Optimizing Agent Placement for Flow Reconstruction of DDoS Attacks

Ömer Demir
Dept. of Information Tech.
Turkish National Police
Ankara, Turkiye
omerdemirkos@gmail.com

Bilal Khan
Dept. of Math & Comp. Science
John Jay College, CUNY
New York, USA
bkhan@jjay.cuny.edu

Ghassen Ben Brahim
Computer Science Dept.
Prince Mohamed Univ.
Al-Khobar, Saudi Arabia
gbrahim@pmu.edu.sa

Ala Al-Fuqaha
Computer Science Dept.
Western Michigan Univ.
Kalamazoo, USA
ala.al-fuqaha@wmich.edu

***Abstract*—The Internet today continues to be vulnerable to distributed denial of service (DDoS) attacks. We consider the design of a scalable agent-based system for collecting information about the structure and dynamics of DDoS attacks. Our system requires placement of agents on inter-autonomous system (AS) links in the Internet. The agents implement a self-organizing and totally decentralized mechanism capable of reconstructing topological information about the spatial and temporal structure of attacks. The system is effective at recovering DDoS attack structure, even at moderate levels of deployment.**

In this paper, we demonstrate how careful placement of agents within the system can improve the system's effectiveness and provide better tradeoffs between system parameters and the quality of structural information the system generates. We introduced two agent placement algorithms for our agent-based DDoS system. The first attempts to maximize the percentage of attack flows detected, while the second tries to maximize the extent to which we are able to trace back detected flows to their sources. We show, somewhat surprisingly, these two objectives are concomitant. Placement of agents in a manner which optimizes in the first criterion tends also to optimize with respect to the second criterion, and vice versa. Both placement schemes show a marked improvement over a system in which agents are placed randomly, and thus provide a concrete design process by which to instrument a DDoS flow reconstruction system that is effective at recovering attack structure in large networks at moderate levels of deployment.

***Keywords*—DDoS, network traffic, flow reconstruction.**

I. Introduction

A denial of service (DoS) is the act of preventing service or shared resources (or services) from reaching legitimate users [10]. When a DoS attack is mounted from large numbers of distributed sources, it is termed a Distributed Denial-of-service (DDoS) attack. Arbor Network's identifies DDoS as the most critical type of attack faced by Internet Service Providers [1].

The Internet architecture itself presents obstacles to the resolution of the DoS/DDoS problem. First, network link resources are shared among all users, but there is no explicit enforcement of fair sharing. Second, core network components need to be simple so they can quickly deal with very high volumes of traffic. This in turn means they must do as little as possible per packet, so the core is unable to provide much security, implying that it is typically a service enforced at the edges. Lastly, interconnected autonomous systems are each managed by different authorities, and their heterogeneity makes widespread deployment of defenses difficult. A more detailed treatment of architectural features of the Internet and their implications for DDoS is given in [6]

Given the aforementioned inherent obstacles, the notion of "Solving the DDoS problem" has many interpretations. Here we focus on the problem of determining the true origins and mechanics of attacks. The source of attack packets is not easy to identify because the IP header's source address may be spoofed and network devices are not required to keep information about the path traveled by packets. We define *Flow Reconstruction* as "Actions taken to find the true sources and/or routes of packets to a given destination". There have been different approaches to this problem, including actively interacting with network traffic [5], [9], probabilistic and packet marking techniques [2], [12], and hash based logging [13]. Next, we give a brief synopsis of prominent examples of each of these approaches:

Active Interaction is a strategy of interfering with attack traffic to deduce information about attack sources based on the systemic reaction to the interference. Backscatter is the prototypical example of this technique [9], operating at the level of BGP level routers. Backscatter finds the point of entry of the attack packets into BGP-level Internet backbone by having a backscatter server announce itself as the destination for spoofed IPs being used as sources addresses in the attack. Then the destination network under attack is made unreachable by having the backscatter server originate a BGP route announcement message. Since attackers continue to send packets to the victim but the target is no longer reachable, the ingress routers reply with a "Destination unreachable" message to the source IP of the attack packets. This ICMP message gets delivered to the backscatter server, thus revealing the entry point of the attack packets into the backbone. Despite its originality, Backscatter approach requires a modification to the BGP protocol and suffers from a collateral effect where good traffic destined to the victim is being dropped at the BGP level.

Packet Marking relies on routers adding identification information to the packets that they forward, so as to reveal the path the packets have taken [2]. Marking every packet is not feasible because of packet processing overhead introduced by checksum recalculation. Probabilistic packet marking (PPM) circumvents this by having routers select which packets are to be marked randomly as they transit. Router information

978-1-4673-2480-9/13/$31.00 ©2013 IEEE

the M1-Greedy algorithm is used to place agents in different networks of different sizes. There are four curves in the graph, with each curve representing the results of the experiment for networks of a different size (100, 200, 400, and 800 respectively). Each of the curves individually shows similar characteristics. However, as the network size increases, we note that the entire curve shifts downward. At an agent density of 0.03 the E[M1] values for 100, 200, 400, and 800 node networks are 0.704, 0.614, 0.526, 0.454 respectively. As the number of nodes in the network doubles, the E[M1] value decreases approximately 13%.

The second graph shows how the E[M3] curve changes when the M1-Greedy algorithm is used to place agents in different networks of different sizes. once Again four curves in the graph represent the results of the experiment for networks of sizes 100, 200, 400, and 800 respectively. Each of the curves individually shows similar characteristics. Once again, as the network size increases, we note that the entire curve shifts downward.

The third graph is very similar to the first one: It shows how the E[M1] curve changes when M3-Greedy algorithm is used to place agents in different networks of different sizes. For the agent density of 0.03 the E[M1] values for 100, 200, 400, and 800 node networks are 0.696, 0.603, 0.510, 0.436 respectively. As the number of nodes in the network doubles, the E[M1] value decreases approximately 14%. The graph shows us that the M3-Greedy agent placement performs better as the net size gets bigger. Likewise, the fourth graph is similar to the second one: It shows how does the E[M3] curve behaves when M3-Greedy algorithm is used to place agents in different networks of different sizes. Once again the graph shows that as network size increases, the curve shifts downwards proving the scalability of the proposed algorithm.

VI. Conclusion and Future Work

We consider the design of a scalable agent-based system for collecting information about the structure and dynamics of DDoS attacks. The agents implement a self-organizing and totally decentralized mechanism capable of reconstructing topological information about the temporal and spatial structure of attacks.

We showed that our system is effective at recovering DDoS attack flow structure, even at moderate levels of agent deployment. We described two effective schemes for selecting the precise locations at which agents should be placed: M1-Greedy and M3-Greedy. We quantified the performance of these schemes and assessed their scalability. In experiment 1, we saw that the greedy algorithms always perform significantly better than random placement, and provide good flow reconstruction capabilities even at modest agent deployment densities. We also showed that the two optimization criteria (M1 and M3) are concomitant: optimizing one tends to optimize the other. In experiment 2 we saw that these conclusions were consistent across different Waxman networks of the same size. Finally, in experiment 3 we saw that the effectiveness of the schemes actually *improves* as network size increases.

Future work. Having established that the greedy algorithms proposed here are effective at determining the placement of agents for optimal DDoS attack flow reconstruction, we plan to use the proposed schemes to determine optimal placement of agents within the real Internet topology, under at various deployment level assumptions. For this purpose we intend to use the inter-AS connectivity database maintained by the CAIDA [11] project. Using the CAIDA topology we will assess the extent to which the proposed system can deliver effective DDoS flow reconstruction services to the Internet community. This will enable us to determine the necessary deployment level required in a real global system, and moreover, give us the precise inter-AS links on which agents should be placed in order to maximize attack flow interception and optimize traceback to attacking nodes.

References

[1] Arbor Networks, http://www.arbornetworks .com

[2] Bellovin, ICMP traceback messages, RFC draft, September http://tools. ietf.org/draft/draft-be llovin-itrace/draft-bellovin-itrace- 00.txt (2000)

[3] Bellovin, Cert advisory ca-1996-26, Cert Advisory, ‘http://www.cert. org/advisories/CA- 1996-26.html (1996)

[4] Bloom, B. H.: Space time trade-offs in hash coding with allowable errors, Commun. ACM, vol. 13, no. 7, pp. 422–426, (1970)

[5] Burch and Hal :Tracing anonymous packets to their approximate source, Proceedings of the 14th USENIX conference on System administration. Berkeley, CA, USA: USENIX Association, 319–328 (2000)

[6] Demir O.: A Survey of Network Denial of Service Attacks and Countermeasures. City University of New York, Computer Science Department. (2009)

[7] Demir, O., Khan, B. : An Agent-based Architecture for Flow Reconstruction of DDoS Attacks. Proceedings of Int. Communications Conference (ICC) 2010, Cape Town, South Africa, 23-27 (2010)

[8] Demir, O., Khan, B.: Quantifying Distributed System Stability through Simulation A Case Study of an Agent-based System for Flow Reconstruction of DDoS Attacks. In: Proceedings of the 1st Intelligent Systems, Modeling and Simulation Conference, Liverpool, England, 27-29 January (2010)

[9] Gemberling B., Morrow, C., and Greene, B.:ISP security-real world techniques. presentation, nanog. NANOG, www.nanog.org (2001)

[10] Gligor V.D.: A Note on Denial-of-Service in Operating Systems. IEEE Trans. Softw. Eng. 10, 320–324 (1984)

[11] Hyun, Y., Huffaker, B., Andersen, D., Aben, E., Luckie, M., Claffy K. C., and Shannon, C. The IPv4 Routed /24 AS Links Dataset - 11/15/2009 , http://www.caida.org/data/active/ipv 4_routed_topology_ aslinks_dataset.xml

[12] Savage, S, Wetherall, D. Karlin, A. and Anderson, T.: Practical network support for IP traceback, SIGCOMM Comput. Commun. Rev., vol. 30, no. 4, pp. 295–306, (2000)

[13] Snoeren, A. C.:Hash-based IP traceback, in SIGCOMM ’01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. New York, NY, USA: ACM, pp. 3–14, (2001)

[14] Waxman, B. M.: Routing of Multipoint Connections.: Broadband Switching: Architectures, Protocols, Design, and Analysis. IEEE Computer Society Press, Los Alamitos, CA, USA (1991)