International Journal of Research in Advent Technology, Vol.6, No.8, August 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Botnet Detection Technique Using Denial of Service (DDOS) Attack Elliptic Curve Digital Signature (ECDSA) Algorithm

[1]S.Soundharya, [2]K.RaviKumar
[1]*Research Scholar, Dept. of. Computer Science, Tamil University, Thanjavur-613010*
[2]*Asst.professor, Dept. of. Computer Science, Tamil University, Thanjavur-613010*
[1]soundharyashankar93@gmail.com,
[2]ravikasi2001@yahoo.com

Abstract-Distributed Denial-of-Service (DDoS) overwhelm are frequently launched through the botnet, an "army" of compromised nodes hidden in the network. Unfortunately, the recent emergence of attacks performed at the application layer has multiplied the number of possibilities that a botnet can exploit to conceal its malicious activities. New challenges arise, which cannot be addressed by simply borrowing the tools that have been successfully applied so far to earlier DDoS paradigms. In this vocation, I offer fundamentally three assistance: i). Introduce nonfigurative model for the aforesaid class of attacks, where the botnet emulates normal traffic by continually learning admissible patterns from the environment; ii). An inference algorithm that is shown to provide a consistent (i.e., converging to the true solution as time elapses) estimate of the botnet possibly hidden in the network; and iii). Verify the validity of the proposed inferential strategy on a testbed environment. iv). Elliptic curve Digital Signature Algorithm (ECDSA) is more recently standardized and supposedly reducing size of digital signatures and cryptographic keys. Our tests show that, for several scenarios of implementation, the proposed botnet identification algorithm needs an observation time in the order of (or even less than) one minute to identify correctly almost all bots, without affecting the normal users' activity.

Index Terms-Distributed Denial-of-Service (DDoS) attacks, Botnet, Learning Admissible Patterns, inference algorithm, Elliptic curve Digital Signature Algorithm

1. INTRODUCTION

The rapid expansion of the Internet over the past decade appeared to have facilitated an augment in the incident of online attacks [1]. One such influential and harmful attack is the denial of service (DoS) attack. A DoS attack significantly threatens the network, particularly if such an attack is distributed. A distributed DoS (DDoS) attack is launched by aninstrument called Botnet through a network of guarded computers. A software program controls the computers and for specific purposes, known as ―bots. Bots are small scripts that have been designed to perform specific, programmed functions. Bots are utilized by agents for Web indexing or ―spidering,as well as to collect online product prices or to performing such duties as chatting. However, bots are unenthusiastically associated with ―remote access Trojan Horses‖ (e.g., Zeus bot) and automaton computers that are created for less favourable purposes [2]. Bots in large quantity provide the power of a processor to create most important tools for such activities as the extensive delivery of SPAM email, click-fraud, spyware installation, virus and worm dissemination, and DDoSattack (e.g., black energy bot) [3]. DDoS attacks usually take advantage of the weaknesses of a network layer, predominantly, SYN, UDP, and Internet control message protocol (ICMP) inundation. Such attacks encroach the network bandwidth and resources of the victim, thus facilitate the denial of legitimate access. A DDoS attack is exemplified by the direct attempt of aggressor to prevent legitimate users from using a specific service [4]. A recent, convoluted, and well-liked method of DDoS attack engagesubmission level flooding, especially in the Web server. Such assault employs various flooding methodologies (e.g., HTTP-GET flood, etc). It can see that HTTP attacks position first in terms of numeral of incidents. HTTPs registered the uppermost incidence of DDoS attacks in 2010, reaching up to 100 Gbps in 2011. This increase accounts for a 700% rise in incidents, as reported by the Cloud Flare Company [6], where the HTTP attacks encompass approximately 80% in

2010, a value that significantly increased to in the region of 88% in 2011. The number of daily objective Web sites unmistakably increased, with management websites becoming anordinary target [5].

2. RELATED WORK

In this situation, the single source of the attack canbe identified by work out its unusually large request rate.The *distributed* alternative of such attacks exploit basically thesame kind of vulnerabilities and repetition schemes, but for thefact that the *large* request rate is now obtained by aggregating*many small* personage bot rates. Nevertheless, in such attacks,the bots can be still identified at a single-user level. Indeed,normal traffic patterns are typically characterized by a certaindegree of innovation, while the reiterationschemes implicitly highlight the bot character. In fact, several useful inferentialstrategies have been projected for such kind of DDoS attacks.The journalism about DDoS attacks is rich. With no pretenceof completeness, we bring in briefly some recent works onthe subject, and we refer the Reader to the survey in [2] fora more comprehensive summary. In [3], arithmetical methodsto recognizeDDoS attacks are proposed, relying on computeentropy and frequency-sorted distributions of selected small packageattributes.

The DDoSrecognition is then based on the discoveryof anomalies in the individuality of the packet attributes.In [4], the Authors propose a hierarchical method based onmacroscopic-level network monitoring to capture shifts inspatial-temporal traffic patterns, which are then used to informa discovery system about where and when a DDoSinundationattack possibly arises in a source set of connections. The work presentedin [5] relies on the application of an entropy detection method,where the key to identify the DDoS attack is the unpredictabilityof some attributes in the packets' headers. In [6], two newinformation metrics, the generalized entropy metric and theinformation distance metric, are employed to detect low rateDDoS attacks, by evaluating the dissimilarity betweenlegitimate and attack traffic.

A statistical model to examineshrew DDoS attacks (where TCP flows are constrained toa small portion of their ideal rate at low attack costs) is introduced in [7]. The Authors propose aattitude aimedat capturing the modification behaviors of TCP congestioncasement at the victim's side, in order to evaluate the interactionbetween attack patterns and set of connectionssurroundings.More closely linked to this work is the new class of*application-layer* DDoS attacks, which is recently emergingas one of the most powerful threats [8]–[10]. In such attacks,the malicious transfer patterns are disguised as normal onesby leveraging the many potential offered at the applicationlayer (for instance, when surfing through a website, more andmore web-pages are likely to be explore as time elapses). Byassigning anenough degree of variability to each individualbot's prototype, identification strategies based on single-user inspectionbecome undamaging. Building on such new potential,in this work we shall introduce a formal model for DDoSattacks where the botnet gets at its discarding a certain *emulationdictionary* to build the interchangeprototype.

3. SYSTEM ANALYSIS

3.1 Existing System

In this situation, the single source of the attack can be identified by computing its unusually large request rate. The distributed variants of such attacks exploit basically the same kind of vulnerabilities and repetition schemes, but for the fact that the large request rate is now obtained by aggregating many small individual bot rates. Nevertheless, in such attacks, the bots can be still identified at a single-user level. Indeed, standardinterchangeprototype is typically characterized by a certain degree of innovation, while the repetition scheme implicitly emphasizes the bot character.

Disadvantages

- The physical layer security is not so secured if the flooding attack and wormhole attack is established.
- Only concentrated on the end to end delay omission. End to end delay can be introduced by the attacker by the DOS attacks.
- Effective routing calculation can be omitted from the existing work because the devices are most possible to communicate directly. Curve Cryptography is not used, so Key size is high.

3.2 Proposed System

The proposed botnet identification algorithm needs an observation time in the order of (or even

less than) one minute to identify correctly almost all bots, without affecting the normal users' activity. Statistical methods to identify DDoS attacks are proposed, relying on computing entropy and frequency-sorted distributions of selected packet attributes. The DDoS identification is then based on the detection of anomalies in the characteristics of the packet attributes. Description of the botnet identification algorithm, it is worth commenting on a possible limitation of the proposed approach. There might be particular situations where the BIC is violated because some normal users, even if acting in uncoordinated manner

Advantages

- Security can be implemented on the physical layer of the wearable devices in BAN.
- End to End delay is eliminated and also prevention is made on DOS attacks.
- The routing mechanism is light weight and not process and resource consuming.
- Elliptic curve Digital Signature Algorithm (ECDSA) is more recently standardized and supposedly reducing size of digital signatures and cryptographic keys.

3.3 Objective:

The objective of the protector is recognizing the individuals from the botnet, keeping in mind the end goal to boycott the bots, without denying the support of typical clients. The most straightforward, incorporated DoS assaults (e.g., TCP SYN flooding) misused vulnerabilities in the convention stack, depending basically on rehashed, high-rate transmissions of a similar demand from a solitary client. In such conditions, the strange transmission rate was adequate to distinguish the wellspring of the assault. Conversely, in a DDoS assault the individual bot's rate is kept direct, while the worldwide assaulting rate must be extensive. By the by, without facilitate complexity, the bargained machines can be as yet recognized at a solitary client level. Actually, movement examples of ordinary clients are normally described by a specific level of advancement (for example, as time slips by, unmistakable Ib-pages are probably going to be gone to), while the redundancy conspire verifiably demonstrates the odd bot character. This work centresaround an all the more difficult variation of DDoS assault, in particular, on the ongoing class of utilization layer DDoS assaults. This impossible to miss type of assaults goes past the least difficult reiteration based assaults, by abusing the plentiful scope of potential outcomes accessible at the application layer . In such novel assaults, the bots pick arbitrarily their solicitations from an arrangement of acceptable messages (a copying word reference), attempting so to mask their movement designs as typical ones.

4. METHODOLOGY

Statistical methods to identify DDoS attacks are proposed, relying on computing entropy and frequency-sorted distributions of selected packet attributes. The DDoS identification is then based on the detection of anomalies in the characteristics of the packet attributes. Description of the botnet identification algorithm, it is worth commenting on a possible limitation of the proposed approach.

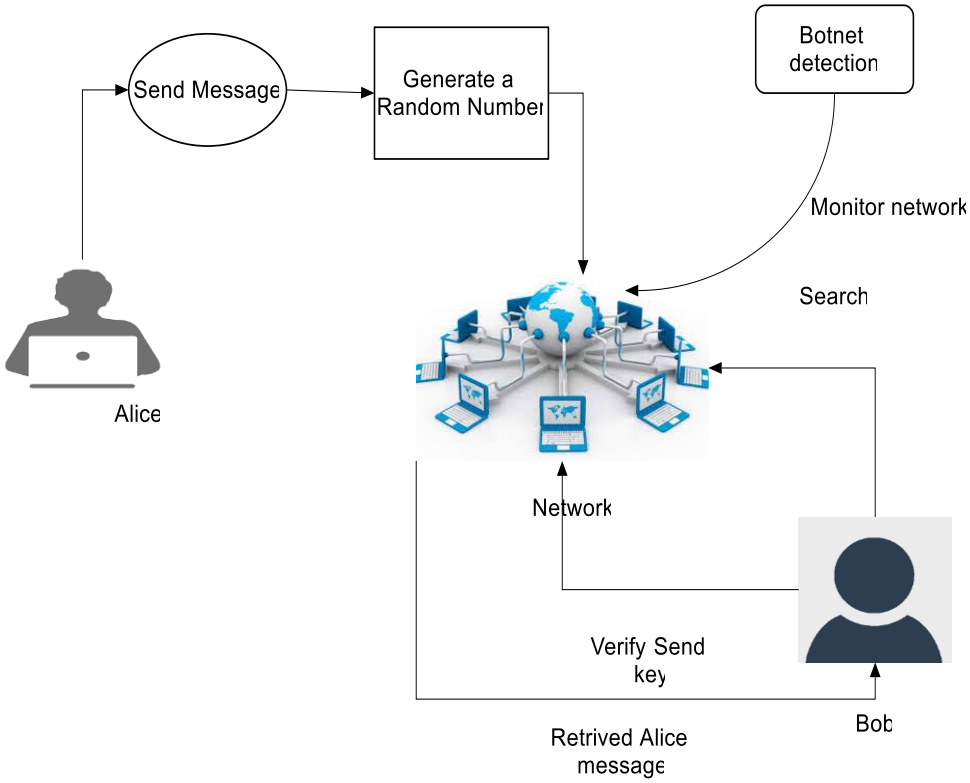

Figure 4.1 Proposed Methodology of Botnet Detection

I currently centre on the induction of the derivation algorithm gone for revealing a botnet conceivably covered up in the system.The BotBuster algorithm is portrayed by the pseudo-code detailed in the correct section above, and essentially misuses the way that, given two disjoint subnets, the BIC permits to segregate the circumstance where both subnets are a piece of a botnet, from the circumstance where no less than one of them is made of ordinary clients. I will demonstrate that the proposed algorithm has the essential necessity of consistency, specifically, the certification that the botnet is accurately distinguished as t develops.Give us a chance to analyze how the algorithm functions. To begin with, take note of that a botnet made of one client, other than having neither rhyme nor reason in hone, is by definition non-identifiable, since I accepted that the attributes of the messages at a solitary client level try not to uncover any unique data. Presently, toward the start of the algorithm, client 1 is at first proclaimed as a bot, to be specific, B ˆ = {1}.

At that point, it is checked whether clients 1 and 2 shape a botnet. Assuming this is the case, B ˆ = {1, 2} is taken as the current botnet appraise. If not, B ˆ = {1} is held. At that point, it is checked whether the presently assessed botnet B ˆ frames a bot with client 3, thus on. Toward the finish of the internal circle, the algorithm winds up with a gauge B ˆ. On the off chance that the cardinality of the assessed set is more prominent than one, it is taken as a present gauge. The method is then restarted by picking client 2 as introductory turn, and successively checking the rest of the clients as explained previously. Toward the finish of the inward circle, the algorithm closes up with another gauge B ˆ. On the off chance that the cardinality of the assessed set is more prominent than one and more prominent than the cardinality of the beforehand assessed set8, at that point it is taken as a present gauge. Something else, the past gauge is held. The methodology closes when the sum total of what clients have been filtered as turns. Elliptic Curve Digital Signature Algorithm is implemented over elliptic curve P-192 as mandated by ANSI X9.62 in C language. The Project includesessential modules for domain parameters generation, key production, signature production, and signature substantiation over the elliptic curve.

BotBuster algorithm 1

Step 1: Algorithm: B ˆ new=BotBuster

Step 2:N = {1, 2, . . . , N}; B ˆ new = ∅;

Step 3:for b0 ∈ N do

Step 4:B ˆ = {b0};

Step 5:for j ∈ N \ {b0} do

Step 6:if ρ ˆ(B ˆ ∪ {j}) < γ(B ˆ, {j}) then

Step 7:B ˆ = B ˆ S{j};

Step 8:end

Step 9:end

Step 10:if |B ˆ | > max(1, |B ˆ new|) then

Step 11:ˆ B new = B ˆ ;

Step 12:end

Step 13:end

Algorithm 2

ECDSA KEY GENERATION:

Acreature A's key pair is connected with a particular set of EC domain constraint D= (q, FR, a, b, G, n, h). E is an elliptic curvaturedefinitein excess ofFq , and P is a point of primary order n in E(Fq), q is a prime. Each entity A does the following:

1. Select a random integer d in the interval [1, n- 1].

2. Compute Q = dP.

3. A's public key is Q, A's private key is d.

ECDSA Signature Generation:

To sign a communication m, an article A with sphere of influenceconstraint D= (q, FR, a, b, G, n, h) does the following: 1. Select anaccidental or pseudorandom numeral k in the interval [1, n-1].

2. Compute kP =x1, y1 and r= x1 mod n (where x1 is look upon as an integer between 0 and q-1). If r= 0 then go back to step 1.

3. Compute k -1mod n.

4. Compute s= k -1 {h (m)+dr} mod n, where h is the Secure Hash Algorithm (SHA-1). If s = 0, then go back to step 1.

.

5. The signature for the message m is the pair of integers (r, s).

5. RESULT AND DISCUSSION

1. Network Traces Collection and DDoS Attack Generation

As regards the measuring stage that precedes the botnet identification algorithm, I adopt the following pipeline. Packets are preliminarily filtered by using a popular software package for packet capturing and network protocol analysis. At the output of such preliminary filtering stage: i) only the traffic directed to the destination that is being monitored is retained; ii) among the surviving packets, only the application layer traffic is retained; iii) the resulting packets are divided on the basis of their source IP address, and are finally fed to the botnet identification algorithm. A popular e-commerce Ibsite has been selected as target destination of the attack. Clearly, the normal users have no attacking intent, they perform ordinary surfing activity. About 20 min of (application-layer) traffic have been collected, from 10 independent users, which Ire students and researchers working in our laboratory, and carrying on their surfing activity almost independently. In order to help understanding the nature and significance of the dataset, I report that the total number of TCP flows is about 26800, the median of flows across users is 2846, the minimum number of flows is 1042, the maximum number of flows is 3925, and the average packet size is 776 bytes. Supported by these numbers, and by a trace-by-trace inspection, I conclude that the activity of the users during the monitored period is reasonably sustained, and compatible with typical traffic, meaning that the patterns are neither trivial (users effectively send requests) nor anomalous (users do not overload the destination with huge rates)

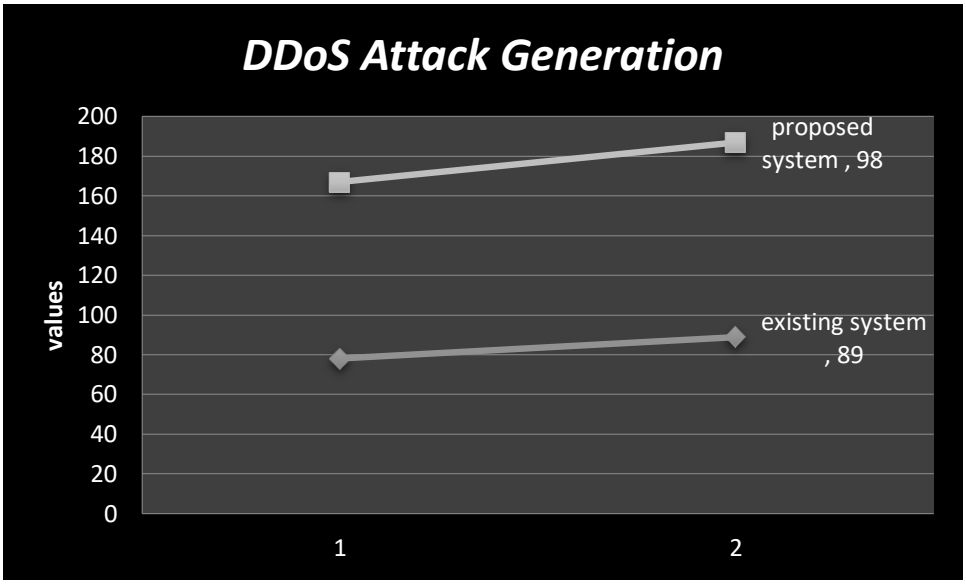

Figure 5.1 Performance Analysis of Existing and Proposed System

2. Bots and/or Spoofed Addresses

The setting considered in this work encompasses naturally the relevant scenario of spoofed source IP addresses, which is becoming rather common in DDoS attacks. In such scenario, each bot can change its source IP address by (randomly) choosing from a collection of spoofed addresses. In the randomized DDoS attack considered in this work, the bot traffic streams are constructed by picking subsequent messages independently from an emulation dictionary that is shared among all the bots. Accordingly, a botnet of B nodes employing a set of A randomly spoofed addresses (with A > B), is equivalent to a botnet of A nodes performing the attack. Since the goal of the network analyst is banning the machines that launch the attack (not associating a physical machine to its IP address), I conclude that the performed analysis applies directly to the case of spoofed IP addresses, provided that the number of bots is replaced by the number of IP addresses globally employed by the botnet. For the sake of brevity, such "effective" number will be still denoted by B.

VI. CONCLUSION

The Considered Distributed Denial of Service (DDoS) attacks launched by bots that are capable to learn the application layer interaction possibilities, so as to avoid repeating one simple operation many times. Such enhanced capability of the attacker makes it impossible to identify one of those many bots relying only on its individual activity patterns. The main contributions of this work are as follows: i) I introduced a formal model for the class of randomized DDoS attacks with increasing emulation dictionary; ii) I proposed an inference algorithm aimed at identifying the botnets executing such advanced DDoS attacks, and I ascertained consistency of the algorithm, namely, the property of revealing the true botnet as time elapses; iii) I evaluated the proposed methodologies on a tested environment. To give a snapshot of the performance delivered by the BotBuster algorithm: for a network with 100 normal users and 100 bots, 90% of the bots are correctly guessed in about a quarter of minute, while the fraction of normal users that are incorrectly banned is in practice zero.

There are many questions that remain open, and that might deserve further investigations. To mention a few: testing the algorithm over more datasets, in order to examine the impact on performance of the nature of the site under attack, and or the different behaviours of users surfing on the Ib; conducting a refined convergence analysis in order to characterize, from an analytical viewpoint, the time needed to reach a prescribed accuracy, and the dependence of such time upon the network/botnet size and other relevant system parameters; examining the problem from an adversarial perspective where the botnet-identification strategy and the kind of DDoS attack are jointly optimized by looking for equilibrium solutions that manage the attacker's and defender's conflicting requirements; generalizing the

theoretical analysis and tools to multi-clustered DDoS attacks, where several botnets (using different emulation dictionaries) launch simultaneously their attack.

REFERENCE:

[1] W. Stallings, *Cryptography and Network Security: Principles and Prac- tice*, 6th ed., Pearson, 2013.

[2] N. Hoque, D. Bhattacharyya, and J. Kalita, "Botnet in DDoS attacks: trends and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2242–2270, fourth quarter 2015.

[3] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proc. DARPA Information Survivability Conference and Exposition*, Washington, DC, USA, Apr. 2003, pp. 303–314.

[4] J. Yuan and K. Mills, "Monitoring the macroscopic effect of DDoS flooding attacks," *IEEE Trans. Depend. Secure Comput.*, vol. 2, no. 4, pp. 324–335, Oct. 2005.

[5] L. Li, J. Zhou, and N. Xiao, "DDoS attack detection algorithms based on entropy computing," in *Proc. ICICS 2007*, Zhengzhou, China, Dec. 2007, pp. 452–466.

[6] Y. Xiang, K. Li, and W. Zhou, "Low-rate DDoS attacks detection and traceback by using new information metrics," *IEEE Trans. Inf. Forensics and Security*, vol. 6, no. 2, pp. 426–437, Jun. 2011.

[7] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew DDoS," *IEEE Trans. Inf. Forensics and Security*, vol. 9, no. 7, pp. 1069–1083, Jul. 2014.

[8] "Layer 7 DDoS." http://blog.sucuri.net/2014/02/layer-7-ddos-blockinghttp- flood-attacks.html.

[9] "Taxonomy of DDoS attacks." http://www.riorey.com/types-of-ddosattacks/# attack-15.

[10] "Global DDoS threat landscape." https://www.incapsula.com/blog/ddosglobal-threat-landscape-report-q2-2015.html.

[11] Ruffing, N., Zhu, Y., Libertini, R., Guan, Y., and Bettati, R. (2016). Smartphone reconnaissance: Operating system identification. In 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), pages 1086–1091.

[12] Schrittwieser, S., Frühwirt, P., Kieseberg, P., Leithner, M., Mulazzani, M., Huber, M., and Iippl, E. R. (2012). Guess who's texting you? evaluating the security of smartphone messaging applications. In NDSS. Citeseer.

[13] Schultz, A. M. (2016). Protecting consumer viewing habits: Reflections on the video privacy protection act.

[14] Shi, Y. and Biswas, S. (2015). Detecting tunneled video streams using traffic analysis. In 2015 7th International Conference on Communication Systems and Networks (COMSNETS), pages 1–8.

[15] Song, D. X., Wagner, D., and Tian, X. (2001). Timing analysis of keystrokes and timing attacks on ssh. In USENIX Security Symposium, volume 2001.