On the Analysis of Backscatter Traffic

Eray Balkanli
Faculty of Computer Science
Dalhousie University
Halifax, Canada
eray.balkanli@dal.ca

A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Canada
zincir@cs.dal.ca

***Abstract*—This work offers in-depth analysis of three different darknet datasets captured in 2004, 2006 and 2008 to provide insights into the nature of backscatter traffic. Moreover, we analyzed these datasets using two well-known open source intrusion detection systems (IDSs), namely Snort and Bro. Our analysis shows that there are interesting trends in these datasets that help us to understand backscatter traffic over a 4-year period of time. However, it also shows that it is challenging to identify the attacks that generated this traffic.**

***Keywords*—Backscatter, DDoS, Darknet, Network measurements, Network Security**

I. Introduction

Denial of Service (DoS) and Distributed Denial of Service (DDoS) are well known network attacks that aim to prevent users to access a system. For example, most recently, Reuters reported that undefined sources generated DoS attacks to the communication channels of the National Security and Defence Council of Ukraine, in March 2014 [1]. Also, One of the popular online role-playing games, Wurm, was taken offline because of a DDoS attack in February 2014 [2]. Given these malicious activities, network measurement and monitoring are important tools to ensure the understanding of attack trends. To this end, network telescopes (darknets) are used for data collection and traffic measurements by many security analysts.

A darknet is a network that consists of IP addresses and ports where there is no network device set up to send/receive data. Darknets allow one to observe different large-scale events taking place on the Internet. Their goal is to observe the traffic targeting the unused (dark) address-space of the network. Since all traffic to these addresses is suspicious, one can gain information about possible network attacks as well as misconfigurations by observing it. The resolution of the darknet depends on the number of dark addresses it monitors. For example, a large one that monitors traffic to 16,777,216 addresses (a /8 network), has a higher probability of observing a relatively small event than a smaller darknet (telescope) that monitors 65,536 addresses (a /16 network). In general, there are two categories of darknets depending on their configurations: Passive and Active. A passive darknet records all the packets received by at least one of the unallocated IP addresses in the range of the darknet. On the other hand, active darknets do not only record these packets but also respond to them in order to collect more information about the attacker and/or the attack process.

In this research, we analyze publicly available darknet traffic data from CAIDA traffic archives [9]. Even though CAIDA does not provide much information about these datasets, it states that they are mostly backscatter traffic. Backscatter traffic is a side effect of spoofed DoS/DDoS attacks. In this kind of DoS attack, the attacker spoofs the source address of the network packets sent to the victim. In general, the victim responds to these spoofed packets as if they were normal. The traffic generated by these responses is called the backscatter traffic. In this work, we employed the backscatter traffic data from years 2004, 2006 and 2008 to be able to perform traffic measurements and analysis. We also assume that these datasets represent the traffic from passive darknets. according to CAIDAs explanations in [9]. It should be noted here that these were the only and latest available backscatter datasets including solely one-way traffic when we started this research. these were the only and latest available backscatter datasets including solely one-way traffic when we started this research. Given these properties about the datasets, our aim is to shed light into the following issues:

1) What is the nature of darknet traffic including mostly backscatter attacks?
2) Are there any well known encrypted ports in this traffic? If so, do they have special roles?
3) Are there any well-known P2P (Peer-to-Peer) applications in this traffic? If so, do they have special roles?
4) What are the geo-locations of the source IP addresses seen in these backscatter traffic?
5) Can the current intrusion detection systems identify the different attack behaviours in this traffic? If so, what do they identify?

To achieve our aim, we begin with analyzing the main characteristics of the darknet traffic by measuring packet distributions, protocol distributions and protocol types in order to understand its nature. To this end, we analyze the encrypted ports such as SSL and P2P applications such as BitTorrent to understand how much they are used in these backscatter (darknet) datasets. In addition, we analyze the geo-location of the source IP addresses to find the countries producing most of the backscatter traffic. Finally, we employ two different open source IDSs, Snort [22] and Bro [23], and also a one-way traffic analyzer, Iatmon [25], to study the malicious behaviours these tools would identify when they are run on a real darknet dataset.

The rest of the paper is organized as follows. Section II discusses the related work in this field. Section III discusses the datasets, techniques and tools employed in our analysis. Section IV presents our experimental results. Finally, Section V draws conclusions and discusses the future work.

978-1-4799-3784-4/14/$31.00 ©2014 IEEE

II. Related Work

There are many studies related to the installation of a darknet platform and analysis of the collected darknet data. Bailey et al. focused darknet configuration techniques, such as activeness or passiveness, and analyzed a large darknet dataset collected via Internet Motion Sensor (IMS) by clustering the data based on unique source IP addresses [3]. Eto et al. introduced a network analysis center for tactical emergency response that aims to monitor various types of darknets, such as /8, /16 and /24 of network addresses. They analyze network attacks by measuring the correlation between the network threats seen in the darknet and malwares captured by various honeypots [4]. Moore et al. analyzed the worldwide DDoS activity by using 22 different backscatter datasets captured between 2001 and 2004 [6]. Wang et al. focused on describing the nature of the Internet worms and employed statistical techniques to measure host infection times and reconstruct worm infection sequences [5]. They experimentally confirmed their analytical results on a worm dataset provided by CAIDA via /8 network telescopes. Pang et al. studied the characterization of both the passive and the active darknet data, collected in 2004 from /8 and /19 iSink [7] darknets as well as /24 Lawrence Berkeley National Laboratory (LBL) darknets. They employed a number of statistical measurements including the nature of data, most used application-level responders and filtering process by source-connection, source-payload, source-port and source-destination [13]. Wustrow et al. reengineered Pang et al.s research [13] without using any filtering to highlight how the network traffic activity has been evolved in the recent years [33]. They explored the characteristics of their datasets by measuring the protocol/port selections, traffic types, sizes etc., and revealed the critical environmental changes of the background radiation such as misconfiguration and location over the unused /8 network blocks. Last but not the least, Fachkha et al. used datasets collected from many /16 address blocks to analyze the nature of the darknet packets, mostly using network, transport and application layer protocols [12]. They also described 30 types of threats observed in their dataset, and studies the relationships between the threats using association rule mining.

Aforementioned studies mostly discuss the importance of configuring and monitoring unused address blocks and analyzing the packets captured in the monitored IP addresses to identify network threats and their characteristics. Commonly, these studies include both analytical and statistical analysis to describe the nature of the datasets used and study any trends indicating the observed threats. Filtering by transport layer protocols, source and destination IP addresses, port numbers and specific time periods are the most favoured techniques implemented during these studies. In general, the datasets used consisted of darknet and network telescope traffic.

Our research is complementary to these studies in terms of providing measurements of darknet data and DDoS attack behaviors. The main difference between the existing researches and our research is that we focus on three different datasets consisting of mostly backscatter data over a 4-year period. This work mainly contributes in the following two aspects: i) We shed light into the backscatter (DDoS) behaviours and how they change (if at all) over time. ii) We analyze the performances of Snort v2.9.1 as well as v2.9.6.1, Bro v2.2 and Iatmon v2.1.2 to observe how successful they are in detecting backscatter traffic.

III. Methodology

In this work, three different publicly available backscatter datasets are used from CAIDA archieves. These traffic files were captured by UCSD Network Telescope [9] also known as a passive darknet at San Diego in 2004, 2006 and 2008 [8]. These darknet backscatter datasets only involve one-way DDoS attack traffic, incoming packets to the darknet. Since UCSD network telescope is a passive darknet, it does not respond to the incoming traffic, so there is no outgoing traffic included in our datasets. Additionally, the destination IP addresses are hidden by assigning their first octets to 0. The earliest day of traffic from each dataset, namely May 28 from 2004, Feb 23 from 2006 and Feb 22 from 2008, are selected for our analysis. In doing so, we aim to study how the behaviour of such attacks vary (if at all) over the two-year intervals. Furthermore, since the knowledge database provided by these NIDSs achieve their highest performance on two-way traffic, we also employed all the November 2008 dataset by using only Iatmon to observe how a special one-way traffic analyzer would measure on this traffic. Table I presents the sizes of the datasets employed in this research.

TABLE I: Sizes of the datasets Employed

	Sizes	
Dataset	**Number of Packets**	**Size (GB)**
May 28, 2004	57,641,141	4.4
Feb 23, 2006	85,547,065	7.5
Feb 22, 2008	81,606,489	6.9
November, 2008	1,317,888,867	102.7

After obtaining these samples, we first analyzed the percentages of scanning, backscatter and misconfiguration traffic in our datasets. Then, we performed measurements on the network, the transport and the application layers of the data obtained to understand the most used ports and protocols, in other words applications. We also analyzed the usage of well known secure ports, as well as P2P application ports to study how much these ports and applications were used in generating backscatter traffic. Moreover, we studied which countries were generating these backscatter traffic. Finally, we evaluated the performance of the network intrusion detection systems, namely Snort v2.9.1 as well as v2.9.6.0, Bro IDS v2.2 and Iatmon v2.1.2 on these darknet datasets to understand how much of the open source IDSs able to detect such attacks.

To the best of our knowledge, this is the first work analyzing aforementioned backscatter datasets by using these techniques. In only order to analyze and measure the trends existing in these datasets, we employ open source tools such as Wireshark [19] and Tshark [20]. Wireshark is an open source network protocol analyzer with a graphical user interface (GUI) that is used for capturing, filtering and analyzing live or previously captured traffic (data) formatted in tcpdump or libpcap. Tshark is also an open source network protocol analyzer without a GUI, but it supports using scripts. In addition, GeoLite Country [21] database is used for detecting geo-locations of the attack sources. Using publicly available

datasets and open source tools, from protocol analysis to intrusion detection for our measurements, ensures that our research can be easily validated and compared against others in the field.

IV. Evaluation

In this section, we discuss the analysis of our dataset measurements.

A. Measurements on Different Types of Traffic

In general, darknet datasets are categorized into three broad classes: Scanning, Backscatter and Misconfiguration. Scanning data represents the traffic generated by the DDoS attacks to discover the vulnerable targets. Backscatter data represents the background noise traffic resulting from these DDoS attacks using many spoofed IP addresses. Misconfiguration data represents the traffic generated by any software or hardware errors as well as user faults. So we also categorized the traffic in our datasets into these three classes based on their TCP flags using a similar technique as in [33] and [6]. In these datasets, SYN packets are categorized as scanning traffic. RST, ACK, SYN + ACK, and RST + ACK packets are categorized as backscatter traffic. Then, the remaining packets are categorized as misconfiguration traffic. Table II shows the percentages of the packets per category for each selected dataset. The main purpose of Table II is to show the different categories of traffic of the datasets employed. Indeed, most of the packets belong to the backscatter category as expected. 28th May 2004 dataset has the most backscatter attack packets and 23rd February 2006 dataset has the most scanning (TCP SYN) and misconfiguration packets.

TABLE II: Types of Traffic Observed in Each Dataset

Dataset	Network Traffic Category		
	Scanning	**Backscatter**	**Misconfiguration**
May 28, 2004	0.01%	90.54%	9.45%
Feb 23, 2006	0.2%	78.1%	21.7%
Feb 22, 2008	0.1%	88%	11.9%

To study the nature of the backscatter traffic in more detail, we categorized the backscatter traffic based on the TCP flags of the packets. Table III shows the overall measurement of the TCP flags used. It is clear that even though SYN+ACK packets were less in 2004, they increased for the years 2006 and 2008, as opposed to the decrease of the RST and RST|ACK packets. As it is well known, the SYN, SYN+ACK and ACK packets are used in TCP connections for three-way handshaking. Furthermore, the RST packets are used by the hosts that do not receive any SYN+ACK packets over a long period of time for any packet they send. Therefore, the total number of RST packets decreases when the total number of SYN+ACK packets increases. In these datasets, the increase in the number of SYN+ACK packets and the remaining the low percentage in the number of ACK packets is an important indicator for the increase of the total number of incomplete three-way handshake connections. This indicates an increase in the number of DDoS attacks in 2008 and 2006 compared to 2004. It should be noted here that, there were DDoS attacks directed to Wikileaks [29] and to some on-line gaming web sites [30] in February 2008. There were also DDoS attacks targeting many prominent blogs [31] and payment gateways [32] in February 2006. On the other hand, in May 2004 the number of known DDoS attacks are relatively lower compared to 2006 and 2008 datasets, Table III.

TABLE III: Backscatter TCP Traffic Distributions

Dataset	TCP Flag Type			
	SYN+ACK	**RST**	**RST+ACK**	**ACK**
May 28, 2004	21.48%	35.8%	42.7%	0.02%
Feb 23, 2006	82.5%	2.4%	15.1%	0%
Feb 22, 2008	77.1%	6.1%	16.8%	0%

B. Measurements on Different Protocols

To identify the importance of the transport and network layer protocols commonly used in the datasets analyzed, the percentages of TCP, UDP and ICMP traffic are measured and shown in Table IV. As seen in Table IV, the major protocol seen in these datasets is TCP. The main reason why TCP is the most likely to be observed is the usage of three-way handshaking with a spoofed IP address is one of the most common ways to perform DDoS attacks, i.e. backscatter traffic. Moreover, there are many attacks both using and targeting TCP ports as presented in [13]. As the TCP traffic decreases (Table IV), also the backscatter traffic decreases (Table II), however the misconfiguration traffic increases (Table II).

TABLE IV: Protocol Measurements

Dataset	Protocol				
	TCP	***ICMP***			***Other***
		Type-11		*Other*	
		TCP	*UDP*	*Only ICMP*	
May 28, 2004	98.4%	1.1%	0.3%	0.12%	0.08%
Feb 23, 2006	88.22%	2.2%	15%	0.63%	0.05%
Feb 22, 2008	87.9%	6%	6%	0%	0.01%

It should be noted here that there is no UDP packets in any of these datasets. However, there are some type-11 ICMP packets. This represents the packets that have zero as their TTL values. In this case, the first 64 bits of these packets are kept, so their transport layer protocol can still be identified as UDP. Table IV shows the percentages of these type-11 ICMP packets. It is seen that the number of type-11 ICMP packets were the lowest in 2004 and the highest in 2006. Even though 2008 dataset is a close second, coming right after 2006, the nature of the type-11 ICMP packets in 2006 are much different than the ones in 2008. In 2008, 12% of the whole dataset is type-11 ICMP packets. Moreover, half of these packets have UDP as their transport layer protocol whereas the other half has TCP. However, in 2006, approximately 10 million (15% of the whole dataset and the 88% of the type-11 ICMP packets) of these packets have UDP as their transport layer protocol. Note that as the number of these UDP packets increases, the percentage of the backscatter traffic decreases. According to Table IV, while the percentage of the UDP packets in 2008 is less than 2006, it is more than 2004. Depending on Table II, backscatter traffic in 2006 is the least, then comes 2008 and after which comes 2004. Finally, the total number of the packets that belong to other protocols, i.e. "others", is less than one million for each dataset analyzed in this work.

C. Application Layer (AL) Protocol Measurements

To identify the AL protocols that are used the most in these datasets, we analyzed the port numbers of the packets, since we do not have access to the payload in these traffic datasets. Given that more than 50% of the packets have port 80 as their destination port, this seems to suggest that they were sent over the HTTP protocol. There is no other major application protocol to emphasize for the 2004 dataset. However, the application layer protocols that are used the most in the 2006 and 2008 datasets show different trends. Figure 1 presents the measurements of the application ports in these datasets.

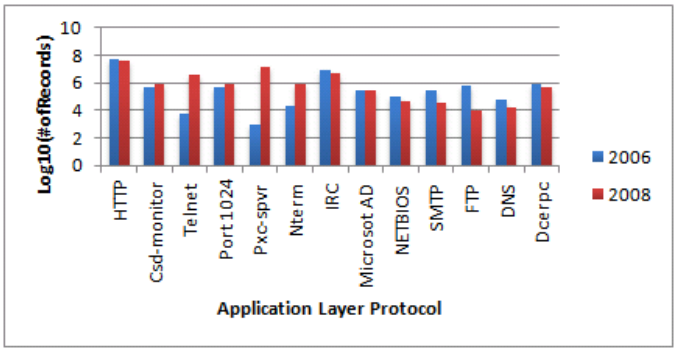

Fig. 1: Application Layer Protocol Usage for the datasets 2006 and 2008.

According to our measurements, the HTTP protocol is again the most used AL protocol in 2006 and 2008 (more than 50% of the packets in total). However, besides the HTTP protocol, the following AL protocols are also used in these datasets: pxc-spvr, nterm, telnet, csd-monitor, IRC, dcerpc, FTP, Microsoft AD, SMTP, NETBIOS and DNS.

In this context, Nterm, which refers $'remote\_login\ network\_terminal'$, is a terminal-based application allowing the easy use of different applications, directories, URLs and documents via port 1026 [11]. Csd-monitor, which uses port 3072, has been reported as a channel used by DDoS attacks [10]. Therefore, this port generally is blocked by firewalls or security programs to avoid such attacks. Dcerpc is a client/server protocol used for running a software application on a remote server over the port 135. Note that in August 2003, it was discovered that the W32.Blaster worm was using this port [14]. This worm temporarily blocks the RPC (Remote Procedure Call) service by using the ports 135, 69 and 4444. This explains why we have observed this port a lot in these datasets. Server Message Block (SMB) is used as an AL protocol to handle shared accesses and data exchanges on multiple threads. SMB uses port 445 to run on Microsoft AD, a special database to manage large amounts of select operations. Furthermore, ports 137/138/139 are used to run NETBIOS, which allows executing applications on different computers over a LAN [14]. The other AL protocols observed in these datasets in the top 10 AL list are well-known protocols in the field. For example, Telnet is a command-based protocol for remote connections via port 23. IRC is a text protocol for chatting on ports 194 or 6667. FTP is a protocol used for file transferring on ports 20 and 21. SMTP is a protocol used for sending e-mails between IP networks on port 25,

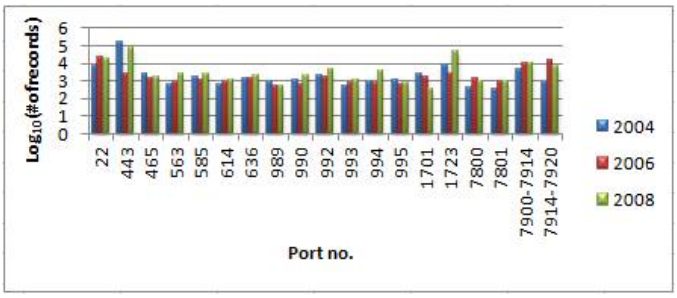

Fig. 2: Encrypted Traffic Measurements

and finally, DNS is a protocol that converts IP addresses into domain names (and vice versa) running on port 53.

D. Encrypted Traffic Measurements

Since we do not have the payload, we cannot be sure how much of the traffic is encrypted in these datasets. However, we measure the usage of the ports, such as port 443 that usually carry encrypted traffic.To this end, we analyzed the datasets to identify the packets which use one of the following 19 ports known to carry potential encrypted traffic [24][26][27]. Hereafter, we refer to these ports as secure ports.

Figure 2 shows the overall usage of these secure ports for each dataset. According to these measurements, the total percentage of potentially encrypted traffic (packets on a secure port / all packets * 100) is less than 0.5% for each dataset. However, within this small portion, SSH and SSL are the most popular secure ports. SSL was mostly used in 2004 and SSH in 2006. This analysis implies that for most of the DDoS (backscatter) attacks performed, the ports (applications) employed are not known to be encrypted. Note that it is also likely to observe encrypted traffic using other ports, e.g. Skype on port 80. However, it is challenging to detect such traffic [17].

E. Peer-to-peer (P2P) Traffic Measurements

P2P traffic has considerably grown over the last several years. In a P2P network, none of the hosts needs to be controlled by a centralized server. File sharing applications are the most popular usage area of the P2P systems. To investigate the P2P activities for our research, 10 popular P2P application protocols are analyzed. These are: Edonkey[15], Gnutella (bearshare and limewire)[15], KaZaA[15], DirectConnect[15], BitTorrent[15], WinMx[16], Ares[16], Soulseek[16] and Waste[18]. Such P2P applications might use different port numbers. However, we have measured the usage of these AL protocols based on their default port numbers.

Figure 3 provides the overall measurements for the P2P application usage. According to these measurements, the total percentage of P2P traffic is less than 0.05%. However within this traffic, SoulSeek, BitTorrent and Edonkey are the most popular P2P applications used by the DDoS attacks in 2004, 2006 and 2008, respectively. The usage of KaZaA, Waste and Soulseek dramatically decreases year by year. It is interesting to note that KaZaA was a popular file sharing application

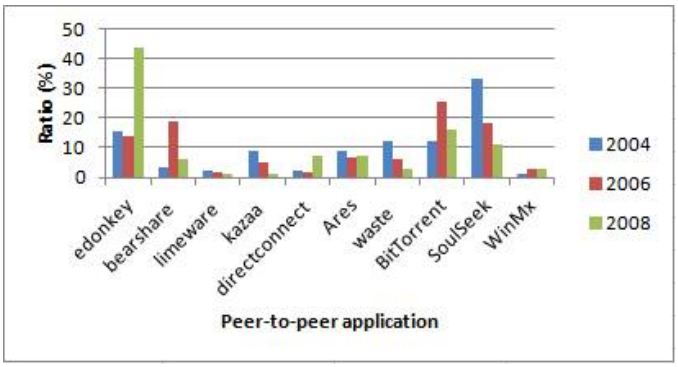

Fig. 3: Peer-to-peer Traffic Measurements

when it is first released. However, after 2006, it lost most of its users since various clients were infected because of the viruses KaZaA includes. Shin et al. [28] demonstrated that more than 15% of the KaZaA data was including 40 different viruses between February and May 2006 and 12% of the KaZaA users were infected. It is also mentioned that approximately 7.2% of the victims could notify they were infected. This is also reflected in our measurements, where the highest usage of KaZaA is in 2004 dataset, the usage decreases in 2006 and goes down to almost 0% in 2008. Moreover, the first version of Bearshare was released in 2005, but it lost its popularity quickly. Therefore, we observe a relatively high usage of Bearshare in 2006, but not in 2004 or 2008.

In the used dataset, the direction of the one-way traffic is to the hosts whose IP addresses were spoofed by an attacker from the backscatter victims. Therefore, we know the IP addresses of the victims, but not know the spoofed IP addresses for the sake of their privacy.

F. Geo-location Analysis

Geo-location technologies are important for showing the movement of darknet sources and generating stronger filters to analyze illegal network traffic. To find the geographic positions of the sources generating these backscatter traffic, the geo-location of all the victim's IP addresses who generates backscatter traffic to the spoofed IPs have been identified.

Table V shows the source countries of such traffic with the ratio of how much malicious traffic they produce in the datasets analyzed in this work. For each dataset, China plays the major role. Also, although Pakistan generates more than 10% of such traffic in 2004, its share dramatically decreases in 2006 and 2008. Actually, the only countries where there is increase in the percentage of generated DDoS traffic are USA and Taiwan. However, note that in 2004 and in 2006, these attacks were more distributed in terms of their sources (over a larger number of countries) since the "other" countries contribute to almost 13% and 14% of the datasets, respectively. However, this number drops to 2% in 2008. This seem to suggest that the attack sources seem to be less distributed in 2008.

TABLE V: Geo-location Measurements

Country	Dataset		
	May 28, 2004	**Feb 23, 2006**	**Feb 22, 2008**
China	67.9%	49.4%	53.1%
Germany	0.3%	8.3%	2%
Korea	0.4%	2.1%	1.1%
Pakistan	11.2%	0.01%	0.03%
Taiwan	0.6%	0.5%	18.3%
USA	6.4%	25.8%	23.4%
Other	13.1%	13.9%	2.1%

G. Measurements Performed Using IDSs

The main purpose of an IDS is to assist a network / security analyst when the traffic analyzed is suspected to be an attack. A signature-based IDS defines a packet as an attack if it matches with a signature that is available in the database of the IDS. We employed Snort [22] and Bro [23], two well known IDSs. We also employ Iatmon on November-2008 backscatter dataset and evaluate its performance by considering only the packets classified under backscatter group. In this case, our goal is to measure and analyze the performance of these systems on the darknet datasets employed in this work. This would give us a better understanding of how much of the attack behaviours seen in these real life datasets can actually be identified with these open source tools. For our research, we employed Snort versions v2.9.1 and v2.9.6.0. We also employed Bro IDS v2.2 with the script "scan.bro". The reason we employed these versions are two fold: (i) The earlier version of Snort (we do not have access to an earlier version of Bro) enables us to analyze the datasets employed with the rules (signatures) known in the years when the datasets were captured, i.e. v2.9.1 is from 2005. On the other hand, (ii) the latest versions of Snort and Bro enable us to analyze the datasets with the rules (signatures) that are used currently, i.e. v2.9.6.0 for Snort is from 2014, and v2.2 for Bro is from 2013. Finally, since the employed dataset includes only one-way traffic, we employed Iatmon v2.1.2, which is specifically developed to inspect one-way network traffic, to measure and compare its performance with the used NIDSs.

1) Signature (Rule) Categorizations: Bro, Snort and Iatmon tools have a number of various signatures (rules) to identify malicious behaviours in the data. For example; for Bro, we used "scan.bro" script since there is not a specific default script to detect DDoS attacks in Bro IDS. Instead of defining specific rules for filtering, this script consists of defining specific conditions (events) as listed below:

- When an unsuccessful connection attempt occurs
- When a TCP connection is rejected
- When an endpoint aborts a TCP connection
- Open connections when Bro is terminated

On the other hand, Snort IDS uses special filters on packets to check if they are suspicious or not. Even though there is not any specific attack category defined in Snort v2.9.1, nine different attack categories have been defined in Snort v2.9.6.0. According to those categories, we classified the rule sets and analyzed the rules that are triggered on the datasets employed. Table VI shows the name of the attack category, the number of sub-categories, specific rules belonging to each category and the number of the triggered rules on our datasets. "# of sub cat." column refers to the number of the different rule

sets under the related category. For instance, icmp.rules and snmp.rules are such rule sets under the protocol category. "# of rules" column describes the total number of the rules for each category. As an example, there are 529 different single rules under protocol category defined in Snort v2.9.1. Finally, "# of trig. rules" column represents the rules that are used (triggered) by Snort when a packet is identified as suspicious, i.e. malicious behaviour. Each rule that is used for at least one of our datasets is counted under this measurement.

TABLE VI: Snort Rule Categorization

Category	Snort v2.9.1			Snort v2.9.6.0		
	# of sub cat.	# of rules	# of trig. rules	# of sub cat.	# of rules	# of trig. rules
Browser	-	-	-	6	2,875	0
File	-	-	-	9	4,624	0
Indicator	8	294	4	6	2,814	0
Malware	1	82	-	4	3,835	0
Operating System	-	-	-	5	722	0
Policy	2	30	-	4	404	0
Protocol	13	529	20	15	1,136	13
PuA	1	18	-	4	876	0
Server	3	582	-	10	3,779	0
Web	7	1,070	-	-	-	-
Other	9	506	7	6	1,173	7
Total	45	3,111	31	69	21,838	20

It is clearly seen that the number of total rules is approximately seven times more in Snort v2.9.6.0 than in Snort v2.9.1. While there is no rule for the browser, file and operating system categories in Snort v2.9.1, Snort v2.9.6.0 has 8221 rules in total for those categories. Although the web category has the most number of rules in Snort v2.9.1, they have all been deleted in Snort v2.9.6.0. This is because the categorization of rules have changed in the latest version of Snort. In this case, the file category, with 4624 rules, has the most number of rules in the new version of Snort. More importantly, even though the number of total rules have increased, the number of triggered rules were decreased by 11 in Snort v2.9.6.0. It means, the version v2.9.6.0 is expected to have less success rate in detecting the malicious behaviours. However, it is important to underline that some rules might cause false alarms based on the change on the signatures of network attacks in time. This explains the reason behind the removal of such rules. Table VII shows the triggered rules, the number of times the rule was triggered, and also both the deleted and the newly added effective rules by Snort v2.9.6.0. Note that the number of "triggered rules" refers to the detected attacks; in each time a rule is triggered, it means it detects an attack packet.

According to these results, the rules in protocol-icmp rule set are the most successful while monitoring a network for the backscatter behaviour. "Destination & Port Unreachable" and "TTL Exceeded in Transit" rules detected more behaviours, specifically backscatter behaviour than the other rules. In addition, this analysis seems to suggest that the rule "Hi Client Unknown Method" is the only added rule by Snort v2.9.6.0 beneficial for analyzing such DDoS behaviours. Moreover, the rule from Snort v2.9.1 -"Destination Unreachable Communication With Destination Host is Administratively Prohibited"- which was effective to identify some of the malicious traffic, seems no longer a part of the new version of Snort.

Furthermore, Brownlee [25] presented an open-source one-way network traffic analyzer, Iatmon (Inter-Arrival Time Monitor), that partitions the traffic into pre-defined subsets depending on their source types and inter-arrival-time (IAT) [34]. Iatmon is capable of providing a durable monitoring of unused addresses as well as unsolicited traffic and ignoring traffic to assigned hosts. There are 14 types and 10 different groups have been defined as default configuration, which indicates 140 subsets. The types are defined based on the information retrieved from the IP header of each packet such as protocol and port number, while the group distribution is based on the packet volume, lifetime, rate and time interval between the successive packets. It is also noteworthy that Iatmon discards the sources that are idle for 120 seconds or send no more than two packets to detect backscatter attacks.

2) Performances of IDSs: To evaluate the performances of Bro and Snort systems on our datasets, we run two experiments: (i) on all the traffic of these datasets; and (ii) on just the backscatter traffic of these datasets. It should be noted here that the scanning data is low in these datasets, Table II. Table VIII shows the performances of these IDSs for the aforementioned experiments. Note that Snort does not analyze the packets that have a secure port (see section 4.D) as their destination. According to the results, the Snort rule set released in 2005 gives a better performance in terms of the number of malicious packets detected. The latest Snort rule set is unsuccessful to detect DDoS attacks, especially for the 2004 and 2006 datasets. These results show Snort cannot identify the backscatter traffic correctly. On the other hand, Bro can identify at most 15% of the backscatter traffic with the minimum false positive error rate.

Although the number of total rules dramatically increased in the latest version of Snort, this new version is not as successful as the older version in detecting malicious packets. However, note that if the "deleted.rules" section of the rules set were in the new version, then the success rate would be closer to the older version. Moreover, these results show that most of the packets detected by Snort belong to the misconfiguration category. On the other hand, Bro has a better success rate than Snort in detecting backscatter packets. However, the most interesting results is the fact that these well known intrusion detection systems cannot detect the malicious behaviour in these datasets from 2004 to 2008. What they detect as malicious seems to be the misconfiguration traffic most of the time, especially with Snort. In practice, these would cause a high false alarm rate and indicate that most of the defined rules are not actually symptom of attacks.

Iatmon detects 3.3% of the packets in November-2008 dataset as backscatter traffic. Moreover, 4.1% of all of the source IP addresses are detected as suspicious sender. There are two important reasons why the detection rate of Iatmon is low: i) Iatmon ignores to analyze the sources which are idle at least 120 seconds or does not send more than two packets. ii) Iatmon detects backscatter traffic by only checking if the values of ACK and RST flags of the inspected packet is 1, or the time-to-live value of the inspected packet is exceeded. However, in the employed datasets, one can see that attackers also used other methods to create these attacks.

TABLE VII: Snort Rule Specification

Snort Rules	# of Triggered Times		
	May 28, 2004	**Feb 23, 2006**	**Feb 22, 2008**
Preprocessor.rules			
Hi Server No Contlen	4	76	1
Excessive Overlap	137	698	139
Tiny Fragment	42	288	0
Short Fragmentation	576	6,763	0
Teardrop	232	71	361
Anomaly Overlap	452	956	144
Hi Client Unknown Method (Added by v2.9.6.0)	804,825	1,306	11406
Anomaly Oversize	12	11	0
Bad Reset	254	88	21
Protocol-icmp.rules			
Address Mask Request	8	0	0
Destination Unreachable Fragmentation Needed and DF bit was set	82	6,794	818
Destination Unreachable Host Unreachable	54,269	1,336,068	420,929
Destination Unreachable Network Unreachable	2,684	39,800	13,835
Destination Unreachable Port Unreachable	5,141	9,343,279	3,146,756
Destination Unreachable Protocol Unreachable	793	1,225	2,987
Echo Reply	28,649	29,386	44,677
Fragment Reassembly Time Exceeded	217	2,681	867
TTL Exceeded in Transit	257,892	2,719,675	4,968,767
TimeStamp Reply	7	0	0
Protocol-snmp.rules			
Request TCP	473	692	3334
Trap TCP	86	364	1275
AgentX/TCP request	26	476	425
Deleted.rules (not included in Snort v2.9.6.0)			
Redirect Host	23,375	169,817	102,122
Redirect Net	426	10,135	2,344
Source Quench	8,319	1,940	1,660
Destination Unreachable Communication Administratively Prohibited	217,046	1,027,602	703,227
Destination Unreachable Communication with Destination Host is Administratively Prohibited	80	17,183	6,568
Destination Unreachable Communication with Destination Network is Administratively Prohibited	51,186	8,682	938
Large ICMP Packet	0	3,663	0
TCP Port 0 Traffic	6,572	2,874	3,188
Same SRC/DST	2,734	1,321	631
IP Reserved Bit Set	1,200	337	23
Bad Frag Bits	207	81	0

In this section, we evaluate the performances of two well-known NIDSs in terms of successful backscatter detection rate by employing darknet datasets that include mostly backscatter traffic. We do not state that Snort and Bro are insufficient

TABLE VIII: IDS Performance Analysis

Dataset		Snort v.2.9.6.0	Snort v2.9.1	Bro v2.2	Iatmon v2.1.2 (only on Nov-2008)
May, 28, 2004	O.B.R	0.01%	0.04%	0.23%	3.3%
	A.R	2%	1.2%	2.1%	
Feb 23, 2006	O.B.R	0.01%	0.01%	11.6%	
	A.R	15.8%	17.2%	15%	
Feb 22, 2008	O.B.R	0.05%	0.01%	5.2%	
	A.R	10.6%	11.6%	5.5%	

[1]O.B.R = Only Backscatter Records, A.R. = All Records

NIDSs, but demonstrate that they are quite inefficient to detect these type of attacks. We also show that even a one-way network traffic analyzer tool, Iatmon, cannot detect such attack behaviour correctly. We think that these are important findings since most of the botnet techniques make the use of backscatter (DDoS) techniques when used for attacking.

V. Conclusion and Future Work

In this paper, we focused on three different darknet (backscatter) datasets from year 2004, 2006 and 2008 provided by CAIDA. These datasets profile a general overview of the backscatter traffic of the time they were captured. Thus, our aim was to discover the trends/ patterns of backscatter traffic and how it changes (if at all) over time. To achieve this, we measure the different network and application level protocols in this traffic as well as analyze from where they originate. Moreover, to the best of our knowledge this is the first time where two well known IDSs are used to evaluate such traffic to understand whether any patterns can be discovered. Our results show that the patterns of backscatter traffic are changing over time. Furthermore, these IDS rules / signatures seem to be insufficient to analyze backscatter traffic from one year to the other. To this end, we presented and discussed the rule sets used by these systems as well as the triggered rules in Snort versions 2.9.1 and 2.9.6.0, and also Bro v2.2. As a summary of our results over these datasets, we give the following list:

- TCP seems to be the major protocol for the transport layer attack / backscatter traffic behaviour.
- HTTP (port 80) seems to be the major protocol for the application layer.
- SSL (port 443) seems to be the most used secure port in 2004 and 2008. However in 2006 SSH (port 22) is used more.
- Soulseek, BitTorrent and eDonkey seem to be the most popular P2P applications in 2004, 2006 and 2008, respectively.
- In these datasets, secure traffic is more than P2P traffic. However, both of them are a small portion of the overall traffic.
- It seems that the IP addresses from China and USA play the major role of generating DDoS attacks in these datasets.

- Misconfiguration data seems to affect Snort IDS negatively, causing high false alarm rates. On the other hand, misconfiguration traffic seems to have less effect on Bro IDS.
- Bro seems to identify the backscatter traffic better than Snort and Iatmon. However, the accuracy is still low, only 15%.

Our analysis also demonstrates that it is not completely possible to predict the backscatter behaviours of a specific year by evaluating the behaviours in the previous years. It is interesting to discover that it is challenging to identify backscatter traffic with the current intrusion detection systems or one-way traffic analysis tools. Given that such traffic is the first phase of botnet related malicious activities, this seems to suggest that the current open source technologies are not able to identify such behaviours.

It should be noted here that our findings regarding the frequent usage of TCP packets for such attacks and the majority of them generating from China support some of the results obtained by the previous work in the field, too. However, different from the previous work, we also demonstrate that: i) The used port numbers for network attacks are changing over the years. ii) It is even possible to use encrypted ports to generate network attacks. iii) P2P applications are not used for generating DDoS attacks. iv) Snort, Bro and Iatmon are not reliable in detecting backscatter traffic. Note that Bro is the most successful among the three (with the highest 15% detection rate) while detecting only backscatter packets.

Future work will explore the performances of the commercial off the shelf intrusion detection systems as well as the network analyzers designed by CAIDA such as Corsaro and Coralreef on such datasets. Furthermore, we will analyze more recent and larger darknet and backscatter datasets as they become available to explore whether they have similar patterns as well. In addition, we will explore the use of machine learning in order to analyze such traffic.

ACKNOWLEDGEMENT

This research is supported by the Natural Science and Engineering Research Council of Canada (NSERC) grant, and is conducted as part of the Dalhousie NIMS Lab at: https://projects.cs.dal.ca/projectx/

REFERENCES

[1] Pavel Polityuk. *Ukrainian authorities suffer new cyber attacks.* Kiev, March 8th, 2014. Reuters http://www.reuters.com/article/2014/03/08/us-ukraine-cricis-cyberattack-idUSBREA270FU20140308

[2] *Launch day DDOS game offline.* http://www.wurmonline.com/2014/02/18/launch-day-ddos-game-offline/

[3] Michael Bailey, Evan Cooke, Farnam Jahanian, Andrew Myrick and Sushant Sinha. *Practical darknet measurement.* Information Sciences and Systems, 40th Annual Conference on pp. 1496-1501. 2006. IEEE.

[4] Masashi Eto, Daisuke Inoue, Jungsuk Song, Junji Nakazato, Kazuhiro Ohtaka, Koji Nakao. *nicter: a large-scale network incident analysis system: case studies for understanding threat landscape.* In Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (pp. 37-45). 2011. ACM.

[5] Qian Wang, Zesheng Chen, and Chao Chen. *Darknet-based inference of Internet worm temporal characteristics.* Information Forensics and Security, IEEE Transactions on 6.4: 1382-1393. 2011.

[6] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker and Stefan Savage. *Inferring internet denial-of-service activity.* ACM Transactions on Computer Systems (TOCS) 24.2: 115-139. 2006. ACM.

[7] Vinod Yegneswaran, Paul Barford, and Dave Plonka. "On the design and use of Internet sinks for network abuse monitoring." Recent Advances in Intrusion Detection. Springer, 2004.

[8] UCSD Network Telescope Backscatter Datasets (Restricted Datasets). http://www.caida.org/data/passive/backscatter_tocs_dataset.xml. Last access: January, 2014.

[9] *The UCSD Network Telescope.* http://www.caida.org/projects/network_telescope/

[10] Port 3072. http://www.pc-library.com/ports/tcp-udp-port/3072/

[11] Port 1026. http://kb.prismmicrosys.com/evtpass/evtpages/PortNo_1026_nterm_55033.asp

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

[13] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson.*Characteristics of internet background radiation.* In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. 2004.

[14] Hal Berghel. *Malware month.* Communications of the ACM, 46.12: 15-19, 2003.

[15] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. *Accurate, scalable in-network identification of p2p traffic using application signatures.* In Proceedings of the 13th international conference on World Wide Web, pp. 512-521. ACM, 2004.

[16] Thomas Karagiannis, Andre Broido, and Michalis Faloutsos. *Transport layer identification of P2P traffic.* In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pp. 121-134, 2004. ACM.

[17] Riyad Alshammari and A. Nur Zincir-Heywood. *Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?.* Computer networks 55.6 (2011): 1326-1350.

[18] Waste. http://waste.sourceforge.net/

[19] Wireshark. http://www.wireshark.org/

[20] Tshark. http://www.wireshark.org/docs/man-pages/tshark.html

[21] Geolite. http://dev.maxmind.com/geoip/legacy/geolite/

[22] Zouheir Trabelsi and Latifa Alketbi. *Using network packet generators and snort rules for teaching denial of service attacks.* In Proceedings of the 18th ACM conference on Innovation and technology in computer science education, pp. 285-290. 2013.

[23] Vern Paxson. *Bro: a system for detecting network intruders in real-time.* Computer networks 31, no. 23 (1999): 2435-2463.

[24] Snort. http://www.snort.org/

[25] Iatmon. http://www.caida.org/tools/measurement/iatmon/

[26] Thomas Berger. *Analysis of current VPN technologies.* In Availability, Reliability and Security, pp. 8. IEEE, 2006.

[27] SecureShell.http://www.rfc-archive.org/getrfc.php?rfc=4251

[28] Seungwon Shin, Jaeyeon Jung, and Hari Balakrishnan. *Malware prevalence in the KaZaA file-sharing network.*Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006.

[29] WikiLeaks hit by DoS attacks. http://www.secure64.com/news-wikileaks-dns-server-fire

[30] DDoS Attacks to gaming sites. http://www.arbornetworks.com/asert/2008/02/ddos-events-of-note-wordpress-gambling-sites/

[31] DoS attacks on blogs. http://news.netcraft.com/archives/2006/02/28/ddos_attacks_target_prominent_blogs.html

[32] DoS attacks on payment gateways. http://news.netcraft.com/archives/2006/02/10/payment_gateway_stormpay_battling_sustained_ddos_attack.html

[33] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston. *Internet background radiation revisited.*, In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 62-74. 2010.

[34] Brownlee, Nevil. "One-way traffic monitoring with iatmon." Passive and Active Measurement. Springer Berlin Heidelberg, 2012.