DDoS Attacks as an Important Network Phenomenon

Ilija Basicevic
and Miroslav Popovic

Faculty of Technical Sciences
University of Novi Sad, Serbia
Email: ilibas@uns.ac.rs
Telephone: (+381) 21 4801 242

Stanislav Ocovaj

RT-RK Institute for Computer Based Systems
Novi Sad, Serbia
Email: stanislav.ocovaj@rt-rk.com

***Abstract*—This paper presents several aspects of Distributed Denial of Service (DDoS) attacks which have been an important network phenomenon since the end of 1990s. A short introduction to this topic is given here. DoS mechanisms, such as reflection and amplification, are explained including example attacks. Also, DNS fast flux as a technique for hiding command and control communication (C&C) is described. A short overview of entropy based DDoS detection is given, and an example method explained. Certain methods for attack mitigation (most notably puzzles) are explained, as well.**

Keywords–network security; Distributed denial of service attacks (DDoS); entropy; DNS amplification attack; SYN Flood attack.

I. INTRODUCTION

DDoS attacks have already become part of the Internet landscape. Although unwanted part, there are currently no signs that they will go away. As it is well known, the aim of a DDoS is to stop or interrupt or slow down the operation of an Internet server (most often a web server, but it can be an e-mail, DNS, or other type of server). There are many ways to inhibit the operation of a computer or a network service, but Internet DoS attacks typically achieve that by depleting resources.

DoS attacks are important because of the impact they have on operation of companies that rely on Internet in their communication with customers. The targets of DoS are not only business companies but government and nongovernment agencies, too.

The first distributed DoS (DDoS) attack was registered in 1999. The attacker used Trinoo tool to attack University of Minnesota computer network [1]. Since then, the damages they inflict to business operations have been constantly rising.

The size of DoS attacks that is seen on the Internet keeps rising every year. In 2012, attacks of the size of 70 Gbps of noisy traffic were seen. In 2013, the largest DoS attack was 300 Gbps. Early in 2014, the attack of 400 Gbps was reported. In 2016, the attack launched by Mirai botnet surpassed 1 Tbps.

Criminal groups that extort money from business companies are often behind DDoS attacks. Companies have reported losing up to $100000 (some even more) per downtime hour during DoS attacks [2]. Companies often decide to pay criminals because the law procedure is slow and criminal groups are often in different countries, which complicates the procedure. There are also politically motivated attacks, which are often led by cyber-terrorists or by countries in conflict. For an example of politically motivated attack, see [3] about DoS attack in Estonia in 2006.

Another purpose of DoS is that it can be used (and is used more and more) to cover up traces of an intrusion - as a distraction mechanism or, to explain in a more figurative way, as a smoke shield. While security and IT officers of a company deal with DoS, the attackers can more easily intrude the company information system undetected.

Having in mind the protocol stack reference model (ISO OSI), the attack can be realized at different levels of the protocol stack.

II. DISTRIBUTED DOS

As already noted, there are many ways to produce denial of service effect. Distributed DoS attacks are usually realized either as:

- Bandwidth attacks attack on the availability of network links to the server, or as
- Resource starvation attacks attack on the availability of resources (e.g. memory) at the server.

Some well-known examples of resource starvation attacks are:

- Sending XML documents with deeply nested schemes, and
- Sending large number of packets that require crypto operations, as those are usually resource intensive.

DoS attacks at network servers are usually realized in a distributed manner (Distributed DoS, DDoS). Typically, there is a large number of computers (thousands or tens of thousands) that are connected to the Internet, and that take part in the attack. The owners of computers are usually not aware that their computers participate in the attack. The computers (sometimes called zombies) are usually infected by malware which installs DoS agents on them – thus they get under the control of the attacker. These computers form a bot net, which is controlled by the attacker.

The aim of attackers is to avoid tracing the IP address of the command computer (and its owner) and special measures are taken to hide the so called Command&Control communication between bot net computers and the attacker.

A. Mathematical model of DDoS

In [4], authors model the system under SYN flood DDoS attack as a two-dimensional queuing model with N servers, two arrival processes and two service times of different distribution. Both the arrival of regular request packets and the arrival of attack packets are modeled as Poisson processes, but with

 ISBN: 978-1-61208-633-0

different arrival rates λ_1 and λ_2. This is in accordance with the prevalent view on properties of Internet traffic [5], [6]. At most N half-open connections are allowed at one moment. Half-open connection for a regular request packet is held for random time which is exponentially distributed. The two arrival processes are independent of each other and of holding times for half-open connections. Based on these assumptions, DDoS is modeled as two-dimensional embedded Markov chain. The model allows calculation of security metrics such as:

- Connection loss probability, and
- Buffer occupancy percentage of half-open connections for regular traffic.

B. DNS fast flux

Domain name system (DNS) fast flux is a technique that attackers can use to hide the C&C center. There are two variants of this technique: single and double fast flux. In the single fast flux, a large set of addresses (hundreds or thousands) is associated with a domain name. These addresses are swapped in and out (for example in a round robin fashion) with high frequency each set of assigned addresses is changed after less than 5 minutes. The computer that connects to a web server every 5 minutes will connect each time to a different address.

In the double fast flux, the same is done as in the single version, but this time with an authoritative name server responsible for the entire DNS zone (containing multiple domains) and not only with a single domain name, as in the single fast flux. This time, the IP address of authoritative Name Server is also changing constantly. This gives an additional layer of protection to the attacker.

C. Reflection and amplification

Reflection is an important mechanism that is used by certain DoS attacks. In that case, the attack traffic is sent to a reflector that replies by sending messages to the source address of received messages, which is falsified (spoofed), and points incorrectly to the target of the attack. Thus, the reflected traffic is directed to the target. Amplification is the effect that the attacker tries to achieve: traffic to the target should be larger than the traffic sent by attackers by some amplification factor. Well-known reflected attacks are:

- Smurf (realized by sending Internet Control Message Protocol (ICMP) Echo Request -ping- with spoofed source address to the broadcast address), Figure 1 and
- Fraggle (realized by sending User Datagram Protocol (UDP) echo with spoofed source address to the broadcast address).

These two attacks presented a real threat in 1990s but today, most of the networks are configured so as not to be vulnerable to those two attacks (IP-directed broadcast is disabled at routers).

D. DNS amplification attack

In this type of attack, the attacker uses publicly accessible DNS servers to flood the target with DNS response traffic. Members of the botnet send a large number of DNS name lookup requests to the DNS server with spoofed source address. Usually, DNS requests are for many names and with the type ANY so that responses are larger and the amplification factor is increased. The ANY type requires all known information to be returned for the specified name. As the response traffic originates from public DNS servers and as it is legitimate data, it is hard to prevent it.

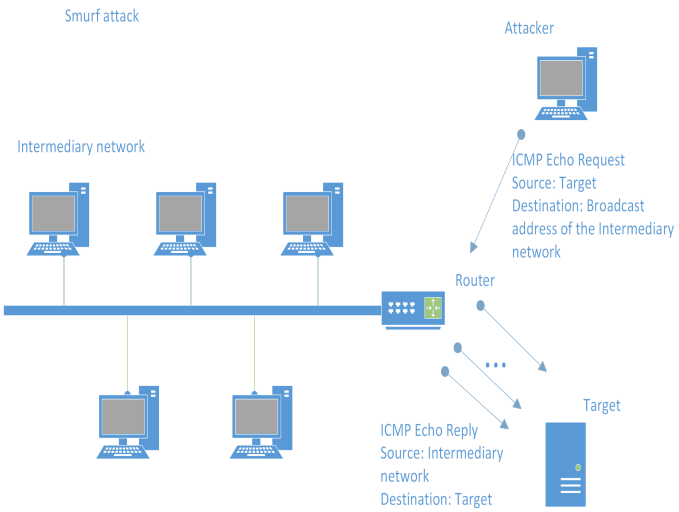

Figure 1. Smurf DDoS attack.

Variations of this attack can include attackers compromising the DNS server and adding large TXT RR records – to increase the amplification factor. TXT RR allows arbitrary text to be inserted in the DNS record. There are existing legitimate large records, so this step is not necessary.

E. SYN Flood attack

This is one of the most frequent DoS attacks for several years and probably the most investigated attack in scientific publications. Boteanu and Fernandez call it Mother of all DoS attacks [7]. In the course of the attack, attackers initiate a large number of Transmission Control Protocol (TCP) three-way handshakes but do not actually complete them.

Thus, resources at the server that are allocated for connection establishment and in normal case released when the connection enters the active state (or is prematurely closed), stay allocated, which leads to resource starvation. Upon receiving the SYN packet, the TCP server side enters the SYNRECEIVED state and starts the timer with the duration of typically 75 s [8].

The attack does not aim to overload the end hosts memory but simply to exhaust the so-called backlog of half-open connections associated with a port number. Backlog is a system limit on a number of TCP Control Block (TCB) structures that can be resident at any time [9]. To achieve the desired effect, the attacker should send new barrage of connection requests as soon as the attacked system starts to reclaim TCB blocks allocated during the previous barrage. The frequency of attack has to be adapted to the TCB reclamation timer. The greater frequency increases the risk of detection without adding to the effectiveness of the attack.

 ISBN: 978-1-61208-633-0

F. Shrew attack

This is a low rate attack that uses attack stream of a square waveform [10] with the following parameters: period T, burst length L, peak rate R. It has the following properties [11]:

- R is enough to exceed link capacity in combination with baseline traffic
- L is long enough to induce timeout (typically greater than round trip time RTT)
- T is scaled in accordance with the minimum retransmission timeout (RTO)

The logic behind the attack is to let TCP module detect that the link is congested. After the initial attack burst, the TCP will wait until the expiry of retransmission timeout. When it does retransmit, it will collide with one of the subsequent attack bursts. As a result, the TCP can experience very low (near zero) throughput and connection close.

III. Methods for detection of DDoS

There are two main classes of DoS attack detection methods: volume based and feature based.

- Feature-based methods rely on inspection of packet headers.
- Volume-based methods monitor changes in traffic volume.

The attackers goal is to achieve DoS effect with as little attack traffic as possible – in order to avoid detection. Having that in mind, the importance of feature-based detection is easier to understand. There are known advantages of feature-based over volume-based methods in detection of small volume attack traffic [12].

A. Entropy based detection

Among feature based methods, entropy has a very important place. It is obvious that network events, such as DDoS attacks and port scans, change the randomness of packets addresses and ports. The most important advantage of entropy is its generality. An entropy based method can detect a wide range of network traffic anomalies (including DDoS attacks, but also flash crowds, and other types of events). Ref [13] presents a comparison of the entropy based method for detection of DDoS attacks and a custom tailored method for detection of SYN Flood attacks [14]. The result of the comparison is that with respect to detection performance, the entropy based method comes close to custom tailored one in detection of SYN Flood attacks, but on the other hand the entropy based method can detect UDP DDoS as well, while the custom tailored method is completely unusable in detection of attacks other than SYN Flood.

Thus, the number of research papers that propose the use of entropy is not small, and in the continuation, we will mention some of them.

Authors experimented with the use of Shannon entropy ([15], [16], [13] and others):

$$H(Z) = -\sum_{i=1}^{N} p(z_i) log(p(z_i)), \quad (1)$$

where $p(z_i)$ is probability that Z takes the value z_i. The entropy value is then normalized by the log (N), where N is the number of distinct z_i values that appear in the observed interval. All logarithms are with the base 2, because the information is represented in bits.

Shannon entropy has following properties:

- Nonnegativity

$$H(Z) \geq 0, \forall p(z_i) \in [0, 1], \quad (2)$$

- Symmetry

$$H(p(z_1), p(z_2), ...) = H(p(z_2), p(z_1), ...), \quad (3)$$

- Maximality

$$H(p(z_1), p(z_2), ..., p(z_n)) \leq H(\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}), \quad (4)$$

- Additivity

$$H(x, y) = H(x) + H(y), \quad (5)$$

if x and y independent variables.

Another entropy formula that is attractive to researchers is Tsallis entropy [17]:

$$H(Z) = \frac{1 - \sum_{i=1}^{N} p_{z_i}^q}{q - 1}, \quad (6)$$

the value of H is in the range (0, H_q^{max}), and

$$H_q^{max} = \frac{1 - N^{1-q}}{q - 1}, \quad (7)$$

Shannon entropy is extensive, while Tsallis is not – it does not have the property of additivity. For q tending to 1, Tsallis formula converges to Shannon. Tsallis has proposed the formula for the investigation of the properties of systems that exhibit fractal and long range dependent behavior. Accordingly, the motivation for the use of Tsallis formula in detection of network traffic anomalies is the well-established notion that Internet traffic is self-similar and long range dependent. Those properties have been discovered more than twenty years ago [18], [19], [20]. Earlier view is that self-similarity is stronger with the increase of network utilization [18]. More recent view is that with a traffic increase, inter arrival process tends to Poisson and packet sizes to be independent [5]. Also, it is noted that on sub second time scales, network traffic can be well represented by the Poisson model [6].

One of the early papers on the use of Tsallis entropy in detection of DDoS is [21]. Tellenbach et al [22] have proposed the use of Traffic Entropy Spectrum (TES) that is based on the use of Tsallis entropy. In ref [23], performance of Tsallis formula is compared to the one of Shannon formula. The conclusion is that Tsallis formula can outperform Shannon in detection of DDoS attacks, but that requires careful tuning of Tsallis Q parameter and there is no universal Q value that performs best in all detection scenarios.

Another generalization of Shannon entropy is Renyi entropy. Renyi entropy is extensive, and for α tending to 1, it converges to Shannon entropy.

$$H(Z) = \frac{1}{1 - \alpha} \log \sum_{i=1}^{N} p(z_i)^{\alpha}, \quad (8)$$

In refs [24], [25] and [26] authors have compared the performance of those formulas. In [24], authors conclude that

with respect to DoS detection, generalized entropy measures outperform Shannon. In [25], authors conclude that generalized entropies and feature-based distributions perform better than Shannon entropy and counter-based methods. The important conclusion of their research is that for successful detection of different anomalies, a wide range of distributions should be used.

To determine the extent of changes between observed and assumed distributions, Kullback-Leibler divergence (K-L) can be used [26]. The formula for K-L divergence is:

$$D_{KL}(p,q) = \sum_{i=1}^{N} p(i) log(\frac{p(i)}{q(i)}, \quad (9)$$

It is also used with the maximum entropy, the application of which is proposed by some researchers, see [27], [28].

The change of randomness is not enough to certainly indicate an attack. Some limitations of entropy are overcome with the use of Kolmogorov complexity [29]. There is a known problem with efficient calculation of Kolmogorov complexity, which today prevents its use in online detection. As an alternative, researchers propose the use of Titchener (T-entropy) [30].

The entropy is calculated on certain distributions of network traffic parameters. Those distributions can be simple (e.g. source and destination addresses/ports) or complex. The most important complex distribution is a flow size distribution (FSD). Flow is a sequence of packets exchanged between two endpoints. The endpoints are defined with a 5-tuple (SrcIp, SrcPort, DstIp, DstPort, Protocol) containing source and destination addresses and ports, and the protocol used for communication. In case of TCP, the flow corresponds to TCP connection. In case of UDP, the flow is defined using maximum allowed time between two consecutive packets in a flow. A comparison on the usability of FSD versus distribution of addresses is given in [31], where it is concluded that FSD in certain scenarios outperforms simple packet distributions.

B. Change point detection

The detection method used in [13], [31], [23] assumes that entropy time series are subject to change point detection algorithm, more specifically CUSUM [32]. On the other hand, the approach in [27] avoids the use of change point detection.

CUSUM is used in many application fields and it is based on hypothesis testing. The input time series are independent and identically distributed random variables that are bounded by a finite value. Two hypotheses define distribution before and after the change. The formulas used in [13], [31], [23] are given here:

$$\mu_n = \beta_1 y_n + (1-\beta_1)\mu_{n-1} \quad (10)$$

$$d_n = max\left\{0, d_{n-1} + y_n - (\mu_n + K)\right\}, d_0 = 0 \quad (11)$$

$$\sigma_n^2 = \beta_2 (y_n - \mu_n)^2 + (1-\beta_2)\sigma_{n-1}^2, H = h\sigma_n \quad (12)$$

If $d_n > H$, a change is reported ($g_n = 1$). Thus, H is the decision threshold. μ_n is an estimation of average value of time series y_n. Counter d_n accumulates deviations of y_n from μ_n that are greater than K. K is typically set to the half of the minimum shift to be detected, here it is set to 0.3. β_1 and β_2 are EWMA adaptation factors - their values are 0.2 and 0.05 respectively. σ_n is an estimation of the standard deviation of time series y_n.

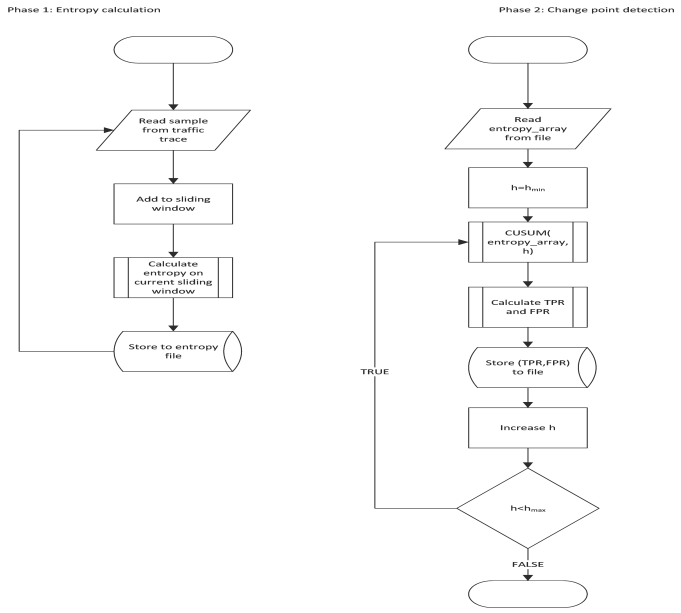

Figure 2. An example of the detection method.

C. The overall method of detection

As an example, the method of detection used in [13], [31], [23] is presented in Figure 2. The traffic trace file is input to the entropy calculation program. For the calculation of entropy, sliding window of 1 s (10 sub-intervals of 0.1 seconds) is used. The output of entropy calculation program is a file containing entropy time series. In the next step, CUSUM is applied to entropy time series for change point detection. The CUSUM h parameter is iterated in the range [0, 20]. For each value, detection is performed and true positive rate (TPR) and false positive rate (FPR) are calculated and stored in a file. The file is used in the last phase for the creation of receiver operating curves (RoC).

The detector perfoemance can be further improved with the use of Takagi-Sugeno-Kang fuzzy method, see [33].

D. The dataset problem

Basically, there are two options: simulations and analysis of real traffic traces. The problem with the second one, which is somehow more appealing for a scientist, is that the number of existing traces available online is rather small and, to the best of our knowledge, there is no existing labeled real network trace that contains both baseline and DoS attack traffic. So, researchers most often opt for two approaches:

- Simulations (ns2 [34], ns3 [35], omnet++ [36], etc.) and
- Injection of artificial DoS traffic to real network traces.

The problem with simulations is, as always, how realistic they are. The problem with injection, is (among other things), that in such a trace is not visible the reaction of the target server and other network subjects to the DoS i.e. the server is less responsive to legitimate clients during the attack thus, they repeat connection requests, etc.

Some of real datasets that are still used in research of DDoS are outdated (e.g. [37]). For an overview of existing datasets, see [38]. Another possibility are emulation testbeds, such as

 ISBN: 978-1-61208-633-0

[39] and [40]. They integrate simulation and real systems, using soft routers.

IV. The reaction to DDoS - attack mitigation

The defense mechanisms can be classified into response mechanisms (primarily filtering, rate-limiting, and capability methods) and tolerance mechanisms (congestion policing, fault tolerance, and resource accounting). Tolerance mechanisms do not rely on attack detection, but in some cases, they are expensive and lead to inefficient use of overprovisioned resources (the case of fault tolerance methods). For a more comprehensive overview of defense mechanisms, see [41].

The first defense measure that has been applied by Internet Service Providers (ISPs) and carriers (for years) has been blackholing the target IP address. The router that has detected a DoS attack will blackhole the target IP address. All traffic destined to that address (normal and attack) will be discarded. The goal of the attack is achieved completely but on the other hand, other hosts in the network have been spared.

A more recent defense technique is based on the use of a scrubbing center. The router that detects DoS attack reroutes the traffic through the scrubbing center. The scrubbing center is typically located in a cloud and it will remove attack packets from the traffic. The problem with out-of-band scrubbing centers is that re-routing decision is done by a human analyst. Thus, there is time required for attack mitigation to take place. A series of short attacks can evade the detector.

One of resource accounting mechanisms attempts to balance the workload between clients and servers by introducing cryptographic puzzles, Figure 3. In that case, the client is required to solve the puzzle before the server allocates resources to processing the clients request.

Figure 3 shows the order of messages in a system which supports puzzles. Upon receiving the request from the client, the protected server forwards the request to the puzzle generator. The puzzle generator generates a puzzle and sends it to the client. When it solves the puzzle, the client sends the solution to the puzzle generator, which checks the solution. Only when the puzzle has been solved, the puzzle generator informs the protected server, which continues the communication with the client.

The puzzles use cryptographic mechanisms that make it very hard for clients to avoid solving the puzzle. Even in such a case, attackers that control large botnets are in a much better position than legitimate clients and actually, the targeted balance of workload between the server and the attacker is not achieved.

A more promising approach is the use of guided tour puzzles. The limiting factor in this case is not the clients Central Processing Unit (CPU) power but the round trip time of the network path, which the attacker cannot overcome by multiplying CPU resources [42].

V. Conclusion

The paper presents the perspective of distributed denial of service attacks. A short introduction is given, including the motivation, size and historical beginnings of this network phenomenon. Important mechanisms that are used by attackers (reflection, amplification, hiding of command and control communication) are described and examples are given. A short overview of entropy based DDoS attack detection is provided as well as a description of certain attack mitigation techniques (including scrubbing centers and puzzles).

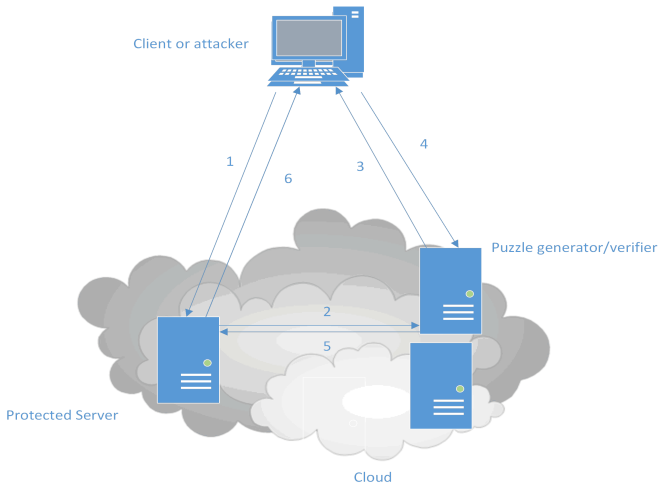

Figure 3. Message exchange in a system which supports puzzles.

Acknowledgment

This work has been partially supported by the Ministry of Education and Science of the Republic of Serbia under the Project TR32031.

References

[1] "DDoS attacks history," URL: https://security.radware.com/ddos-knowledge-center/ddos-chronicles/ddos-attacks-history/ [retrieved: 1, 2018].

[2] T. Matthews, "Incapsula Survey : What DDoS Attacks Really Cost Businesses," 2014.

[3] "A brief history of DDoS attacks," 2016, URL: https://eugene.kaspersky.com/2016/12/06/a-brief-history-of-ddos-attacks/ [retrieved: 1, 2018].

[4] Y. Wang, C. Lin, Q.-L. Li, and Y. Fang, "A queueing analysis for the denial of service (dos) attacks in computer networks," Comput. Netw., vol. 51, no. 12, Aug. 2007, pp. 3564–3573.

[5] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "Internet traffic tends to poisson and independent as the load increases," Bell Labs, Tech. Rep., 2001.

[6] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary poisson view of internet traffic," in Proceedings of IEEE INFOCOM, 2004, pp. 1558–1569.

[7] D. Boteanu and J. M. Fernandez, "A comprehensive study of queue management as a DoS counter-measure," International Journal of Information Security, vol. 12, no. 5, 2013, pp. 347–382.

[8] H. Wang, D. Zhang, and K. Shin, "Detecting SYN flooding attacks," in Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 2002, pp. 1530–1539.

[9] W. Eddy, "TCP SYN flooding attacks and common mitigations," 2007, rfc: RFC 4987.

[10] A. Kuzmanovic and E. W. Knightly, "Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants," in 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03), 2003, pp. 75–86.

[11] C.-W. Chang, S. Lee, B. Lin, and J. Wang, "The taming of the shrew: Mitigating low-rate tcp-targeted attack," IEEE Transactions On Network Service Management, vol. 7, no. 1, 2010.

 ISBN: 978-1-61208-633-0

[12] P. Du and S. Abe, “Detecting dos attacks using packet size distribution,” in 2007 2nd Bio-Inspired Models of Network, Information and Computing Systems, 2007, pp. 93–96.

[13] I. Basicevic, S. Ocovaj, and M. Popovic, “Evaluation of entropy-based detection of outbound denial-of-service attacks in edge networks,” Security and Communication Networks, vol. 8, 2015, pp. 837–844.

[14] V. A. Siris and F. Papagalou, “Application of anomaly detection algorithms for detecting syn flooding attacks,” in Globecom, 2004, pp. 2050–2054.

[15] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, “An empirical evaluation of entropy-based traffic anomaly detection,” in Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, 2008, pp. 151–156.

[16] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” in Proceedings of the ACM SIGCOMM 2005, 2005, pp. 217–228.

[17] C. Tsallis, “Possible generalization of boltzmann-gibbs statistics,” Journal of Statistical Physics, vol. 52, no. 1-2, 1988, pp. 479–487.

[18] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic,” in SIGCOMM ’93 Conference proceedings on Communications architectures, protocols and applications, 1993, pp. 183–193.

[19] M. Crovella and A. Bestavros, “Self-similarity in world wide web traffic: evidence and possible causes,” in Proceedings of the 1996 ACM SIGMETRICS international conference on measurement and modeling of computer systems, 1996, pp. 160–169.

[20] V. Paxson and S. Floyd, “Wide area traffic: the failure of poisson modeling,” IEEE/ACM Transactions on Networking, vol. 3, no. 3, 1995.

[21] A. Ziviani, A. T. A. Gomes, M. L. Monsores, and P. S. S. Rodrigues, “Network anomaly detection using nonextensive entropy,” IEEE Communications Letters, vol. 11, no. 12, 2007.

[22] B. Tellenbach, M. Burkhart, D. Sornette, and T. Maillart, “Beyond shannon: Characterizing internet traffic with generalized entropy metrics,” in Proceedings of the 10th International Conference on Passive and Active Network Measurement, 2009, pp. 239–248.

[23] I. Basicevic, S. Ocovaj, and M. Popovic, “Use of tsallis entropy in detection of syn flood dos attacks,” Security and Communication Networks, vol. 8, 2015, pp. 3634–3640.

[24] C. Lima, F. de Assis, and C. de Souza, “A comparative study of use of shannon, rnyi and tsallis entropy for attribute selecting in network intrusion detection,” in Intelligent Data Engineering and Automated Learning - IDEAL 2012. Lecture Notes in Computer Science, vol 7435. Springer, Berlin, Heidelberg, 2012.

[25] P. Bereziski, J. Pawelec, M. Maowidzki, and R. Piotrowsk, “Entropy-based internet traffic anomaly detection: A case study,” in Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, 2014, pp. 47–58.

[26] P. Berezinski, B. Jasiul, and M. Szpyrka, “An entropy-based network anomaly detection method,” Entropy, no. 17, 2015, pp. 2367–2408.

[27] Y. Gu, A. McCallum, and D. Towsley, “Detecting anomalies in network traffic using maximum entropy estimation,” in 5th ACM SIGCOMM conference on Internet measurement (IMC ’05), 2005, pp. 32–32.

[28] A. Coluccia, “Glrt-based change detection for non-stationary data via maximum entropy,” in GTTI Annual Meeting, 2010.

[29] A. Wagner and B. Plattner, “Entropy based worm and anomaly detection in fast ip networks,” in 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE’05), 2005, pp. 172–177.

[30] U. Speidel, R. Eimann, and N. Brownlee, “Detecting network events via t-entropy,” in 6th International Conference on Information, Communications Signal Processing, 2007, pp. 1–5.

[31] I. Basicevic, S. Ocovaj, and M. Popovic, “The value of flow size distribution in entropy-based detection of dos attacks,” Security and Communication Networks, vol. 9, 2016, pp. 958–965.

[32] E. S. Page, “Continuous inspection scheme,” Biometrika, vol. 41, no. 1/2, 1954, pp. 100–115.

[33] M. Petkovic, I. Basicevic, D. Kukolj, and M. Popovic, “Evaluation of takagi-sugeno-kang fuzzy method in entropy-based detection of ddos attacks,” Computer Science and Information Systems, vol. 15, no. 1, 2016, pp. 139–162.

[34] “The network simulator - ns2,” http://www.isi.edu/nsnam/ns [retrieved: 3, 2018].

[35] “The network simulator - ns3,” https://www.nsnam.org/ [retrieved: 3, 2018].

[36] “Omnet++ discrete event simulator,” https://www.omnetpp.org/ [retrieved: 3, 2018].

[37] “M. i. t. lincoln laboratory, darpa intrusion detection evaluation data set,” https://www.ll.mit.edu/ideval/data/ [retrieved: 3, 2018], 2018.

[38] S. Behal and K. Kumar, “Trends in validation of ddos research,” Procedia Computer Science, vol. 85, no. 2016, 2016, pp. 7–15.

[39] “Emulab - network emulation testbed home,” https://www.emulab.net/ [retrieved: 3, 2018].

[40] “The deter project,” https://deter-project.org/ [retrieved: 3, 2018].

[41] M. Abliz, “Internet denial of service attacks and defense mechanisms,” 2011, report: TR-11-178, University of Pittsburgh.

[42] M. Abliz, T. Znati, and A. Lee, “Mitigating distributed service flooding attacks with guided tour puzzles,” International Journal on Advances in Security, vol. 5, 2012, pp. 121–133.

 ISBN: 978-1-61208-633-0