Quantifying Distributed System Stability through Simulation: A Case Study of an Agent-Based System for Flow Reconstruction of DDoS Attacks

Omer Demir
Department of Computer Science
City University of New York Graduate Center
New York, U.S.A.
Email: odemir@gc.cuny.edu

Bilal Khan
Department of Mathematics and Computer Science
John Jay College
New York, U.S.A.
Email: bkhan@jjay.cuny.edu

***Abstract*—We investigate the stability properties of a novel agent-based system for the detection of network bandwidth-based distributed denial of service (DDoS) attacks. The proposed system provides a description of the structure of flows which comprise the DDoS attack. In doing so, it facilitates DDoS mitigation at or near attack traffic sources. The constituent agents within the system operate at the inter autonomous system (AS) level, comprising a distributed collection of IP-layer network taps which self-organize in response to attack flows. We formalize the notion of stability for the proposed system, and show how we can use simulation to identify regions of instability within the system's parameter space. We then modify our system design to circumvent the uncovered singularities, and demonstrate the efficacy and tradeoffs implicit in our redesigned system.**

***Keywords*-stability; simulation; agent-based; distributed denial of service; flow reconstruction.**

I. Introduction

Denial of service (DoS) occurs when legitimate users are prevented from getting access to shared resources or services. If DoS is originated from a large number of distributed attackers, the event is termed a Distributed Denial-of-service (DDoS) attack. DDoS attacks have been identified as the most urgent Internet security concern by Arbor Network's 2008 survey [1].

The roots of the DDoS problem lie in the design of the Internet architecture [2] itself: (1) In order to gain the most of the Internet, its network link resources are shared, but there is no enforcement of fair sharing; (2) The network core processes high volumes of traffic so core components can do very little processing per packet, thus all computations (e.g. those ensuring security) must be performed at the edge; (3) The Internet's constituent networks are managed by different authorities, and this heterogeneity makes widespread deployment of DDoS defense mechanisms difficult.

"Solving the DDoS problem" has many interpretations. The one we focus on here is the problem of finding the true sources and mechanics of attacks. Finding the source of attack packets is difficult because the source address field of IP header of packets is easy to forge or spoof. Moreover, current standards do not require network devices to maintain information about paths that packets take. We define Flow Reconstruction as "Actions taken to find the true source and route of packets". There have been different approaches to this problem, including actively interacting with network traffic [3], [4], probabilistic and packet marking techniques [5], [6], and hash based logging [7].

II. Related Work

Active Interaction is a strategy of interfering with attack traffic in order to deduce information about attack sources based on the systemic reaction to the interference. Backscatter is the prototypical example of this technique [4], operating at the level of BGP level routers. Backscatter finds the point of entry of the attack packets into BGP-level Internet backbone. While innovative, Backscatter has many drawbacks, most notably, a huge collateral effect by which legitimate traffic to the victim will also be blocked at the BGP level.

Packet Marking relies on routers adding identification information to the packets that they forward to that packets reveal the path they have taken [5]. Marking every packet is not feasible because of packet processing overhead introduced by checksum recalculation. Probabilistic packet marking (PPM) selects packets randomly and marks them with transit router address information (typically using the the IP identification field). The main limitation of packet marking is that the path convergence is slow. Park and Lee [8] showed that PPM is effective at localizing the attack origin in single-source attacks but that as the number of mounted attack sources increases, the traceback is rendered more difficult.

Hash-Based Traceback was introduced by Snoeren and Alex [7]. In their system the packets are stored in a space-efficient data structure called a Bloom filter [9] as follows: Periodically, a 2^n-sized bit-array is initialized to all zeroes. Whenever a packet arrives, k distinct n-bit packet digest functions are computed from the packet's immutable header fields; the array is marked at indices corresponding to these k values. When an IP Traceback request arrives, it specifies the time, and provides a copy of the packet to be traced. The

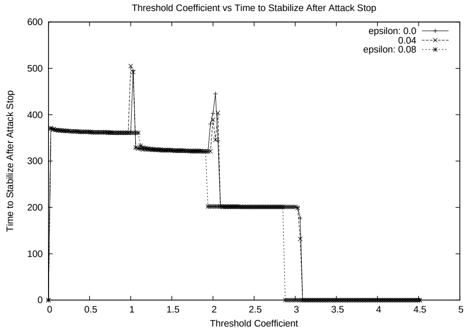

Figure 6. System convergence time at attack stop

and will be presented later.

The curve shows that the agent system gets stabilized in constant time,but that for certain values of ATC near integer values, it never stabilizes. For ATC sufficiently large, convergence is instantaneous since no attack is detected at such high thresholds.

Although it is expected that there is a decrease as the value of T increases, the timers within the agent FSMs force them to wait for certain amount of time to decide whether they are a parent (or not) before they change state to their final converged state. This results in not seeing a noticeable decrease in the value of CT during the attack.

Similarly, in Figure 6 we see the relationship between the CT and ATC, once the attack ceases. The $epsilon$=0 curve shows a stepping decrease as the value of T increases. This is because the number of agents involved in the attack decreases as the value of T increases, as describe above. The difference here is that there is no wait time; as soon as the traffic goes below T the agent changes it state.

In Figure 5 and Figure 6 the curves show peaks at certain points. The peaks goes to infinity for the first case and a finite high value in the latter setting. These peaks occur at integral values because the expected traffic and the value of the threshold are very close to each other and since we use a random process to model interpacket times the value of $X_v(t)$ fluctuates in the neighborhood of T, cauing AH and BH events and preventing state convergence.

Since this kind of behavior is undesired, we modified our system so it generates AH only when traffic $X_v(t)$ exceeds $T = ATC(W/\mu)(1 + epsilon)$ and generates BH only when $X_v(t)$ falls below $T = ATC(W/\mu)(1 - epsilon)$. After modification of the system we repeated the experiments for $epsilon$ values of 0.04 and 0.08.

In Figure 5 and Figure 6 we see that the spikes decreased as we used greater $epsilon$. This shows that the system can be made stable as a whole, over the entire parameter space by a minor modification of the local attack detection logic within its constituent agents.

X. Conclusion

We report on the use of simulation to characterize the performance and stability of a novel distributed system for DDoS flow reconstruction. Preliminary simulation results showed that in certain settings the system state was unstable. We were able to locate the parameter ranges with the help of simulations, and determine the cause of instability, based on which we modified our system design. Finally, using simulation once again, we verified that the augmented protocols exhibit both good performance and systemic stability.

Acknowledgments

The first author thanks the Turkish National Police for supporting his graduate studies at City University of New York, of which this research is a part.

References

[1] “Arbor networks worldwide infrastructure security report,” 2008. [Online]. Available: http://www.arbornetworks.com

[2] O. Demir, “A survey of network denial of service attacks and countermeasures,” City University of New York, Computer Science Department, Tech. Rep., 2009.

[3] Burch and Hal, “Tracing anonymous packets to their approximate source,” in *LISA ’00: Proceedings of the 14th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 2000, pp. 319–328.

[4] B. Gemberling, C. Morrow, and B. Greene, “ISP security-real world techniques. presentation, nanog.” NANOG, 2001. [Online]. Available: www.nanog.org

[5] Bellovin, “ICMP traceback messages,” RFC draft, September 2000. [Online]. Available: ‘http://tools.ietf.org/draft/draft-bellovin-itrace/draft-bellovin-itrace-00.txt

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

[7] Snoeren and 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, 2001, pp. 3–14.

[8] Bellovin, “Cert advisory ca-1996-26,” Cert Advisory, 1996. [Online]. Available: ‘http://www.cert.org/advisories/CA-1996-26.html

[9] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[10] G. H. Mealy, “A method to synthesizing sequential circuits.” *Bell Systems Technical Journal*, pp. 1045–1079, 1955.

[11] D. Wagner and D. Dean, “Intrusion detection via static analysis,” 2001.

[12] Z. Sun, D. He, L. Liang, and H. Cruickshank, “Internet qos and traffic modelling,” *IEE Proceedings - Software*, vol. 151, no. 5, pp. 248–255, 2004. [Online]. Available: http://link.aip.org/link/?IPS/151/248/1