

DDoS Experiment Methodology *

Alefiya Hussain, Stephen Schwab,
Roshan Thomas
SPARTA Inc

Sonia Fahmy Jelena Mirkovic
Purdue University University of Delaware

June 15 2006

1 Introduction

The main objectives of the EMIST DDoS group is to advance the state of the art in rigorous evaluation of distributed denial of service attack-defense scenarios in the Internet. Over the last three years, we have developed an evaluation methodology using a combination of simulation, emulation, modeling, and analysis techniques that allows independent comparison of different DDoS defense systems.

We have identified five high-level dimensions that the experimenter needs to carefully design in order to conduct an effective evaluation: (1) attack mechanism, (2) background traffic, (3) network topology (4) defense mechanism, (5) measurements and metrics. The methodology provides a sequence of well defined steps that guide the experimenter in defining and conducting the evaluation.

In this paper, we briefly discuss the current state of art in each of these five dimensions of attack-defense evaluation and provide references for in-depth information. Section 2 discusses the distribution and activities of hosts involved in a DDoS attack for both current and future attacks. Section 3 discusses legitimate traffic workload creation using various types of background traffic generators. Section 4 discusses topological characteristics of the Internet and how they impact DDoS attack-defense evaluation. Section 5 discusses various types of defense technologies that can be evaluated using the methodology framework and lastly Section 6 discusses the necessary and sufficient set of measurements and metrics for evalu-

ating the impact of attacks and the efficacy of the defense mechanisms. Additionally, each section also provides references to tools that can be used to automate various aspects of the evaluation methodology.

2 Attack Mechanisms

Denial of service attacks can deny legitimate service in two ways: (1) by consuming some critical resource in the network or at the end host via abundant or complex traffic, or (2) by exploiting some vulnerability within a router, an end host's operating system or an application to make a service inoperable. Attacks of the first type are frequently called *flooding* attacks while those of the second type are called *vulnerability* attacks. Experimenters can opt for using real attack tools, captured from the wild. Quite a few of these tools can be found at [33]. On the other hand, writing one's own packet flooding tool or using an existing tool written by security researchers can have significant advantages over real attack tools. Real packet flooders generate very simple flooding traffic — their primary sophistication lies in control mechanisms used to coordinate agent networks and to hide their presence from defenders. If an experimenter's goal is to test an attack-defense combination, rather than a security system that detects agent machines based on their coordination activity, researcher-written tools can simplify testing because they have many customizable parameters and can generate a wider variety of flooding attacks than real tools.

Because of the possibility of misuse researcher-developed tools cannot be freely downloaded but they can be obtained by contacting corresponding project leaders via E-mail addresses found on the DETER tools web

*This material is based on work partially supported by National Science Foundation under Cooperative Agreement No. ANI-0335298 with support from the Department of Homeland Security

page. The UCLA Laboratory for Advanced Systems Research (LASR) has developed the *Cleo* attack tool. This tool has a master-slave architecture. The slave code is installed at several clients and their IP addresses are specified in a configuration file used by the master file to start or stop the packet flood at the child nodes. Cleo is capable of generating various kinds of attacks such as constant rate attacks, pulsing attacks where the active period and inactive period can be specified, increasing rate attacks, and periodic increasing rate attacks. The tool also has options to specify the spoofing technique, set packet size, customize targeted ports, and it can generate TCP, UDP, ICMP traffic or combinations of the three. *MACE* is a versatile tool, developed by Wisconsin Advanced Internet Laboratory (WAIL), that can generate a variety of DoS and worm traffic scenarios. It provides a high-level language for attack traffic specification and contains a small, but easily extensible, database of attacks. The EMIST DDoS group has also incorporated an attack agent within the set of tools available on DETER. This agent can be scripted to perform a wide range of attacks within an automated test scenario.

Finally, some tools for network auditing can be used to generate packet floods. Packit tool [7] can generate traffic with many spoofed fields in TCP, UDP, ICMP, IP, ARP, RARP, and Ethernet header. Nmap tool [18] can generate a variety of packet floods, and some of probe packets generated by this tool can crash certain operating systems, thus recreating a vulnerability DoS attack.

3 Cross Traffic

Cross traffic modeling is an important step in evaluating a defense mechanism as different conclusions can be derived about the performance based on the composition of the cross traffic.

The simplest form of background traffic generation is using packet trace replay [44]. Many defense systems need to be tested under realistic traffic conditions at high data transmission rates. Replaying real packet traces from high-speed links using multiple PCs can allow the experimenter to stress the defense system under high traffic rates and evaluate performance.

Another approach is using *application-specific* traffic generators such as Surge [4], trafgen [12], PackMime [9].

They model network traffic based on different applications, such as a web browser or FTP. A combination of these traffic generators can be used to model an application mix on the network.

Some traffic generators are *application independent* and create traffic at the IP flow level. Examples include Harpoon [34] and D-ITG [2] that create network traffic based on probabilistic distributions and stochastic processes for various traffic parameters such as inter-packet gap interval and packet size.

Lastly some traffic generators support parametrization of traffic models from real network measurements, for example RAMP [22] and LTPProf [28].

The EMIST DDoS team has developed tools that allow configuring a wide mix of background traffic that consists of TCP traffic created using Harpoon [34], DNS traffic by setting up a server and periodically issuing requests from various locations in the topology, and ICMP echo request and reply traffic using the ping utility.

4 Topology

DDoS attacks may target routers/links or services in the network, and traffic from multiple attackers may be aggregated within the network. Hence, topology is an extremely important dimension in DDoS testing. Selecting benchmark topologies with realistic routing parameters and representative resources and services is an extremely challenging problem [1]. Internet topology characterization has been the subject of significant research for over a decade [45, 16, 8, 10, 19, 41]. Several researchers have examined Internet connectivity data at both the Autonomous System (AS) level and at the router level, and characterized the topologies according to a number of key metrics. A well-studied metric is the degree distribution of nodes in a topology, especially at the Autonomous System level, which was found to be heavy-tailed – a phenomenon typically referred to as “the power law phenomenon” [16, 8, 10]. Clustering characteristics of the nodes have also been examined, and the term “the small world phenomenon” [40, 19, 3] was used to denote preference to local connectivity. Recent work [23] uses joint degree distributions to capture different metrics such as clustering, assortativity, rich club connectivity, distance, spectrum, coreness, and betweenness.

One of the earliest and most popular topology generators is GT-ITM [45], which used a hierarchical structure of transit and stub domains. GT-ITM and other structural topology generators are believed to generate representative topologies when the number of nodes in the topology is small [37]. In fact, a key problem with selecting benchmark topologies is the scale-down of a topology of several thousand or even millions of nodes to a few hundred nodes (which is the number of nodes available on a testbed like DETER).

Routers within a domain typically use a routing protocol such as Open Shortest Path First (OSPF) or IS-IS. Configuring *border* routers in a topology to run the Border Gateway Protocol (BGP) poses a significant challenge, since Internet Service Providers (ISPs) use complex BGP policies for traffic engineering. The work by Gao et al. [17, 39] infers AS relations and this information can be used to configure BGP routers. Further information on other topology generation and routing configuration tools we have developed for DETER can be found in [11].

Assigning link delays and link bandwidths is non-trivial, since delay and bandwidth data, especially within an enterprise network, is not public, and is sometimes impossible to infer. Tools such as [15, 35, 25] have been proposed to measure *end-to-end* bottleneck link capacity, available bandwidth, and loss characteristics. Standard tools such as ping and traceroute can give end-to-end delay or *link delay* information, when their probe packets are not dropped by firewalls. Identifying *link bandwidths* is perhaps the most challenging problem. Therefore, an experimenter usually resorts to using information about typical link speeds (optical links, Ethernet, T1/T3, DSL, cable modem, dial up, etc) to assign link bandwidths in benchmark topologies.

5 Defense Mechanisms

A large number of DDoS defense systems have been proposed in recent years. Because DDoS is a multifaceted threat, proposed defenses vary greatly in their approaches to a defense. Some systems aim only at detecting attacks, others attempt to also filter attack traffic, while protecting legitimate user's traffic. Some systems also attempt to locate attack sources. Finally, there are systems that prevent

certain types of DDoS attacks by modifying underlying communication protocols.

To thoroughly evaluate a defense, one must be aware of its approaches to attack detection, response, prevention or traceback, and stress test them by generating attacks that attempt to bypass or crash the defense. Below we list recommended test scenarios for some general defense categories, defined in [27]:

- Defenses that train behavioral models to learn the difference between the legitimate traffic and the attack (e.g., [26]) should be tested with flooding attacks that mimic legitimate traffic features and slowly increase their rate to achieve values that deny service.
- Defenses that use resource accounting should be tested with highly distributed attacks, where each attacker sends at a low rate.
- Defenses that use resource multiplication should be tested with highly distributed attacks, generating high-rate traffic that challenges resource replication.
- Interdependent defenses should be challenged with attacks on the defense itself, and in presence of control message loss, to evaluate whether defense modules can function when isolated from their peers.
- Defenses that perform agent identification (such as traceback [5, 31]) should be tested in topologies that have high levels of path-sharing between legitimate users and attackers, and with highly distributed attacks where each agent floods at a low packet rate. This setup challenges a defense to precisely separate the legitimate from the attack traffic, and also tests its scalability.
- Defenses that detect attacks and respond to them in some fashion should be tested with short-duration, repetitive attacks to evaluate the cost of turning the defense on and off and the overall protection offered to the attack victim.
- Defenses that deploy some kind of a cooperative defense (e.g., [21, 43]) should be tested for insider attacks to evaluate the damage that a trusted member could inflict to a system, if compromised by an attacker.

6 Metrics

One of the challenges in addressing the measurements and metrics dimension in our evaluation methodology is the lack of a standard set of metrics that can be used to evaluate a mix of DDoS attacks and defenses in various experiments. Our review of the literature indicated that individual research efforts and commercial products utilized a variety of metrics to measure and assess the results of their respective techniques, products and technologies. Also, interesting is that most of the metrics are not specifically DDoS-centric; rather, they are straightforward applications of well-known metrics used by researchers and practitioners in networking, performance, and quality of service evaluations.

In developing our methodology, we wanted to get better insights into how these metrics can be applied to a broad set of DDoS experimental settings that utilize multiple attack types, defenses, topologies and background traffic as well as how they can be used as a basis for the development of more DDoS-centric measures. To enable this, we developed a high level framework for analyzing and categorizing network and system performance metrics [32]. First, this framework divides all metrics into two broad categories, namely *extrinsic* and *intrinsic*. Extrinsic metrics are measures that can be computed and observed by external parties in relation to the object (attack, defense etc.) being measured. On the other hand, intrinsic metrics can only be computed by the object being measured and only by analyzing the internal algorithms and data structures such as queues and connection tables. Given the distinction between extrinsic and intrinsic metrics, we can further categorize an individual extrinsic or intrinsic metric in two dimensions that reflect the granularity, relevance and application. The first dimension is the topological granularity that the metric applies to, namely end-point (including client and server side), link-level or end-to-end. The second dimension is the layer in the protocol stack, starting at the bottom of the stack with packet-level metrics and moving progressively up the stack to flow-level (such as TCP) and aggregate-level and application-level metrics.

Given the above two-dimensional characterization of extrinsic and intrinsic metrics, we review various metrics in three areas, namely, characterizing traffic, assessing attack impact and assessing DDoS defense effectiveness

and point out where they have been used in the DDoS literature for research and experiments. Unfortunately, space constraints prevent us from discussing the context and details of the research efforts where these metrics were used. We identify each metric as extrinsic or intrinsic by the notation (E) and (I), respectively.

Metrics for characterizing traffic: These are well known in the networking community. Examples of end-point packet-level metrics observable at client side include server response rate (E), average response-time (E), server-error rate (E) and those at the server-side include per client packet rate (E), packet-drop-rate (I), per packet overhead (I), etc. Link and end-to-end packet metrics at the packet-level include link and end-to-end throughput, error rates and latencies. In the context of DDoS attacks, such packet level traffic characterization metrics are used to characterize attack traffic in terms of their attack rate (intensity) and attack duration [29] as well as goodput (the throughput of legitimate traffic). Moving up to the flow-level (i.e. relevant to connections such as that in TCP), client-observable end-point metrics include average connection establishment time (E) and server connection completion rates (E) while those observable at the server-side include the per client connection request rate [20] (I) and per-client goodput (I).

Link-level flow metrics include the per flow or per-connection throughput observed at the link. Examples of end-to-end flow metrics include those used by protocols like TCP for flow and congestion control such as throughput, round trip time (RTT) for a connection (E), per connection retransmission timeout value (I), per connection loss rate (I). End-point aggregate-level metrics observable at the server-side include per aggregate arrival rate (I) and aggregate service rate (I) [24]. Collectively, we refer to these traffic metrics as *base metrics* as we expect these to be leveraged and composed to form more meaningful higher-level composite DDoS-centric metrics. An example of such a higher level metric is the probability of denied service proposed in [6]. It is based on ratios of the number of initiated, established, and completed TCP connections. Also, traffic characterization metrics at the application level are very closely tied application-specific semantics and perceptions of quality of service. Thus for streaming video an extrinsic metric is the mean opinion score (MOS) computed by asking end users their opinions on the video quality for Voice over IP (VoIP), met-

rics include round-trip-delay between VoIP endpoints(E), percentage of Packets discarded by the jitter buffer (I), mean-length-of-bursts (I) [13] etc.

Metrics for assessing attack impact: A first step in measuring the impact of an attack is to assess how various base metrics presented above degrade over the course of the attack. Attacks typically become noticeable when metrics such as goodput and server response times degrade beyond what is expected from routine fluctuations. The actual degradation may be measured and presented in various forms, including percentage drops in values and various statistical measures. The long term goal would be to use degradation in base metrics to develop higher-level attack impact assessments in terms that the end users perceive at the application layer.

Metrics for assessing defense effectiveness: Minimally, metrics for assessing the effectiveness of a DDoS defense must measure the accuracy and efficiency of a defense. A common measure of accuracy is the rate and probabilities of false positives and false negatives in attack detection. For example, this metric is used in works such as [14] and [30] to get a better sense of the accuracy of attack filtering algorithms. Another metric tied to the accuracy of a DDoS defense is the “probability of detection” [42]. Other metrics that could be used to assess the effectiveness of a defense include those that can be used to characterize some percentage improvement in one or more base and composite metrics. For example one could devise a metric that measures the time taken to achieve a 50 percent increase in goodput when a defense is turned on. An example of this that measures improvements in TCP throughput is discussed in section 5 of [36].

Another aspect of measuring the effectiveness of a defense is to formulate metrics that will pinpoint the exact breaking point of a defense. An example of this would be the maximum attack rate that a DDoS filtering scheme could handle without unacceptable packet loss of legitimate traffic. An example of this is discussed in [38] where legitimacy tests were used to filter incoming packets on a Gigabit link using an Intel IXP2400 network processor and it was discovered that the maximum sustainable attack rate was around 140Mbps due to the overhead involved in administering the legitimacy tests at the packet level. Thus, although the network processor device was capable of processing legitimate traffic close to the line speed of 1 Gbps, its breaking point for attack traf-

fic was 140 Mbps.

7 Conclusions and Future Work

We briefly discussed the five dimensions that are important to consider when setting up a DDoS attack-defense evaluation. The EMIST DDoS group has developed several tools in each area to aid in automation and ensure the fidelity of the experiment. Additionally, we have also exercised the methodology outlined in this paper and applied it to compare the performance of three defense systems.

However, there are a number of additional areas we need to address in the future. We aim to semi- or totally-automate the reuse of existing software and tools to create a DDoS experiment scenario allowing the experimenter to rapidly test systems. This work is on-going within the framework of the DETER security experimenter’s workbench. Additionally we will archive several experiment descriptions along with data and results to seed the process, and expand the DETER experiment archive as additional experimenters make use of the facility to study other defensive technologies and attack scenarios.

References

- [1] K. Anagnostakis, M. Greenwald, and R. Ryger. On the sensitivity of network simulation to topology. In *Proc. of MASCOTS*, 2002.
- [2] Stefano Avallone, Antonio Pescape, and Giorgio Ventre. Distributed Internet Traffic Generator D-ITG: Analysis and Experimentation over heterogeneous networks. In *Proceedings of ICNP*, 2003.
- [3] A. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.
- [4] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *Measurement and Modeling of Computer Systems*, pages 151–160, 1998.
- [5] S. Bellovin, M. Leech, and T. Taylor. ICMP Traceback Messages. *Internet draft, work in progress*, October 2001.

- [6] W. Blackert, A. Castner, D. Gregg, R. Hom, R. Jokerts, and E. Kyle. Distributed Denial of Service Defense Attack Tradeoff Analysis, Final Report. In *Johns Hopkins University Applied Physics Lab, Technical Report VS-03-073*, 2003.
- [7] Darren Bounds. Packit network injection and capture. "<http://www.obtuse.net/software/packit/>".
- [8] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *Proc. of IEEE INFOCOM*, June 2002.
- [9] J. Cao, W. Cleveland, Y. Gao, K. Jeffay, F.D Smith, and M. Weigle. Stochastic Models for Generating Synthetic HTTP Source Traffic. In *IEEE Infocom*, 2004.
- [10] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The Origin of Power Laws in Internet Topologies Revisited. In *Proc. of IEEE INFOCOM*, June 2002.
- [11] Roman Chertov, Sonia Fahmy, Pankaj Kumar, David Bettis, Abdallah Khreishah, and Ness B. Shroff. Topology generation, instrumentation, and experimental control tools for emulation testbeds. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation*, 2006.
- [12] Rigoberto Chinchilla, John Hoag, David Koonce, Hans Kruse, Shawn Ostermann, and Yufie Wang. Characterization of internet traffic and user classification: Foundations for the next generation of network emulation. In *Proceedings of the 10th International Conference on Telecommunication Systems, Modeling and Analysis*, 2002.
- [13] A. Clark. Common VoIP Metrics. In *Proc. of Workshop on End-to-End Quality of Service*, Geneva, October 2003.
- [14] M. Collins and M. Reiter. An Empirical Analysis of Target-Resident DoS Filters. In *Proc. Of the IEEE Symposium on Security and Privacy*, 2004.
- [15] C. Dovrolis and P. Ramanathan. Packet dispersion techniques and capacity estimation. *IEEE/ACM Transactions on Networking*, December 2004.
- [16] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proc. of ACM SIGCOMM*, pages 251–262, August 1999.
- [17] L. Gao. On inferring autonomous system relationships in the internet. In *Proc. IEEE Global Internet Symposium*, November 2000.
- [18] InSecure.org. *nmap security scanner*. Available at <http://www.insecure.org/>.
- [19] S. Jin and A. Bestavros. Small-world Characteristics of Internet Topologies and Multicast Scaling. In *Proc. of IEEE/ACM MASCOTS*, 2003.
- [20] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In *Proc. of 11th International World Wide Web Conference*, pages 252–262, 2002.
- [21] A. D. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *Proceedings of SIGCOMM*, 2002.
- [22] Kun-chan Lan and John Heidemann. RAMP: A tool for RAPid Model Parameterization and its applications. In *MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, pages 76–86, New York, NY, USA, 2003. ACM Press.
- [23] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat. The internet AS-level topology: Three data sources and one definitive metric. Technical report, University of California, San Deigo, 2005. Short version appears in ACM CCR, January 2006.
- [24] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. In *Computer Communications Review*, pages 62–73, 2002.
- [25] R. Mahajan, N. Spring, David Wetherall, and Thomas Anderson. User-level internet path diagnosis. In *Proceedings of ACM SOSP*, October 2003.

- [26] Mazu Networks. *Mazu Technical White Papers*. http://www.mazunetworks.com/white_papers/.
- [27] J. Mirkovic and P. Reiher. A Taxonomy of DDoS Attacks and Defense Mechanisms. *ACM Computer Communication Review*, 32(2):39–54, April 2004.
- [28] Jelena Mirkovic. D-WARD: Source-End Defense Against Distributed Denial-of-Service Attacks. Ph.D Thesis, 2003.
- [29] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring internet Denial-of-Service activity. In *Usenix Security Symposium*, pages 9–22, 2001.
- [30] T. Peng, C. Leckie, and K. Ramamohanarao. Protection from Distributed Denial of Service Attack Using History-based IP Filtering. In *Proceedings of the IEEE International Conference on Communications*, Anchorage, Alaska, May 2003.
- [31] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *ACM SIGCOMM Conference*, August 2000.
- [32] S. Schwab, B. Wilson, and R. Thomas. Methodologies and Metrics for the Testing and Analysis of Distributed Denial of Service Attacks and Defenses. In *IEEE MILCOM*, 2005.
- [33] Packetstorm Security. Distributed denial of service tools. "<http://www.packetstormsecurity.org/distributed/>".
- [34] Joel Sommers, Hyunshuk Kim, and Paul Barford. HARPOON:A Flow-Level Traffic Generator for Router and Network Tests. In *ACM SIGMETRICS*, June 2004.
- [35] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *ACM Internet Measurement Conference*, October 2003.
- [36] Haibin Sun, John Lui, and David Yau. Defending Against Low-Rate TCP Attacks: Dynamic Detection and Protection. In *12th IEEE International Conference on Network Protocols (ICNP)*, pages 196–205, 2004.
- [37] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *Proceedings of ACM SIGCOMM*, 2002.
- [38] R. K. Thomas, B. Mark, T. Johnson, and J. Croall. High-speed Legitimacy-based DDoS Packet Filtering with Network Processors: A Case Study and Implementation on the Intel IXP 1200. In *Network Processor Design: Issues and Practices, Volume 2*, July 2003.
- [39] F. Wang and L. Gao. On inferring and characterizing Internet routing policies. In *ACM Internet Measurement Conference*, October 2003.
- [40] D. Watts and S. Strogatz. Collective Dynamics of Small-world Networks. *Nature*, 363:202–204, 1998.
- [41] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, Univ. of Michigan, 2002.
- [42] Y. Xu and R. Guerin. On the robustness of router-based denial-of-service DoS defense systems. In *ACM Computer Communication Review*, pages 47–60, July 2005.
- [43] X. Yang, D. Wetherall, and T. Anderson. A DoS-limiting network architecture. In *ACM SIGCOMM Conference*, 2005.
- [44] Tao Ye, Darryl Veitch, Gianluca Iannaccone, and Supratik Bhattacharyya. Divide and conquer: PC-Based packet trace replay at OC-48 speeds. *TRIDENTCOM*, 00:262–271, 2005.
- [45] E. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proc. of IEEE INFOCOM*, volume 2, pages 594–602, March 1996.