

Benchmarks for DDoS Defense Evaluation

Jelena Mirkovic, Erinc Arikan and Songjie Wei
University of Delaware
Newark, DE

Roshan Thomas
SPARTA, Inc.
Centreville, VA

Sonia Fahmy
Purdue University
West Lafayette, IN

Peter Reiher
University of California Los Angeles
Los Angeles, CA

Abstract—This paper addresses the critical need for a common evaluation methodology for distributed denial-of-service (DDoS) defenses. Our work on developing this methodology consists of: (i) a benchmark suite defining the necessary elements of DDoS attack scenarios needed to recreate them in a testbed setting, (ii) a set of performance metrics for defense systems, and (iii) a specification of a testing methodology that provides guidelines on using benchmarks and summarizing and interpreting performance measures. We characterize the basic elements of a typical DDoS attack scenario and describe how to embody those elements in a benchmark. We describe a set of automated tools we developed to harvest real data on attacks, legitimate traffic, and real network topologies. This data guides our benchmark design. We also describe the major difficulties in achieving realism in the various elements of DDoS defense evaluation in a testbed setting.

I. INTRODUCTION

Distributed denial-of-service (DDoS) attacks are a serious threat for the Internet’s stability and reliability. DDoS attacks have gained importance because the attackers are becoming more sophisticated and organized, and because several high-profile attacks targeted prominent Internet sites [12], [20]. To evaluate the many defenses that have been proposed against DDoS, it is necessary to develop an objective, comprehensive and common evaluation platform for testing them.

In this paper we describe our ongoing work on the development of a common evaluation methodology for DDoS defenses. This methodology consists of three components: (1) a *benchmark suite*, defining all the necessary elements needed to recreate a comprehensive set of DDoS attack scenarios in a testbed setting, (2) a set of *performance metrics* for defense systems, and (3) a specification of a *testing methodology* that provides guidelines on using benchmarks and summarizing and interpreting performance measures. Our methodology is specifically designed for use in the DETER testbed [2].

II. DDoS DEFENSE BENCHMARKS

DDoS defense benchmarks must specify all elements of an attack scenario that influence its impact and a defense’s effectiveness. We consider these elements in three dimensions:

- *DDoS attack* — features describing a malicious packet mix arriving at the victim, and the nature, distribution and activities of machines involved in the attack.
- *Legitimate traffic* — features describing a legitimate packet mix and the communication patterns in the target network. During the attack, legitimate and attack traffic compete for limited resources. The legitimate traffic’s features determine how much it will be affected by this competition.

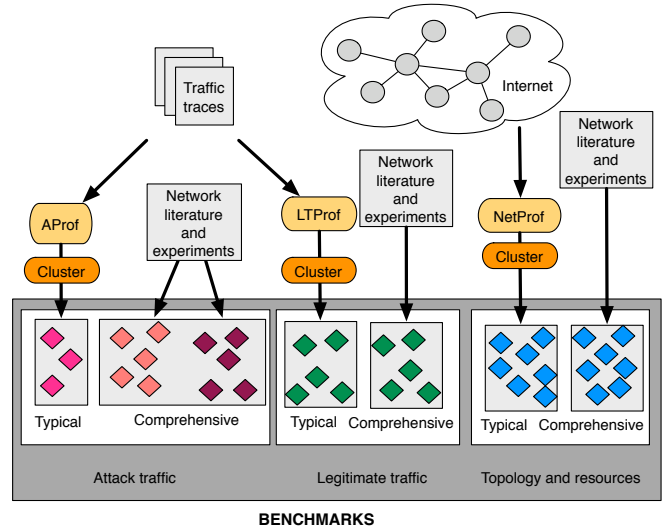


Fig. 1. Benchmark components and their generation

- *Network topology and resources* — features describing the target network architecture. These features identify weak spots that may be targeted by a DDoS attack and include network topology and resource distribution. In addition to this, performance of some defenses will depend on the topology chosen for their evaluation.

The basic benchmark suite will contain a collection of *typical* attack scenarios, specifying typical settings for all three benchmark dimensions. We harvest these settings from the Internet, using automated tools. The *AProf* tool collects attack samples from publicly available traffic traces. The *LTPProf* tool collects legitimate traffic samples from public traces. The topology/resource samples are collected and clustered by the *NetProf* tool, which harvests router-level topology information from the Internet and uses the *nmap* tool to detect services within chosen networks.

The typical suite provides tests that recreate attack scenarios seen in *today’s* networks. To facilitate in-depth understanding of a defense’s capabilities, the benchmark will also contain a *comprehensive* suite, which will define a set of traffic and topology features that influence the attack impact or the defense’s performance, and a range in which these features should be varied in tests. Instead of performing an exhaustive testing in this multi-dimensional space, our work focuses on understanding the interaction of each select feature with an attack and a defense. Figure 1 illustrates the benchmark’s components.

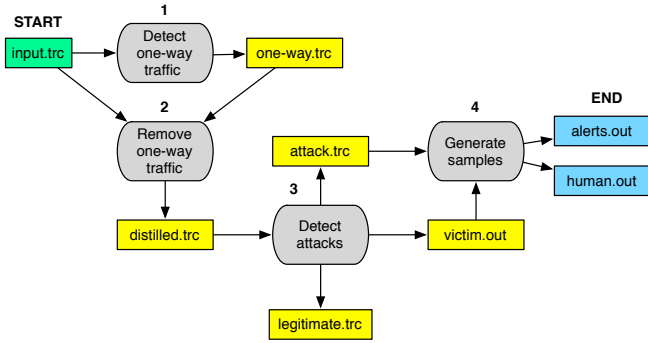


Fig. 2. Attack sample generation with AProf

III. ATTACK TRAFFIC

The attack traffic dimension specifies the attack scenarios observed in today’s incidents and hypothetical scenarios, designed by security researchers, that may become popular in the future.

A. Typical attack scenarios

Typical attack scenarios are obtained by building the *AProf* automatic toolkit to harvest attack information from public traffic traces stored in *libpcap* format. They detect attacks in the trace, separate legitimate from the attack traffic, and create attack samples that describe important attack features such as strength, number of sources, etc. Finally, attack samples are clustered to yield representative attack categories.

Attack samples are generated in four steps, shown in Figure 2:

- 1) *One-way traffic removal*. One-way traffic is collected if there is an asymmetric route between two hosts and the trace collection occurs only on one part of this route. Some of our attack detection tests use the absence of reverse direction traffic as an indication that the destination may be overwhelmed by a DDoS attack. One-way traffic, if left in the trace, would naturally trigger a lot of false positives. We identify hosts on asymmetric routes by recognizing one-way TCP traffic, performing some legitimacy tests on this traffic to ensure that it is not part of the attack, and recording its end points. We then remove from the original trace all packets between hosts on asymmetric routes.
- 2) *Attack detection* is performed by collecting traffic information at two granularities: for each connection (traffic between two IP addresses and two port numbers) and for each destination IP address observed in a trace. A packet belonging to a specific connection or going to a given destination is identified as malicious or legitimate using the detection criteria associated with: (1) this packet’s header, (2) this packet’s connection and (3) the features of the attack, which was detected based on the packet’s destination. We currently perform several checks to identify attack traffic, including examination of TCP characteristics, matching of application-level UDP and TCP traffic, detection of high-rate ICMP traffic, and several others. Space does not permit detailing of these techniques here.

Each packet is classified as legitimate or attack as soon as it is read from the trace. Packets that pass all detection steps without raising an alarm are considered legitimate. We store attack packets in *attack.trc* and we store legitimate

packets in *legitimate.trc*. Each attack packet is also used to update the information about the attack features (rate, type, spoofing, etc.). When a new attack is detected, this information is written to a file called *victim.out*.

- 3) *Attack sample generation*. Attack features are selected from the *attack.trc* file by first pairing each attack trace with alerts from *victim.out*. and then extracting attack characteristics from the attack trace. This step produces two output files: *human.out*, with the alert and traffic information in a human readable format and *alerts.out*, with the alerts only, specifying attack details such as rate, level of spoofing, attack type, number of attack sources, attack packet size and port distribution, etc.

Although it is too early to offer conclusions about typical attack scenarios, our preliminary results indicate that an overwhelming majority of attacks are TCP SYN attacks, sent at a low rate (2-5 packets per second) from many machines, and lasting from several minutes to several hours.

B. Comprehensive attack scenarios

We are applying three approaches to build comprehensive attack scenarios: (1) We use network literature to identify attacks that are particularly harmful to certain proposed defenses, (2) We use network literature and experiments to identify attacks that target critical network services, and (3) We investigate the link between the attack features (rate, packet mix, dynamics, etc.) and the attack impact, for a given test setting (network, traffic and defense), to identify relevant features and their test values.

IV. LEGITIMATE TRAFFIC

Legitimate traffic is specified in our benchmarks by host models that describe a host’s sending behavior. We build host models by automatically creating host profiles from public traffic traces and clustering these profiles based on their feature similarity to generate representative models, using the *LTPProf* tool we developed. For the comprehensive suite, we use network literature and tests to investigate how legitimate traffic features determine an attack’s impact and effectiveness of various defense systems.

We extract features for host profiles from packet header information, which is available in public traffic traces. Each host is identified by its IP address. Selected features include open services on a host, TTL values in a host’s packets, an average number of connections and their rate and duration. We also profile several of the most recent TCP and UDP communications and use the Dice similarity of these communications as one of the host’s features. This feature reflects the diversity of all the communications initiated by a host. We cluster host profiles using their feature similarity to derive typical host models.

Our preliminary results for legitimate traffic models are from the Auckland-VIII data set from NLANR-PMA traffic archive. This data set was captured in December 2003 at the link between the University of Auckland and the rest of the Internet. After filtering out little-used hosts, we have 62,187 host profiles left for clustering. The data is random-anonymized, so we could not identify inside vs. outside hosts. Thus, the resulting models characterize both the incoming and the outgoing traffic of the University of Auckland’s network.

We first identify four distinct host categories: (1) NAT boxes, with very diverse TTL values that cannot be attributed to routing changes, (2) scanners, which only generate scan traffic, (3) servers, which have some service port open; we differentiate between DNS, SMTP and Web servers, and (4) clients, which have no open ports and initiate a consistent volume of daily communications with others. We then apply clustering within each host category. The Table I shows the clustering result, illustrating that clustering generates several compact and large clusters in each category, that contain the majority of hosts.

TABLE I
LEGITIMATE HOST CATEGORIES

Host category	Hosts	All clusters	Top clusters
DNS servers	44%	62	Top 6 clusters contain 96% of hosts
SMTP servers	6.4%	65	Top 8 clusters contain 88% of hosts
Web servers	4.4%	85	Top 6 clusters contain 74% of hosts
Clients	28%	27	Top 6 clusters contain 90% of hosts
NAT boxes	9%	94	Top 7 clusters contain 67% of hosts
Scanners	5%	9	Top 5 clusters contain 99% of hosts

V. TOPOLOGY AND RESOURCES

To reproduce multiple-AS topologies, at the router level, we are developing a *NetTopology* tool similar to *RocketFuel* [22]. *NetTopology* relies on invoking *traceroute* commands from different servers [24], performing alias resolution, and inferring several routing and geographical properties.

For DETER, we have developed two additional tool suites: (i) *RocketFuel-to-ns*, which converts topologies generated by *NetTopology* tool or *Rocketfuel* to DETER-compliant configuration scripts, and (ii) *RouterConfig*, which takes a topology input and produces router (software or hardware) BGP and OSPF configuration scripts according to routers' relationships in the specified topology. We apply the methods of Gao et al. [9], [25] to infer AS relationships and use that information to generate configuration files for BGP routers. Jointly, *NetTopology*, *RocketFuel-to-ns* and *RouterConfig* tools form the *NetProf* toolkit.

A major challenge in reproducing realistic Internet-scale topologies in a testbed setting is scaling down a topology of thousands or millions of nodes to a few hundred nodes (the number of nodes available on a testbed like DETER [2]), while retaining important topology characteristics. *RocketFuel-to-ns* allows a user to specify a set of Autonomous Systems, or to perform breadth-first traversal of the topology graph from a specified point, with specified degree bounds and number of nodes bound. This enables the user to select smaller portions of very large topologies for testbed experimentation. The *RouterConfig* tool works both on (a) topologies based on real Internet data, and on (b) topologies generated from the GT-ITM topology generator [29]. One major focus of our future research lies in defining how to properly scale down DDoS experiments, including the topology dimension.

Another challenge in defining realistic topologies lies in assigning realistic link delays and link bandwidths. Tools such as [16], [6], [23], [18] have been proposed to measure *end-to-end* such characteristics, and standard tools like ping and traceroute can produce end-to-end delay or *link delay* information. Identifying *link bandwidths* is perhaps the most challenging problem. Therefore, we use published information about typical link speeds [26] to assign link bandwidths in our benchmark topologies.

For localized defense testing, it is critical to characterize enterprise network topologies and service. We analyzed enterprise network design methodologies typically used in the commercial marketplace, such as Cisco's classic three-layer model of hierarchical network design [21], [27]. Our analysis of the above commercial network design methodologies shows that there are at least six major properties that impact enterprise network design. These include: (1) the edge connectivity design (multi-homed vs. single-homed); (2) network addressing and naming (private vs. public and routable, for example); (3) the design of subnet and virtual local area networks (VLANs); (4) the degree of redundancy required at the distribution layer; (5) load sharing requirements across enterprise links and servers and (6) the placement and demands of security services such as virtual private networks and firewalls. We next plan to study how network topology properties define the impact of DDoS attacks and defense effectiveness in real enterprise networks.

VI. PERFORMANCE METRICS

To evaluate DDoS defenses we must define an effectiveness metric that speaks to the heart of the problem — *do these defenses remove the denial-of-service effect?* The metrics previously used for this purpose, such as the percentage of attack traffic dropped, fail to capture whether legitimate service continues during the attack. Even if all attack traffic is dropped to preserve a server's capacity, if the legitimate traffic does not get delivered and serviced properly, the attack still succeeds.

We propose a metric that directly expresses whether the legitimate clients received acceptable service or not. This metric requires considering traffic at the application level and considering quality of service needs of each application. Specifically, some applications have strict delay, loss and jitter requirements and will be impaired if any of these are not met. Other real-time applications have somewhat relaxed delay and loss requirements. Finally, there are applications that conduct their transactions without human attendance and can endure significant loss and delay as long as their overall duration is not impaired.

We measure the overall denial-of-service by extracting transaction data from the traffic traces captured at the legitimate sender and the attack target during the experiment. A *transaction* is defined as a high-level task that a user wanted to perform, such as viewing a Web page, conducting a telnet session or having a VoIP conversation. Each transaction is categorized by its application, and we determine if it experienced DoS effect by evaluating if the application's QoS requirements were met. The DoS impact measure expresses the percentage of transactions, in each application category, that have failed.

The proposed metric requires (1) determining which applications are most important, both by their popularity among Internet traffic and the implications for the rest of the network traffic if these applications are interrupted, and (2) determining acceptable thresholds for each application that, when exceeded, indicate a denial-of-service. Both tasks are very challenging, since the proposed applications and thresholds must be acceptable to the majority of network users.

The defense performance metrics must also capture the delay in detecting and responding to the attack, the deployment and

operational cost, and the defense's security against insider and outsider threats. Each of these performance criteria poses unique challenges in defining objective measurement approaches.

VII. MEASUREMENT METHODOLOGY

The benchmark suite will contain many test scenarios, and our proposed metrics will produce several performance measures for a given defense in each scenario. The measurement methodology will provide guidelines on aggregating results of multiple measurements into one or a few meaningful numbers. While these numbers cannot capture all the aspects of a defense's performance, they should offer quick, concise and intuitive information of how well this defense handles attacks and how it compares to its competitors. We expect that the definition of aggregation guidelines will be a challenging and controversial task.

VIII. RELATED WORK

Space does not permit detailed discussion of other related benchmarking efforts. Particularly relevant are:

- IRTF's Transport Modeling Research Group's work to standardize testing methodologies for transport protocols [11].
- The Center for Internet Security's benchmarks for evaluation of operating system security [8]
- Work on quality of service that impacts on our proposed DDoS metrics [10].
- Work on differentiated services (DiffServ) and Per-Hop Behaviors [13], [14].
- Internet topology characterization, represented by [1], [29], [7], [3], [5], [15], [28], [17], among many others.
- Studies on characterizing Internet denial-of-service activity, generally based on limited observations [19], [4].

Briefly, while much existing research has shed light on important aspects of the problem, no previous concerted effort has been made to define all aspects required to create usable DDoS defense benchmarks. Our work borrows liberally from this previous work, wherever possible, but many critical issues require fresh attention.

IX. CONCLUSIONS AND FUTURE WORK

The major remaining technical challenges for DDoS benchmarking are: (1) collecting sufficient trace and topology data to generate typical test suites, (2) understanding the interaction between the traffic, topology and resources and designing comprehensive, yet manageable, test sets, (3) determining a success criteria for each application, (4) defining a meaningful and concise result aggregation strategy, (5) updating benchmarks. The value of any benchmark lies in its wide acceptance and use. The main social challenge for our work lies in gaining acceptance for all three components of our common evaluation methodology from wide research and commercial communities.

Our existing methods have some clear limitations, because they rely on trace analysis for definition of typical scenarios. Only a limited number of traces are currently publicly available, which may bias our conclusions. Keeping in mind these limitations, we believe that information we may glean from traffic traces will still offer a valuable insight for design of realistic test scenarios.

Designing benchmarks for DDoS defenses is sure to be an ongoing process, both because of these sorts of shortcomings

in existing methods and because both attacks and defenses will evolve. However, there are currently no good methods for independent evaluation of DDoS defenses, and our existing work shows that defining even imperfect benchmarks requires substantial effort and creativity. The benchmarks described in this paper represent a large improvement in the state of the art for evaluating proposed DDoS defenses.

REFERENCES

- [1] K. Anagnostakis, M. Greenwald, and R. Ryger. On the sensitivity of network simulation to topology. In *Proc. of MASCOTS*, 2002.
- [2] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. Experiences with deter: A testbed for security research. In *2nd IEEE Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities*, March 2006.
- [3] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *Proc. of IEEE INFOCOM*, June 2002.
- [4] Kun chan Lan, Alefiya Hussain, and Debojyoti Dutta. The Effect of Malicious Traffic on the Network. In *Passive and Active Measurement Workshop (PAM)*, April 2003.
- [5] 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.
- [6] C. Dovrolis and P. Ramanathan. Packet dispersion techniques and capacity estimation. *IEEE/ACM Transactions on Networking*, December 2004.
- [7] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proc. of ACM SIGCOMM'99*, pages 251–262.
- [8] The Center for Internet Security. Cis standards web page. <http://www.cisecurity.org/>.
- [9] L. Gao. On inferring autonomous system relationships in the internet. In *Proc. IEEE Global Internet Symposium*, November 2000.
- [10] M. W. Garrett. Service architecture for ATM: from applications to scheduling. *IEEE Network*, 10(3):6–14, May/June 1996.
- [11] IRTF TMRG group. The transport modeling research group's web page. <http://www.icir.org/tmrg/>.
- [12] Ann Harrison. Cyberassaults hit Buy.com, eBay, CNN, and Amazon.com. Computerworld, February 9, 2000 <http://www.computerworld.com/news/2000/story/0,11280,43010,00.html>.
- [13] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, June 1999. <http://www.ietf.org/rfc/rfc2597.txt>.
- [14] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, June 1999. <http://www.ietf.org/rfc/rfc2598.txt>.
- [15] S. Jin and A. Bestavros. Small-world Characteristics of Internet Topologies and Multicast Scaling. In *Proc. of IEEE/ACM MASCOTS*, 2003.
- [16] K. Lai and M. Baker. Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. In *Proc. of USENIX Symposium on Internet Technologies and Systems*, March 2001.
- [17] 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, UCSD, 2005.
- [18] R. Mahajan, N. Spring, David Wetherall, and Thomas Anderson. User-level internet path diagnosis. In *Proceedings of ACM SOSP*, October 2003.
- [19] D Moore, G Voelker, and S Savage. Inferring internet denial-of-service activity. Proceedings of the 2001 USENIX Security Symposium, 2001.
- [20] Ryan Naraine. Massive DDoS attack hit DNS root servers. <http://www.internetnews.com/dev-news/article.php/1486981>.
- [21] Priscilla Oppenheimer. *Top-Down Network Design*. CISCO Press, 1999.
- [22] N. Spring, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. In *Proceedings of ACM SIGCOMM*, 2002.
- [23] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of ACM IMC*, October 2003.
- [24] Traceroute.org. Traceroute tool, 2006. <http://www.traceroute.org>.
- [25] F. Wang and L. Gao. On inferring and characterizing internet routing policies. In *Proc. Internet Measurement Conference (Miami, FL)*, 2003.
- [26] Websiteoptimization.com. *The Bandwidth Report*. <http://www.websiteoptimization.com/bw/>.
- [27] Russ White, Alvaro Retana, and Don Slice. *Optimal Routing Design*. CISCO Press, 2005.
- [28] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, Univ. of Michigan, 2002.
- [29] E. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proc. of IEEE INFOCOM*, volume 2, pages 594 –602, March 1996.