTTP protocol. Our aim in this work is not to devise approaches to generate malware signatures from network traffic, but rather to correlate probing traffic extracted from the darknet and malware samples to infer Internet-scale infections.

3. Proposed approach

In this section, we elaborate on the proposed approach as depicted in Fig. 1. Specifically, we discuss its rationale, describe its components and present its employed mechanism, methods and techniques.

[1] We argue that the provider, by solely observing the network traffic generated by the malicious host prior to capturing the malware sample through the honeypot, might not be able to directly infer the exact infection/malware type.

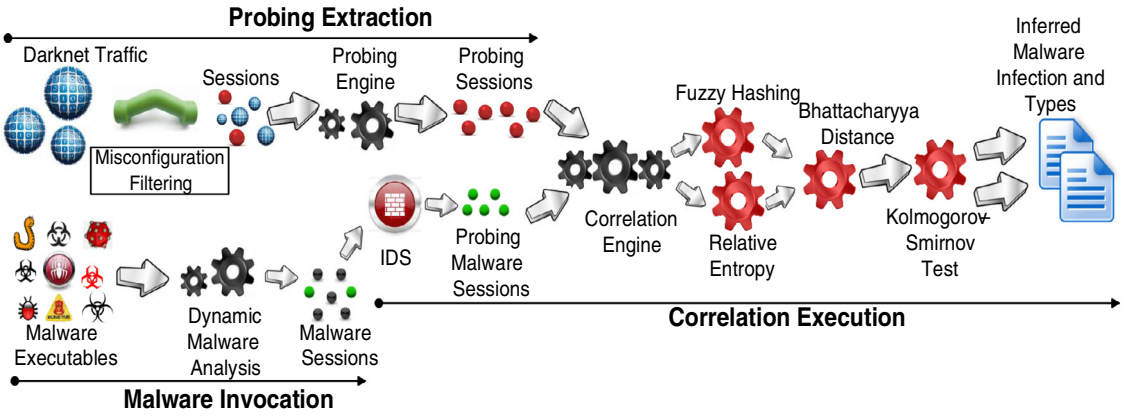


Fig. 1. The components of the proposed approach.

The rationale behind the proposed approach stems from the need to detect the infection at early stages of contamination. In this context, probing or scanning activities are known to be the very first symptoms of infection [23,24]. On the other hand, the Internet dark space (i.e., dark sensors) has shown to be an effective method to generate Internet-scale cyber threat intelligence [25–27]. In brief, darknet traffic is Internet traffic destined to routable but unused Internet addresses. Since these addresses are unallocated, any traffic targeting them is deemed as suspicious. Thus, in a nutshell, the proposed approach aims at extracting probing activities as received by a darknet and subsequently correlating them with malware samples. By leveraging geo-location information, the approach strives to generate insights related to worldwide compromised machines in addition to identifying their exact infected malware type/family. The approach is envisioned to be operated by a central authority, for example, an incident response center, where the latter will distribute, in real-time, the extracted inferences to concerned parties. In the sequel, we elaborate on each component of the proposed approach in accordance with Fig. 1.

3.1. Misconfiguration filtering

As mentioned, Internet traffic destined to routable yet unallocated IP addresses is commonly referred to as telescope or darknet data. Such malicious traffic is frequently, abundantly and effectively exploited to generate various cyber threat intelligence related but not limited to, scanning activities, distributed denial of service attacks and malware identification. However, such data typically contains a significant amount of misconfiguration traffic caused by network/routing or hardware/software faults. The latter immensely affects the purity of darknet data, which hinders the accuracy of detection algorithms that operate on such data in addition to wasting valuable storage resources. This section proposes a probabilistic model to preprocess telescope data to prepare it for effective use. The aim is to fingerprint darknet misconfiguration traffic and subsequently filter it out. The model is advantageous as it does not rely on arbitrary cut-off thresholds, provide separate likelihood models to distinguish between misconfiguration and other darknet traffic and is independent from the nature of the source of the traffic. To the best of our knowledge, the proposed model renders a first attempt ever to tackle the problem of preprocessing darknet traffic.

In a nutshell, the model aims at computing an access probability distribution for each darket IP address, derived across all remote sources that target those dark IPs. Thus, the model initially estimates the degree to which access to a given dark IP address is unusual. The model further considers the number of distinct dark IP addresses that a given remote source has accessed. Subsequently, the joint probability is formulated, computed and compared. If the probability of the source generating a misconfiguration is higher than that of the source being malicious (i.e., scanning or backscattered), then the source is deemed as misconfigured, subsequently flagged, and its corresponding generated darknet flows are filtered out.

Let $D = \{d_1, d_2, d_3, \ldots\}$ represent the set of darknet IP addresses and D_i a subset of those accessed by source s_i. First, the model captures how unusual the accessed destinations are. The idea behind this metric stems from the fact that misconfigured sources access destinations that have been accessed by few other sources [28,29]. Thus, the model estimates the distribution of a darknet IP d_i being accessed by such a source as

$$P_{misc}(d_i) = \frac{n_s(d_i)}{\sum_{\forall d_j \in D} n_s(d_j)} \quad (1)$$

where $n_s(d_i)$ is the number of sources that have accessed d_i. In contrary, a malicious darknet source[2] will access a destination at random. Typically, defining a suitable probability distribution to model the randomness of a malicious source targeting a specific darknet destination is quite tedious and unsystematic; often a simplistic assumption is applied to solve this issue. In this context, a very recent work by Durumeric et al. [30] assumed that darknet sources will probe their targets following a random uniform distribution. By adopting that assumption, one can model the probability

[2] In a given set of packets received on the dark IP space, malicious darknet sources represent malicious Internet hosts while darknet addresses represent the destinations in those packets.

of a darknet destination accessed by a malicious source as

$$P_{mal}(d_i) = \frac{1}{|D|} \quad (2)$$

where $|D|$ represents the dark IP space.

However, in this work, we thought it would be beneficial and more precise to verify the soundness of that assumption before completing the description of the proposed model. To successfully achieve the latter, we perform three experiments using simulation, emulation and real malicious darknet traffic. The first experiment is rendered by a simulation executed using Opnet Modeler[3]. The simulation is comprised of a probing source and 100 probing destinations/targets. The source and the destinations are represented using commodity machines. The probing source is instructed to generate three types of probing towards the targets, namely, TCP SYN, UDP and ACK scanning, for a duration of 15 min. The latter are typically common types of probing activities [23]. Fig. 2a represents the outcome of this experiment. According to the assumption given by the uniform distribution, each target should theoretically receive around 100 packets. On average, the goodness-of-fit was around 76%, significantly below the acceptable 95% threshold [31]. To further assess the accuracy of the uniform distribution in modeling the target access distribution, we executed a second experiment. In this experiment, we employed Nmap, an open source scanning tool, to emulate the probing traffic. The emulation environment included a probing machine running the tool in addition to 5 target Virtual Machines (VMs). The probing machine repetitively executed Null, FIN and Xmas scanning towards the virtual machines for a duration of 10 min. The depiction of Fig. 2b clearly demonstrates that the uniform distribution does not appear to be a good fit for such traffic. Although the latter simulation and emulation experiments demonstrated that the assumption of modeling malicious packets using the random uniform distribution is not quite accurate, we thought it would be interesting and more realistic to utilize real darknet traffic to assess that hypothesis. Subsequently, we extract one day of darknet data (≈8 GB) from April, 2014, divide it into slots of 4 h and monitor the number of probing packets targeting 20 darknet destinations over the 6 slots. Fig. 2c illustrates the outcome of this experiment. According to the uniform distribution, each target should theoretically receive around 120 probing packets. On average, the goodness-of-fit was less than 78%. Such results from all the above experiments concur that the assumption of modeling darknet malicious packets towards darknet destinations by adopting the uniform distribution is not accurate; there indeed exists an imperative requirement to determine the most appropriate probability distribution model that best fits malicious darknet packets.

Recall, that the latter requirement needs to be fulfilled to accurately model darknet destinations targeted by a malicious source in order to continue the elaboration of the proposed model. To achieve that, we proceed by performing another set of experiments. Such experiments also rely on simulation, emulation and the utilization of real darknet traffic, and their corresponding setup environments is quite similar to those of the previous experiments. One difference is that we execute the experiments for 10 times and average the outcome. Another difference is related to the number of targeted destinations, in which we increase the latter number, for accuracy purposes, from 100 to 500 targeted destinations, 5 to 12 VMs and 20 to 110 monitored darknet destination, in the simulation, emulation and real traffic experiments, respectively. Please note that this increase does not affect the outcome while simultaneously providing a clearer illustration that the inferred distribution provides a better modeling for darknet malicious packets. Various experiments that we have performed (omitted here) validated the latter. To determine the best fit model, we utilized a generic Matlab parametric probability distributions' fitting function and calculated the Bayesian Information Criterion (BIC) [32] for each of the models in relation to our data. The latter metric is an established criterion for model selection among a finite set of models. Typically, the lower the BIC is, the better is the fit. Fig. 3a–c demonstrate the outcome probability density estimations in the three experiments. In the simulation experiment, it is evident that the Gaussian or the Normal distribution provides the best fit for modeling malicious darknet packets. In the second experiment, namely, the emulation experiment, it is revealed that the Pareto distribution provides the best fit; this is quite interesting as such distribution is often employed in modeling normal (i.e., benign) network traffic. Fig. 3b further demonstrates that the inverse Gaussian distribution, which is analogous to the Normal distribution, also shows a positive BIC, resembling a good fit. Finally, the third experiment that employs real darknet traffic undoubtedly validates that the Gaussian distribution provides the best fit to model malicious packets targeting darknet destinations. It is worthy to note that neither of the three experiments portray the uniform distribution between the top 6 models providing a best fit. Such results concur that it is relatively accurate and practical to adopt the Gaussian distribution instead of the uniform distribution to model targeted darknet destinations.

Thus, at this point, Eq. (2) could be adjusted to match the probability density function of a Normal distribution. Consequently, we can now model the probability of a darknet destination accessed by a malicious source as

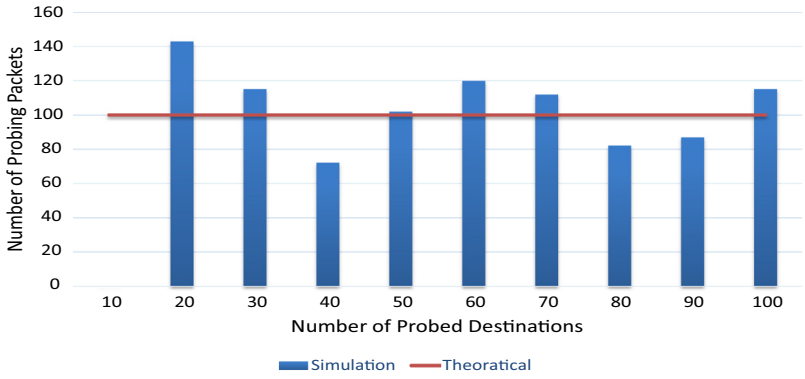
$$P_{mal}(d_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (3)$$

where σ is the standard deviation, μ is the mean, σ^2 is the variance and x is the location of the darknet destination following the distribution. Recall that Eqs. (1) and (3) allows the model to initially capture how unusual the accessed destinations are. However further, the model considers how many darknet destinations have been accessed by a given source. The latter will be subsequently described.

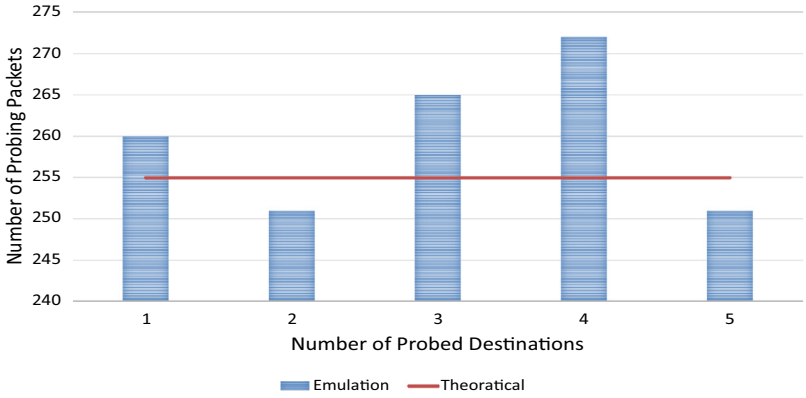
Given a set of D_i, darknet destinations accessed by a specific source s_i, the model eventually aims at measuring two probability distributions, namely, $P_{misc}(D_i)$ and $P_{mal}(D_i)$. The former being the probability that D_i has been generated by a misconfigured source while the latter is the probability that D_i has been generated by a malicious darknet source.

Let $D_1 = \{d_{i1}, d_{i2}, d_{i3}\}$ be those darknet addresses accessed by s_1. The model captures the probability $P(D_1)$ of the source generating $\{d_{i1}, d_{i2}, d_{i3}\}$ as the probability of s_1

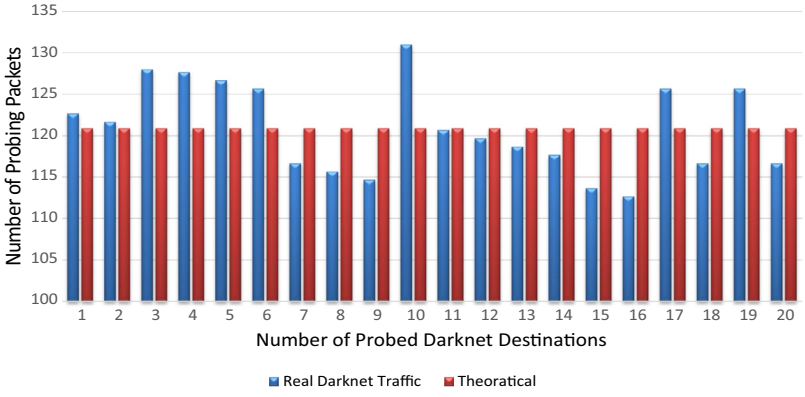
[3] http://tinyurl.com/nclm3gp.



(a) Simulation Experiment



(b) Emulation Experiment



(c) Real Traffic Experiment

Fig. 2. Uniform distribution of malicious packets in all experiments.

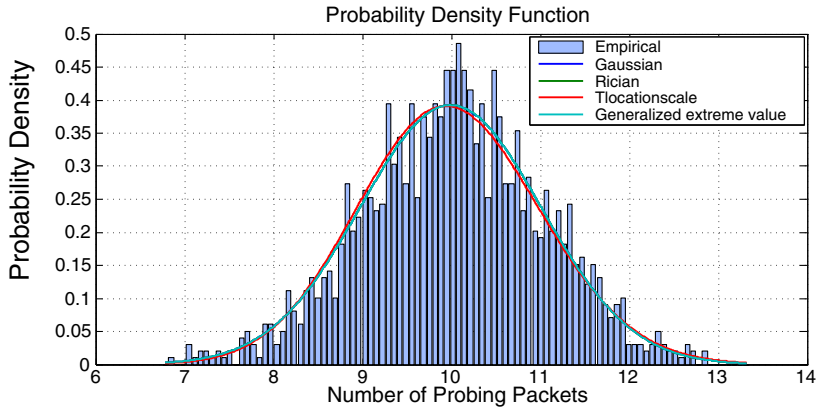
accessing this specific combination of destinations knowing that it targeted three destinations multiplied by the probability of s_1 accessing any three destinations. The latter could be generalized and formalized as

$$P(D_i) = P(D_i = \{d_{i1}, d_{i2}, \ldots, d_{in}\} \quad | \quad |D_i| = n) \times P(|D_i| = n) \tag{4}$$

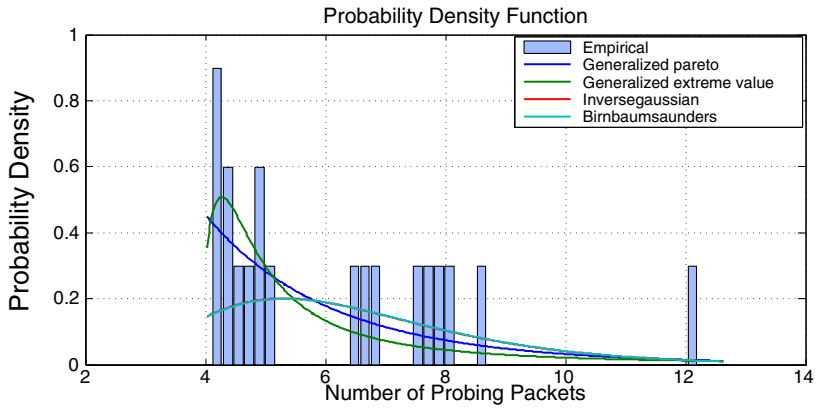
For both, a misconfigured and a malicious source, the first term of Eq. (4) could be modeled as

$$P_{misc}(D_i = \{d_{i1}, d_{i2}, \ldots\} \quad | \quad |D_i|) = \frac{1}{K} \prod_{\forall d_j \in D_i} P_{misc}(d_i) \tag{5}$$

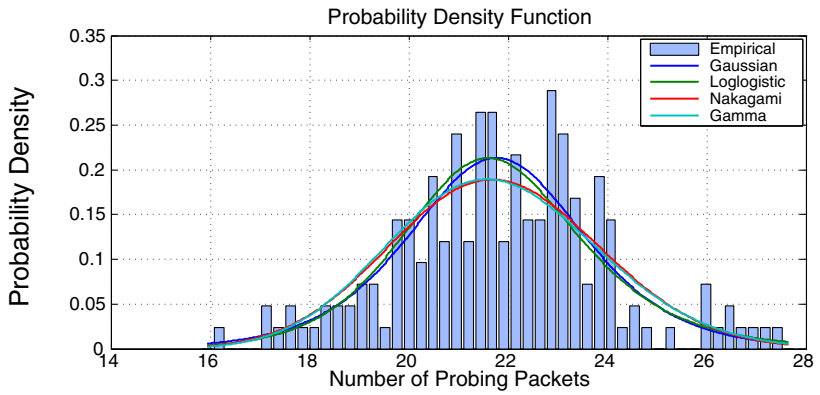
$$P_{mal}(D_i = \{d_{i1}, d_{i2}, \ldots\} \quad | \quad |D_i|) = \frac{1}{K} \prod_{\forall d_j \in D_i} P_{mal}(d_i) \tag{6}$$



(a) Simulation Experiment



(b) Emulation Experiment



(c) Real Traffic Experiment

Fig. 3. Model fitting sorted by BIC in all experiments.

where K, a normalization constant which is solely employed to allow the probabilities to sum to 1, could be defined as

$$K = \frac{|D|!}{n!(|D|-n)!} \times \frac{1}{|D|^n} \tag{7}$$

Please note that K is a typical normalization constant that is often employed in Bayesian probability[4]. Further, n represents all the sources in the data set, while $|D|$ represents the dark IP space.

[4] http://www.cs.ubc.ca/~murphyk/Bayes/bayesrule.html.

The likelihood that a source will target a certain number of darknet destinations (i.e., the second term of Eq. (4)) depends on whether the source is malicious or misconfigured. Characteristically, misconfigured sources access one or few destinations while malicious sources access a larger pool of destinations. We have modeled such distributions as

$$P_{misc}(|D_i|) = \frac{1}{(e-1)|D_i|!} \tag{8}$$

$$P_{mal}(|D_i|) = \frac{1}{|D|} \tag{9}$$

where the term $(e-1)$ in Eq. (8) guarantees that the distribution will sum to 1. It is noteworthy to mention that Eq. (8) ensures that the probability will significantly decrease as the number of targeted destinations increases. In contrast, Eq. (9) captures a malicious darknet source accessing a random number of darknet addresses.

By combining the above equations, we can model the probability of a source being a misconfigured or malicious, given a set of darknet destination addresses,

$$P_{misc}(D_i) = \frac{1}{K(e-1)|D_i|!} \prod_{\forall d_j \in D_i} P_{misc}(d_i) \tag{10}$$

$$P_{mal}(D_i) = \frac{1}{K|D|} \prod_{\forall d_j \in D_i} P_{mal}(d_i) \tag{11}$$

Indeed, Algorithm 1 provides a simplistic mechanism to infer misconfigured sources by employing the proposed darknet preprocessing model. It is worthy to note that step 6 of the algorithm (i.e., the computation of $P_{misc}(D_i)$ and $P_{mal}(D_i)$) is easily accomplished in practice by computing the negative log-likelihoods,

$$\begin{aligned} L_{misc}(D_i) &= -lnP_{misc}(D_i) \\ L_{mal}(D_i) &= -lnP_{mal}(D_i) \end{aligned} \tag{12}$$

Thus, Algorithm 1 deems a source and its corresponding flows as misconfiguration if $L_{mal}(D_i) - L_{misc}(D_i) > 0$.

Please recall that the attributes that can alter the performance are the number of misconfiguration sessions and the number of targeted ports within those sessions. For both, the performance of Algorithm 1 will be positively or negatively affected in linear O(n) time.

Algorithm 1: Inferring misconfiguration flows using the probabilistic model.

Data: Darknet Flows, *DarkFlows*
Result: Flag, *MiscFlag*, indicating that the *DarkFlow* is originating from a misconfigured source

1 **for** *DarkFlows* **do**
2 *MiscFlag* = 0
3 i = *DarkFlows*.getUniqueSources()
4 Amalgamate $DarkFlows_i$ originating from a specific source s_i
5 Update $s_i(D_i)$
6 Compute $P_{misc}(D_i)$, $P_{mal}(D_i)$
7 **if** $P_{misc}(D_i) > P_{mal}(D_i)$ **then**
8 *MiscFlag* = 1
9 **end**
10 **end**

3.2. Probing extraction

In accordance with Fig. 1, another component of the proposed approach is probing extraction. In [24], we have proposed a new approach to fingerprint probing activities. The approach aimed at detecting the probing activity and identifying the exact technique that was employed in the activity. The approach is advantageous in comparison with other methods [33–37] as it 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 depend (for detection) on a certain predefined alert threshold, the transport protocol used (TCP or UDP) or the number of probed destinations and ports. When empirically evaluated using a significant amount of real darknet data, the approach yielded 0 false negative and 2% false positive [24] in comparison with the leading Network Intrusion Detection System (NIDS), Snort[5]. In this work, and to successfully extract probing activities from darknet traffic, we adopt and leverage an enhanced version of that previously proposed approach. In what follows, we (1) briefly elaborate on the approach in [24] for self-containment purposes, (2) perform another empirical evaluational of [24] in comparison with the Bro NIDS[6] for validation purposes and (3) describe the enhancements on [24] that are employed in this work.

(1) The rationale of the previously proposed probing fingerprinting approach [24] states that regardless of the source, strategy and aim of the probing, the scanning activity should have been generated using a certain literature-known scanning technique (i.e., TCP SYN, UDP, ACK, etc. [23]). We have observed and validated 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, the approach can differentiate between probing and other malicious traffic (i.e., Denial of Service (DoS)) based on the possessed traffic correlation status. It 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, the approach statistically estimates 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 uniquely employs the time series Detrended Fluctuation Analysis (DFA) technique [38]. Fig. 4 provides a holistic view of the previously proposed approach. Readers that are interested in more details related to the approach, including its inner workings, developed

[5] http://www.snort.org/.
[6] http://www.bro.org/.

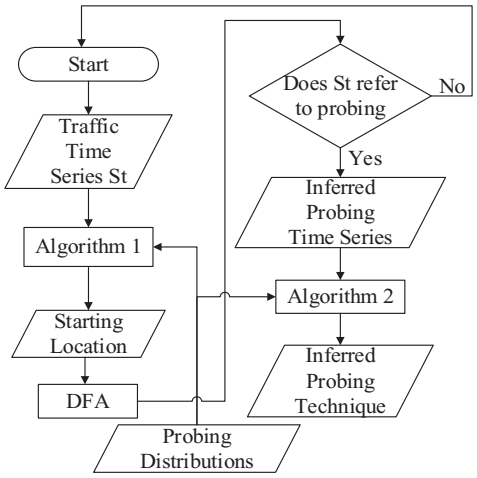


Fig. 4. Employed system process in [24].

algorithms, and leveraged statistical and heuristical techniques, are kindly referred to [24].

(2) Although the previously proposed approach was empirically evaluated and validated against Snort NIDS [24], we thought it would be also interesting in this section to further evaluate it against another NIDS, namely, Bro [39]. This procedure aims at providing more confidence in the approach in addition to significantly motivating its use as one of the initial components in Fig. 1.

To achieve this procedure, we experimented with Bro NIDS. Paxson's Bro [39] identifies sources as scanners if they execute *N* failed connection attempts to a configurable list of services. A connection is considered to have failed if it is unanswered or if it generates a TCP reset in response. For other services, Bro records information for all connection attempts. If the number of connection attempts for these services exceed *N*, this source is also identified as a scanner. The list of services and *N* are configured in various policy scripts implemented by Bro. For the sake of this work, we deployed Bro 2.3 with default configuration and employed the provided scanning script[7]. We use one week of darknet data (52 GB) that was collected during the duration of March 1st to March 7th 2014, to empirically evaluate the approach in [24] against Bro. Fig. 5 demonstrates the outcome of this experiment. By investigating the sources of the scanners, it was disclosed that the approach in [24], similar to the outcome when previously compared with Snort NIDS, yielded 0 false negative; all the scanners detected by Bro were also detected by the fingerprinting approach. Further, it could be noted that the approach generated an average of 8% false positive. It is worthy to pinpoint that the approach, contrary to Bro NIDS, can detect various types of scans, which could 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. Nevertheless, in an attempt to reduce the number of false positives, we enhance the approach as explained in the following.

[7] https://github.com/bro/bro-scripts/blob/master/scan.bro.

Table 1
UDP vulnerable services and corresponding ports.

UDP service	Port number
DNS	53
NTP	123
SNMPV2	161
NetBIOS	137
SSDP	1900
CharGEN	19
QOTD	17
Quake network protocol	26000

(3) We perform a crucial enhancement to [24] that we employ in this work. The modification is rendered by UDP investigation. One of the issues in [24] is that it does not differentiate between UDP probing and UDP packets arising from distributed reflective denial of service attacks [40]; this can explain the relatively high percentage of false positives. Indeed, such attacks are an emerging form of distributed denial of service attacks that rely on the use of publicly accessible UDP servers as well as bandwidth amplification factors to overwhelm a victim with UDP traffic [41]. The idea is to send simple queries to such resolvers in which the replies, that aim at flooding the victim, are orders of magnitude larger. Such approach is behind the notorious 300 and 400 Gbps attacks that hit the Internet in the last few years [42]. Fig. 6 depicts a specific scenario where the resolvers are open Domain Name System (DNS) servers. The compromised machines are directed to execute a reflective dos attack against Org1. To achieve that, they initially spoof their identities by using those of Org1 and subsequently send simple `ANY` queries to the DNS open resolvers. The latter DNS query intends to pull all the available information from the DNS resolvers related to a requested domain. The domain in the request trace is often a noteworthy one that possess a significant amount of information. Commonly, the compromised machines will spay such queries on the Internet space in a hope to reach as many open resolvers as possible in order to increase the overall amplification factor. Intuitively, some of those requests will hit the network telescope/dark space and hence will be captured. Requests that actually reach open resolvers will be amplified and directed towards Org1.

Typically, the UDP services that could be leveraged for UDP amplification attacks are summarized in Table 1. To deal with this issue, we extend [24] by employing the post-processing Algorithm 2.

The algorithm aims at validating whether the inferred UDP probing flow or session from the outcome of [24] is indeed a probing activity or it is actually a false positive related to UDP packets originating from distributed reflective denial of service attacks. The algorithm achieves this by initially assuming that the flow is indeed a probing flow. Subsequently, it attempts to negate the latter assumption by relying on two observations. First, it monitors if the UDP flow in question is targeting one of those services of Table 1; a positive result, at this stage, suggests that the UDP flow is either probing that service or it is attempting to amplify a denial of service attack. If a positive result is recorded from the latter, the algorithm then verifies whether the source is spoofed or not.

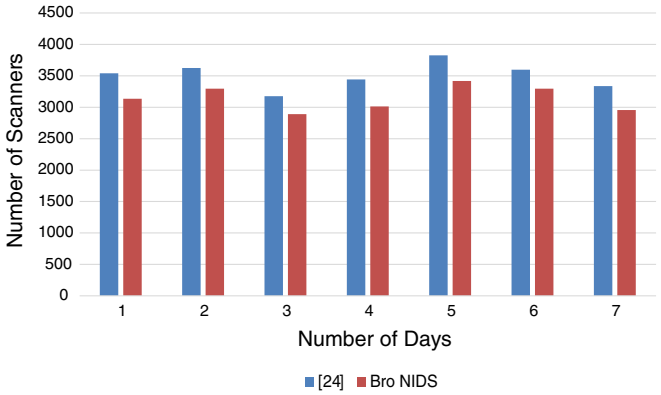


Fig. 5. Empirical evaluation of [24] against Bro NIDS.

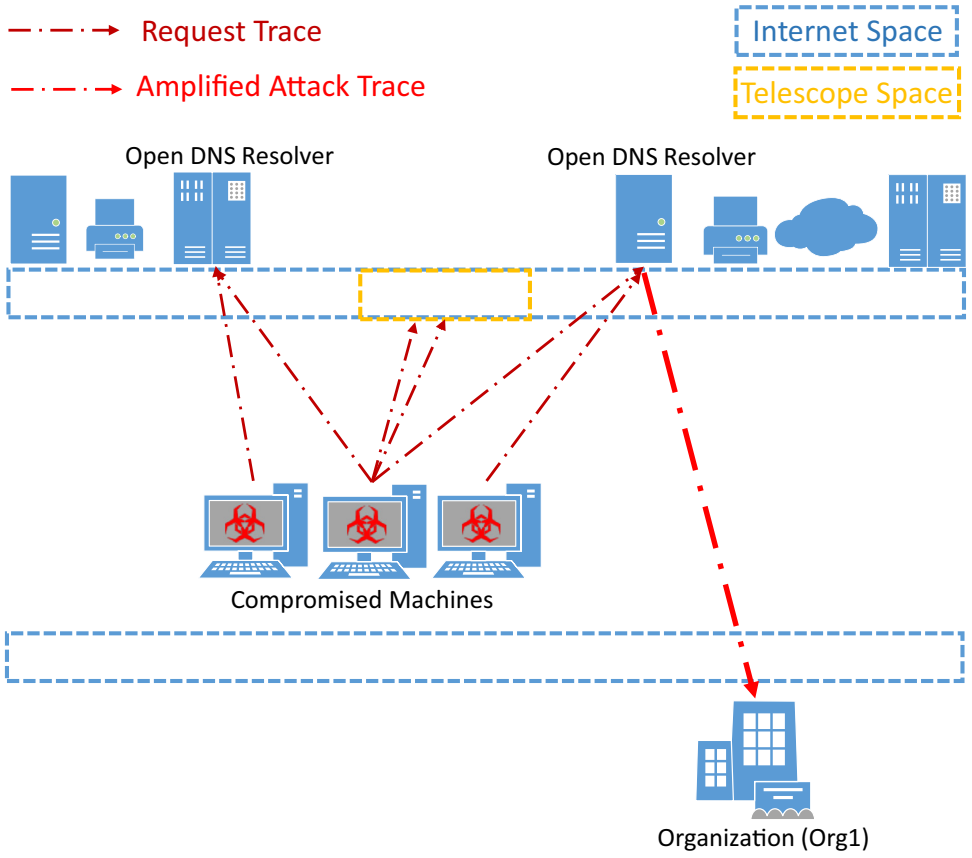


Fig. 6. A network telescope pinpointing sources of reflective DoS attacks.

The rational behind that verification is based on the fact that probing packets are not spoofed (so the scanner/attacker can actually retrieve back the result of the scan) while those of a denial of service attack are indeed spoofed for identity anonymization purposes. The algorithm accomplishes this by leveraging the work of Templeton and Levitt [43] that maintains a track of the Time-To-Live (TTL) value of the packets generated by the same source. The core idea behind that approach is rendered by the fact that packets' TTL values fluctuate with source IP addresses; that is, given a non-spoofed IP address, its packets' TTL should remain constant (or within 10% change) throughout its communication period. It is worthy to mention that in terms of complexity, the algorithm possesses O(PF) space complexity and O(N) time complexity where PF is the number of probing flows and N is the number of UDP flows. By implementing the proposed Algorithm 2, we recorded a decrease of 3.5% in false positives when we re-executed the empirical evaluation (against Bro NIDS) of Section 3.2. Such result demonstrates that (1) the algorithm is effective in distinguishing between UDP probing

Algorithm 2: Verifying if the UDP flow is a probing activity or related to UDP distributed reflective denial of service attack.

```
Data: Probing Flows, ProbingFlows;
      List of Vulnerable UDP Services, UDPVulList
Result: Flag, ProbFlag, indicating whether the UDP
        flow, UDPFlow, is a true probing or not
for UDPFlow in ProbingFlows do
    ProbFlag = 0;
    if UDPFlow.getTarget().getPort() in UDPVulList then
        TTL = UDPFlow.getTTL();
        for UDPFlow in ProbingFlows do
            if UDPFlow.getTTL()! = TTL then
                ProbFlag = 1;
            end
        end
    end
end
```

and UDP packets originating from amplified denial of service attacks and (2) that the majority of the false positives of Section 3.2 were indeed caused by this issue.

Please recall that the outcome of this procedure (i.e., probing extraction in Fig. 1) are accurate and validated probing sessions in packet capture format (i.e., pcaps).

3.3. Malware invocation

In accordance with Fig. 1, another component of the proposed approach is malware invocation. We operate a dynamic malware analysis module that is based on ThreatTrack Security sandbox environment[8] (i.e., controlled environment). After receiving daily malware samples from ThreatTrack feeds, they are interactively sent to the sandbox, where they are executed by client machines. The clients could be virtual or real and possess the capability to run Windows or Unix, depending on the malware type under execution. The behavior of each malware is monitored and all its corresponding activities (i.e., created files, processes, network traffic, etc.) are recorded. For the sake of this work, we extract the network traffic generated by approximately 65 thousand unique and recent malware samples as pcaps. The pcaps contain communication traffic generated from the malware to other internal or external hosts. Those malware samples belong to diverse malware types including, Trojan, Virus, Worm, Backdoor, and AdWare coupled with their corresponding families and variants. We rely on Kaspersky for a uniform malware naming convention[9].

3.4. Correlation execution

Consistent with Fig. 1, the correlation engine formulates the problem of correlating probing and malware sessions as follows. Given a probing session that is extracted from darknet traffic, (1) investigate whether or not the session originates from a malware and (2) identify the exact or probable malware type/family that is generating such probing, if it was shown that the session is malware-related. In an attempt to address this problem, we perform the following. We leverage Snort's probing engine, the sfPortscan pre-processor, to detect which malware pcaps possess any signs of probing activity. We omit those malware pcaps that demonstrate a negative output. To attribute a specific malware to a probing session, we adopt a two-step procedure. First, we apply the notion of fuzzy hashing [44] between the probing session and the remaining malware pcaps. Fuzzy hashing is advantageous in comparison with typical hashing as it can provide a percentage of similarity between two samples rather than producing a null value if the samples are different. This popular technique is derived from the digital forensics research field and is typically applied on files or images [44,45]; to the best of our knowledge, our approach is among the first to explore the capabilities of this technique on cyber security data. Readers that are more interested in the advantages of fuzzy hashing and its applicability to malware research are kindly referred to [46,47]. We further apply an information theoretical metric, relative entropy, as proposed by Lee and Xiang [48], between the given probing session and the malware pcaps. Relative entropy, which is defined by

$$d = \sum_k p_k \log \frac{p_k}{q_k}$$

is a measure of the distance of the regularities between two datasets, p_k and q_k. If the relative entropy is $= 0$, this indicates that the two datasets have the same regularity. At this point, we (1) omit the probing sessions that demonstrate less than 5% similarity using both tests[10] and (2) select the top 10% malware pcaps that were found to minimize the entropy and maximize the fuzzy hashing percentage. The rational behind the latter approach stems from the need to filter out the malware pcaps that do not possess probing signs similar to the probing session. Second, using the remaining 10% malware pcaps, we extract their probing sessions as pinpointed by sfPortscan. For each of the malware probing sessions, we apply the Bhattacharyya distance [49] between those and the given probing session. The latter statistic test, which is defined by

$$d(p, p') = \sqrt{1 - L(p, p')}$$

where

$$L(p, p') = \sum_{i=1}^{N} \sqrt{p(i)p'(i)},$$

is an established and an effective metric to determine the overlap of two sample distributions, p and p'. The Bhattacharyya distance is often employed to measure the disjunction of classes in a typical classification problem and it is considered to be more reliable than other metrics, including for instance, the Mahalanobis distance [50]; when the two classes have similar means but different standard deviations,

[8] http://www.threattracksecurity.com/.

[9] http://securelist.com/en/threats/detect?chapter=136.

[10] These sessions indicate that they do not possess any malware-related behavior.

the Mahalanobis distance would tend to zero, while the Bhattacharyya distance would grow and yield better results depending on the difference between the standard deviations. By selecting 1% of malware pcaps that were shown to reduce the Bhattacharyya distance, we further significantly reduce the possible malware pcaps that the given probing session could be similar to. Finally, to exactly attribute the given probing session to a specific malware, we employ the two sample Kolmogorov–Smirnov statistic test [51] between the remaining malware probing sessions and the given probing session. The test will output 0 if a positive match occur; 1 otherwise. If a positive match occurs, this indicates that the probing session has been generated from the inferred exact malware. Otherwise, we refer back to the output of the Bhattacharyya distance and select a set of probable malware pcaps that were shown to be relatively close to the given probing session.

It is worthy to mention that from a processing overhead perspective, the parameters that can indeed affect the performance are the number of darknet malicious sessions as well as the number of malware samples. In the worst case, the approach will be executed in $O(n^2)$ time. Practically and on average, the proposed approach is able to process and correlate one day of darknet data (≈8 GB) with 65 thousands malware samples in less than 30 min. The latter information strongly advocate that the approach is practically viable in a real-world environment and that it can be easily rendered as an operational cyber security capability providing prompt near real-time inferences related to worldwide-compromised machines. Additionally, we would expect an even better performance if the approach is implemented into a single, coherent module using a low-level coding language such as *C*. The latter task is currently work-in-progress.

4. Empirical evaluation

We possess real darknet data that we are receiving on a daily basis from a trusted third party, namely, Farsight Security (https://archive.farsightsecurity.com/SIE_Channel_14/). 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. Recall that darknet traffic is Internet traffic destined to routable but unused Internet addresses. Since these addresses are unallocated, any traffic targeting them is deemed as suspicious. The data mostly consists of unsolicited TCP, UDP and ICMP traffic. It might contain as well some DNS traffic. We use one week of data (60 GB) that was collected during the duration of April 1st to April 7th 2014, to empirically evaluate our approach.

Darknet traffic is typically composed of three types of traffic, namely, scanning, backscattered and misconfiguration [52]. 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, as tackled in Section 3.1, is due to network/routing or hardware/software faults causing such traffic to be sent to the darknet sensors.

We first aggregate the darknet traffic connections into sessions using an approach similar to the first step algorithm by Kannan et al. [53]. 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 proposed threshold as in [53], $T_{aggreg} = 100$ s, and found that this seems to correctly group the majority of connections between any given pair of hosts. To filter out misconfiguration data, we employ the proposed probabilistic model of Section 3.1 coupled with Algorithm 1. Fig. 7 depicts the distribution of darknet sessions between misconfiguration and malicious sessions while Table 2 presents a sample of 10 misconfigured sources as inferred by the algorithm.

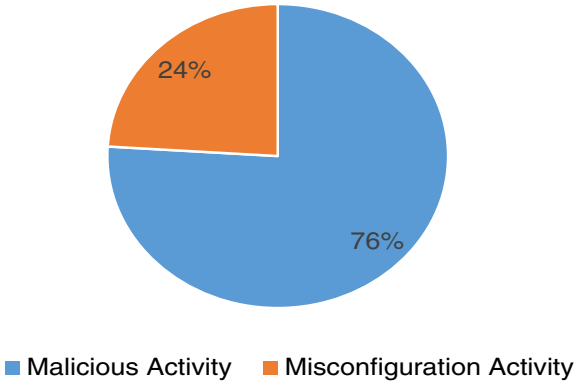


Fig. 7. The distribution of darknet sessions.

Table 2
A sample of 10 misconfigured sources.

Source	$L_{mal} - L_{misc}$
1	99
2	132
3	113
4	14
5	146
6	106
7	39
8	97
9	2
10	133

It is revealed that close to 25% of the sessions are indeed related to misconfiguration. This is relatively consistent with a previous study that we have performed in 2012 [54], which was solely based on manual verification. By further manually investigating the inferred misconfiguration sessions, it was shown that all the sessions are single flows that targeted the dark space only once in which 87% of them are malformed packets.

We further execute the enhanced probing fingerprinting approach that was highlighted in Section 3.2 to extract 400 probing sessions. It is noteworthy to mention, that each of those probing sessions is being generated by a unique source. Keeping in mind that most probing activity is generated from non-spoofed IP addresses (so the actual scanner can essentially receive back the probing results), the extracted probing sessions indeed resemble real machines.

We proceed by investigating any signs of probing activities within the 65 thousand malware pcaps. The output

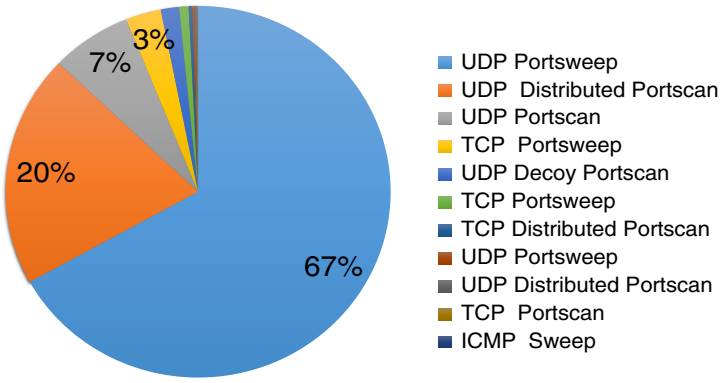


Fig. 8. Types of probing activities generated by malware.

disclosed that 13,105 malware pcaps possessed such activities. The latter corroborates that a significant amount of malware samples in fact generate probing activities; leveraging such activities for early infection detection might be a viable and a promising approach. The distribution of the types of those probing activities within the identified malware pcaps is depicted in Fig. 8. It is disclosed that UDP probing is the most employed probing technique; such result is consistent with other malware studies [55], where the authors revealed that UDP is the most used transport-layer protocol for malicious command-and-control communications. Further, Fig. 9 illustrates the top 10 malware types that were found to trigger such probing activities. One interesting observation that could be extracted from such result is related to 'Virus.Win32.Sality.bh'; Dainotti et al. [1] have recently documented a large-scale probing campaign that was able to probe VoIP (SIP) servers of the entire IPv4 address space in 12 days. The authors pinpointed that the malware responsible for such campaign was in fact the Sality malware; the same malware that we found, using our data set, to be generating the majority of the probing activities.

We proceed in an attempt to reduce the number of malware pcaps that could be eventually attributed to the darknet probing sessions. In accordance with Section 3.4, we executed the fuzzy hashing and relative entropy approach. To accomplish the former, we leveraged deeptoad[11], a fuzzy hashing implementation, while we employed matlab[12] to accomplish the latter. Subsequently, we select the top 10% (1,310) malware pcaps that were found to minimize the entropy and maximize the fuzzy hashing percentage, in comparison with the darknet probing sessions. At this point, 212 probing sessions were filtered out, indicating that they do not possess any malware-related behavior.

For the purpose of attributing the remaining probing sessions to a manageable and a probable set of malware pcaps, in coherence with the proposed approach of Section 3.4, we proceed by executing the Bhattacharyya distance and selecting the 1% of malware pcaps (13 pcaps) that are shown to be statistically close to each of the probing sessions. Table 3 provides a specimen that couples 5 probing sources with few

Table 3
Probing sources coupled with their probable malware samples.

Probing source 1	Trojan-Downloader.Win32.KiayksayRen.b
	Trojan-FakeAV.Win32.SmartFortress2012.il
	Trojan.Win32.VBKrypt.hadj
	Trojan-Dropper.Win32.Agent.dtki
Probing source 2	Trojan-Downloader.Win32.KiayksayRen.b
	Trojan-Dropper.Win32.Dapato.aflm
	Trojan-Dropper.Win32.Injector.dknf
	Trojan.Win32.Jorik.Shakblades.foc
Probing source 3	Trojan-Downloader.Win32.KiayksayRen.b
	Worm.Win32.AutoIt.xl
	Trojan.Win32.Scar.furz
	Packed.Win32.PolyCrypt.d
Probing source 4	DTrojan-Downloader.Win32.KiayksayRen.b
	Trojan-Downloader.Win32.Dapato.gje
	Trojan.Win32.Agent.btmu
	Virus.Win32.Sality.bh
	Trojan-FakeAV.Win32.SmartFortress2012.v
Probing source 5	Trojan-Downloader.Win32.KiayksayRen.b
	Trojan-Spy.Win32.SpyEyes.acxb
	Trojan.Win32.FakeAV.lete
	Packed.Win32.PolyCrypt.d

of their corresponding possible set of malware infections. Intuitively, we record the complete malware outcome for all the probing sources.

Note that 'Trojan-Downloader.Win32.KiayksayRen.b', that frequently appears in Table 3, has been confirmed by numerous anti-malware engines and services to be a significant sign of machine exploitation[13], particularly those running the Windows operating system. Thus, the proposed approach seems accurate and practical in pinpointing compromised machines in addition to disclosing the probable malware types that caused their contamination.

To exactly identify which malware sample is responsible for the darknet extracted probings sessions, we proceed by employing the Kolmogorov–Smirnov statistic test, as instructed in Section 3.4. Table 4 shows the extracted insights for a sample of 10 probing sessions while Fig. 10 visualizes the worldwide map of the fingerprinted infections.

Note that we also generate supplementary material related to the infections including geo-location information per real source (i.e., hostname), organization, ISP, city, region and

[11] https://code.google.com/p/deeptoad/.

[12] http://www.mathworks.com/matlabcentral/fileexchange/35625-information-theory-toolbox/content/relativeEntropy.m.

[13] http://tinyurl.com/ktlqp4r.

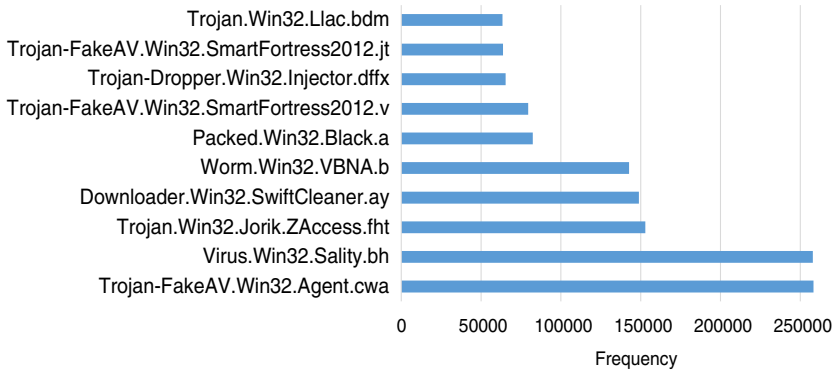


Fig. 9. Distribution of probing malware types/families.

Fig. 10. Inferred worldwide infections by correlating malware and probing activities.

Table 4
A sample of inferred infections.

Probing source a	Trojan-Spy.Win32.VB.gt
Probing source b	Virus.Win32.Sality.s
Probing source c	Trojan-FakeAV.Win32.SmartFortress2012.jv
Probing source d	Trojan-Dropper.Win32.Injector.dpdj
Probing source e	Backdoor.Win32.Bifrose.fur
Probing source f	Virus.Win32.Cabanas.MsgBox
Probing source g	Trojan.Win32.Jorik.Downloader.akr
Probing source h	Trojan-FakeAV.Win32.Agent.cwa
Probing source i	Trojan.Win32.Swisyn.bahq
Probing source j	Worm.Win32.Juched.djh

country. Although, we refrain from publishing those due to sensitivity/legal issues, we can note that the infections originate from 67 diverse operational providers, 67 distinct ISPs and 38 different countries. Such results, which we postulate to be communicated to concerned providers, concur that the proposed approach possesses the capability to infer compromised machines in addition to pinpointing the exact malware type/family that was responsible for their contamination.

Although we are unable to validate the existence of every single obtained inference related to the extracted worldwide infections due to legal and logistic constraints, we perform two activities that advocate the accuracy and completeness of the proposed approach.

First, we have observed, from the obtained results, a number of events that support the proposed approach. (1) We inferred that the majority of the infections that are related to the previously mentioned 'Virus-Win32.Sality.bh' are originating from Thailand; in [1], the authors disclosed that the bots that contributed to the large-scale VoIP probing campaign that were found to be infected by the same Sality malware, were in fact attributed to Thailand. (2) We noticed that Chinese ISPs lead in the number of generated infections. According to our results, one of the top extracted malware infections that is generated from those ISPs is the 'Trojan-Banker.Win32.Banker.adx'. This malware is a data stealing program that captures banking credentials such as account numbers and passwords from infected users. The latter insights were confirmed by McAfee in which they further concurred that China is in fact responsible from more than 45% of such contamination[14]. (3) From our results, we deduced that the malware 'Backdoor:Win32/Bifrose' was originating from a specific middle-Eastern country. The latter malware allows an external attacker to access the compromised machine to perform various malicious actions. McAfee also

[14] http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=853515.

confirmed our findings by revealing that the same country is indeed the most contributor to such an infection[15].

Second, we relied on third party publically available data sources provided by the online services, ThreatStop[16], MxLookup[17], brightcloud[18] and ReputationAuthority[19] to validate the obtained contamination incidents. The latter cyber security data repositories provide information on Internet-scale contamination incidents per IP address. We compare the extracted malware IP addresses that were inferred by the proposed approach against those repositories. The outcome discloses that around 91% of the extracted malware IP addresses from the proposed approach were indeed found as malware-related from those repositories. The outcome also pinpointed that 57% of those overlapping confirmed incidents are still active on operational machines.

To further demonstrate that the proposed approach is general and prompt in inferring Internet-scale infections, we have performed the following auxiliary experiment. We have applied the proposed approach on a very recent darknet dataset of 8 GB obtained on September 1st, 2015. In particular, we have executed the correlation mechanism of Section 3.4 between this new dataset and around 60 GB of malware samples. The rationale of choosing and experimenting with one very recent dataset in contrast to selecting different (yet older) datasets at different points in time, is that the recent dataset allows us to validate the outcome of the approach against infections that are actually still active, in the wild, on operational machines. By executing this experiment, the outcome disclosed 133 infected machines hosted at 7 unique ISPs. Cross-validating the latter infections with the above mentioned publicly available threat repositories, showed that 93.9% (125 IP addresses) of the inferred infections are also mentioned in those threat repositories, where 82.4% of them (103 IP addresses) are still active infections in the wild. Such results demonstrate that the extracted inferences from the proposed approach exhibit noteworthy accuracy and can generate significant cyber security insights that could be used for prompt mitigation.

5. Approach limitations

It is realistic to acknowledge a number of limitations in the proposed approach. First, the approach leverages the dark space to infer Internet-scale probing activities. Although the monitored space is relatively large (i.e., /13), we are unable to monitor events that do not target such space. Subsequently, the approach will be unable to correlate those "unseen" activities with malware samples, and thus will fail to detect and identify their corresponding malware infections. Please note that it is feasible to expand the dark IP space by leveraging approaches similar to those in [15]. The latter would also aid entities that aim at adopting our proposed approach, in which they do not have access to a darknet, and thus there exists a need to discover and monitor darknet IP addresses. Second, another limitation is that the approach relies on malware samples that actually execute probing activities. Although, from our experiments, the number of those malware seems to be significant, the approach will not be able to detect malware that do not probe. In this case, our correlation engine could be used in conjunction with already deployed approaches, similar to those that rely on honeypots or active detection of malicious machines [19,20] to accomplish the detection. Third, in relation to the misconfiguration darknet preprocessing model, there is a need to design and implement confidence levels to assess whether the difference of the two probability estimates is large enough to safely choose one model over the other. This point is left for future work. Fourth, the approach is still experimental; its development is ongoing for the purpose of making it operational in an automated and a real-time fashion.

6. Conclusion

This paper presented a new approach to infer malware-infected machines. The approach aims at providing network operators with a cyber security capability to detect their clients' compromised machines in addition to pinpointing the exact malware type that caused their contamination. The approach is efficient as it does not record or analyze the symptoms of infection. Further, it is prompt as it exploits probing activities, which are the very first indications of contamination. Moreover, the proposed approach is cost-effective as it does not require any implementation or maintenance costs at the providers' sides. To accomplish its goals, the proposed approach exploits the dark space to infer and validate Internet-scale probing activities after filtering out misconfiguration traffic using a newly proposed probabilistic model. Consequently, it correlates probing activities with malware samples by uniquely employing various statistical, fuzzy hashing and information theoretical metrics. The approach was empirically evaluated using a significant amount of real darknet and malware samples. The extracted inferences and insights revealed promising accuracy in addition to concurring that the rationale of exploiting probing activities for worldwide early malware infection detection is indeed practically viable. As for future work, we strive to leverage this work coupled with clustering mechanisms based on probing behavioral analysis in an attempt to infer malware-orchestrated campaigns.

[15] http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=1594628.

[16] http://www.threatstop.com/.

[17] http://mxtoolbox.com/.

[18] http://www.brightcloud.com/.

[19] http://www.reputationauthority.org/.

References

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

[2] J. Mirkovic, P. Reiher, A taxonomy of ddos attack and ddos defense mechanisms, ACM SIGCOMM Comput. Commun. Rev. 34 (2) (2004) 39–53.

[3] M. Daly, Advanced persistent threat , Usenix 4 (Nov 2009).

[4] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, I. Osipkov, Spamming botnets: signatures and characteristics, in: ACM SIGCOMM Computer Communication Review, 38, ACM, 2008, pp. 171–182.

[5] Panda Security, Worldwide infected machines, http://tinyurl.com/o24ky8t. (accessed November 2015).

[6] ScMagazine-McAfee, The state of malware in 2013, http://tinyurl.com/ohjprsc. (accessed November 2015).

[7] Parliament of Canada, BILL C-28, http://tinyurl.com/avh9vzv. (accessed November 2015).

[8] S.H. Usman, A review of responsibilities of internet service providers toward their customers' network security, J. Theor. Appl. Inf. Technol. 49 (1) (2013).

[9] ZDNet, ISPs accused of ignoring botnet invasion, http://tinyurl.com/lt48jzl. (accessed November 2015).

[10] E. Bou-Harb, C. Fachkha, M. Debbabi, C. Assi, Inferring internet-scale infections by correlating malware and probing activities, in: Communications (ICC), 2014 IEEE International Conference on, IEEE, 2014, pp. 640–646.

[11] K. NAKAO, D. INOUE, M. ETO, K. YOSHIOKA, Practical correlation analysis between scan and malware profiles against zero-day attacks based on darknet monitoring, IEICE Trans. Inf. Syst. 92 (5) (2009) 787–798, doi:10.1587/transinf.E92.D.787.

[12] D. Inoue, M. Eto, K. Yoshioka, S. Baba, K. Suzuki, J. Nakazato, K. Ohtaka, K. Nakao, nicter: An incident analysis system toward binding network monitoring with malware analysis, in: Information Security Threats Data Collection and Sharing, 2008. WISTDCS '08., 2008, pp. 58–66, doi:10.1109/WISTDCS.2008.14.

[13] J. Song, J. Shimamura, M. Eto, D. Inoue, K. Nakao, Correlation analysis between spamming botnets and malware infected hosts, in: 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet, 0, 2011, pp. 372–375, doi:10.1109/SAINT.2011.71.

[14] M. Eto, K. Sonoda, D. Inoue, K. Yoshioka, K. Nakao, A proposal of malware distinction method based on scan patterns using spectrum analysis, in: C. Leung, M. Lee, J. Chan (Eds.), Neural Information Processing, Lecture Notes in Computer Science, 5864, Springer Berlin Heidelberg, 2009, pp. 565–572, doi:10.1007/978-3-642-10684-2_63.

[15] E. Cooke, M. Bailey, F. Jahanian, R. Mortier, The dark oracle: Perspective-aware unused and unreachable address discovery., in: NSDI, 6, 2006. 8–8.

[16] G. Gu, P.A. Porras, V. Yegneswaran, M.W. Fong, W. Lee, Bothunter: Detecting malware infection through ids-driven dialog correlation., in: USENIX Security, 7, 2007, pp. 1–16.

[17] D. Whyte, E. Kranakis, P.C. van Oorschot, Dns-based detection of scanning worms in an enterprise network., in: NDSS, 2005.

[18] S.E. Schechter, J. Jung, A.W. Berger, Fast detection of scanning worm infections, in: Recent Advances in Intrusion Detection, Springer, 2004, pp. 59–81.

[19] Z. Xu, A. Nappa, R. Baykov, G. Yang, J. Caballero, G. Gu, Autoprobe: towards automatic active malicious server probing using dynamic binary analysis, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 179–190.

[20] A. Nappa, Z. Xu, M.Z. Rafique, J. Caballero, G. Gu, Cyberprobe: Towards internet-scale active detection of malicious servers, in: Network and Distributed System Security Symposium, 2014.

[21] M.Z. Rafique, J. Caballero, Firma: Malware clustering and network signature generation with mixed network behaviors, in: Research in Attacks, Intrusions, and Defenses, Springer, 2013, pp. 144–163.

[22] R. Perdisci, W. Lee, N. Feamster, Behavioral clustering of http-based malware and signature generation using malicious network traces., in: NSDI, 2010, pp. 391–404.

[23] E. Bou-Harb, M. Debbabi, C. Assi, Cyber scanning: a comprehensive survey, IEEE Commun. Surv. Tutor. PP (99) (2013) 1–24, doi:10.1109/SURV.2013.102913.00020.

[24] E. Bou-Harb, M. Debbabi, C. Assi, A statistical approach for fingerprinting probing activities, in: 2013 Eighth International Conference on Availability, Reliability and Security (ARES), 2013, pp. 21–30, doi:10.1109/ARES.2013.9.

[25] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson, et al., 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. 167–179.

[26] C. Fachkha, E. Bou-Harb, A. Boukhtouta, S. Dinh, F. Iqbal, M. Debbabi, Investigating the dark cyberspace: Profiling, threat-based analysis and correlation, in: 2012 7th International Conference on Risk and Security of Internet and Systems (CRiSIS), 2012, pp. 1–8, doi:10.1109/CRISIS.2012.6378947.

[27] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, S. Sinha, Practical darknet measurement, in: Information Sciences and Systems, 2006 40th Annual Conference on, IEEE, 2006, pp. 1496–1501.

[28] M. Ford, J. Stevens, J. Ronan, Initial results from an ipv6 darknet13, in: Internet Surveillance and Protection, 2006. ICISP'06. International Conference on, IEEE, 2006. 13–13.

[29] R. Berthier, M. Cukier, The deployment of a darknet on an organization-wide network: an empirical analysis, in: High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE, 2008, pp. 59–68, doi:10.1109/HASE.2008.54.

[30] Z. Durumeric, M. Bailey, J.A. Halderman, An internet-wide view of internet-wide scanning, in: USENIX Security Symposium, 2014.

[31] K. Dowd, A.J. Cairns, D. Blake, G.D. Coughlan, D. Epstein, M. Khalaf-Allah, Evaluating the goodness of fit of stochastic mortality models, Insur. Math. Econ. 47 (3) (2010) 255–265.

[32] Y. Sakamoto, M. Ishiguro, G. Kitagawa, Akaike Information Criterion Statistics, Dordrecht, The Netherlands: D. Reidel, 1986.

[33] J. Treurniet, A network activity classification schema and its application to scan detection, IEEE/ACM Trans. Netw. 19 (5) (2011) 1396–1404.

[34] S. Staniford, J. Hoagland, J. McAlerney, Practical automated detection of stealthy portscans, J. Comput. Secur. 10 (1/2) (2002) 105–136.

[35] W. Zhang, S. Teng, X. Fu, Scan attack detection based on distributed cooperative model, in: Proceedings of the International Conference on Computer Supported Cooperative Work in Design, CSCWD 2008, IEEE, 2008, pp. 743–748.

[36] R. Baldoni, G.D. Luna, L. Querzoni, Collaborative detection of coordinated port scans, Distributed computing and networking, Springer, 2013, pp. 102–107. (accessed November 2015).

[37] G. Conti, K. Abdullah, Passive visual fingerprinting of network attack tools, in: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, ACM, 2004, pp. 45–54.

[38] C.-K. Peng, S.V. Buldyrev, S. Havlin, M. Simons, H.E. Stanley, A.L. Goldberger, Mosaic organization of dna nucleotides, Phys. Rev. E 49 (2) (1994) 1685.

[39] V. Paxson, Bro: a system for detecting network intruders in real-time, Comput. Netw. 31 (23) (1999) 2435–2463.

[40] C. Rossow, Amplification hell: revisiting network protocols for DDoS abuse, in: Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium, 2014.

[41] C. Fachkha, E. Bou-Harb, M. Debbabi, Fingerprinting internet DNS amplification DDOS activities, in: New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on, IEEE, 2014, pp. 1–5.

[42] Largest ever ddos attack peaks at 400 gbps, Info Securityl, http://tinyurl.com/nljf3ct. (accessed November 2015).

[43] S.J. Templeton, K.E. Levitt, Detecting spoofed packets, in: DARPA Information Survivability Conference and Exposition, 2003. Proceedings, 1, IEEE, 2003, pp. 164–175.

[44] J. Kornblum, Identifying almost identical files using context triggered piecewise hashing, Digit. Investig. 3, Supplement (0) (2006) 91–97, doi:10.1016/j.diin.2006.06.015. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[45] Y.-P. Huang, T.-W. Chang, F.-E. Sandnes, An efficient fuzzy hashing model for image retrieval, in: Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American, 2006, pp. 223–228, doi:10.1109/NAFIPS.2006.365412.

[46] J. Kornblum, Fuzzy hashing, ManTech SMA, 2012. http://jessekornblum.com/presentations/htcia06.pdf

[47] K. Dunham, A fuzzy future in malware research, The ISSA J. 11 (8) (2013) 17–18.

[48] W. Lee, D. Xiang, Information-theoretic measures for anomaly detection, in: Proceedings of the 2001 IEEE Symposium on Security and Privacy, 2001. S P 2001., 2001, pp. 130–143, doi:10.1109/SECPRI.2001.924294.

[49] T. Kailath, The divergence and bhattacharyya distance measures in signal selection, IEEE Trans. Commun. Technol. 15 (1) (1967) 52–60, doi:10.1109/TCOM.1967.1089532.

[50] R. De Maesschalck, D. Jouan-Rimbaud, D.L. Massart, The mahalanobis distance, Chemom. Intell. Lab. Syst. 50 (1) (2000) 1–18.

[51] H.W. Lilliefors, On the Kolmogorov–Smirnov test for normality with mean and variance unknown, J. Am. Stat. Assoc. 62 (318) (1967) 399–402.

[52] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, G. Huston, Internet background radiation revisited, in: Proceedings of the 10th annual conference on Internet measurement, ACM, 2010, pp. 62–74.

[53] J. Kannan, J. Jung, V. Paxson, C.E. Koksal, Semi-automated discovery of application session structure, in: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, ACM, 2006, pp. 119–132.

[54] C. Fachkha, E. Bou-Harb, A. Boukhtouta, S. Dinh, F. Iqbal, M. Debbabi, Investigating the dark cyberspace: Profiling, threat-based analysis and correlation, in: Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on, IEEE, 2012, pp. 1–8.

[55] T. Karagiannis, A. Broido, M. Faloutsos, K. claffy, Transport layer identification of p2p traffic, in: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, in: IMC '04, ACM, New York, NY, USA, 2004, pp. 121–134, doi:10.1145/1028788.1028804.

Elias Bou-Harb is currently a visiting research scientist at Carnegie Mellon University (CMU) under the sponsorship of Professor Bruno Sinopoli. He is also a permanent research scientist at the National Cyber Forensic and Training Alliance (NCFTA) of Canada. The latter is an international organization which focuses on the investigation of cyber-crimes impacting citizens and businesses. Elias holds a Ph.D. degree in computer science from Concordia University in Montreal, Canada, which was executed under the supervision of Professors Mourad Debbabi and Chadi Assi. In January 2016, Elias will be joining the department of computer science at Florida Atlantic University (FAU) as an assistant professor. His research and development activities and interests focus on the broad area of operational cyber security, including, attacks detection and characterization, Internet measurement, cyber security for critical infrastructure and mobile network security. Elias has been supported by the prestigious Alexander Graham Bell Canada Graduate Scholarship (CGS) from the Natural Sciences and Engineering ResearchCouncil (NSERC) of Canada.

Dr. Mourad Debbabi is a Full Professor at the Concordia Institute for Information Systems Engineering. He holds the Concordia Research Chair Tier I in Information Systems Security. He is also the President of the National Cyber Forensics Training Alliance (NCFTA Canada). He is the founder and one of the leaders of the Computer Security Laboratory (CSL) at Concordia University. In the past, he was the Specification Lead of four Standard JAIN (Java Intelligent Networks) Java Specification Requests (JSRs) dedicated to the elaboration of standard specifications for presence and instant messaging. Dr. Debbabi holds Ph.D. and M.Sc. degrees in computer science from Paris-XI Orsay, University, France. He published 2 books and more than 230 research papers in journals and conferences on computer security, cyber forensics, privacy, cryptographic protocols, threat intelligence generation, malware analysis, reverse engineering, specification and verification of safety-critical systems, formal methods, Java security and acceleration, programming languages and type theory. He supervised to successful completion 20 Ph.D. students and more than 60 Master students. He served as a Senior Scientist at the Panasonic Information and Network Technologies Laboratory, Princeton, New Jersey, USA; Associate Professor at the Computer Science Department of Laval University, Quebec, Canada; Senior Scientist at General Electric Research Center, New York, USA; Research Associate at the Computer Science Department of Stanford University, California, USA; and Permanent Researcher at the Bull Corporate Research Center, Paris, France.

Chadi Assi received his B.Eng. degree from the Lebanese University, Beirut, Lebanon, in 1997 and his Ph.D. degree from the City University of New York (CUNY) in April 2003. He is currently a full professor with the Concordia Institute for Information Systems Engineering, Concordia University. Before joining Concordia University in August 2003 as an assistant professor, he was a visiting researcher with Nokia Research Center, Boston, Massachusetts, where he worked on quality of service in passive optical access networks. His research interests are in the areas of networks and network design and optimization. He received the prestigious Mina Rees Dissertation Award from CUNY in August 2002 for his research on wavelength-division multiplexing optical networks. He is on the Editorial Board of IEEE Communications Surveys & Tutorials, IEEE Transactions on Communications, and IEEE Transactions on Vehicular Technologies.