DDoS Benchmarks and Experimenter's Workbench for the DETER Testbed

Jelena Mirkovic
Songjie Wei
University of Delaware

Alefiya Hussain
Brett Wilson
Roshan Thomas
Stephen Schwab
SPARTA, Inc.

Sonia Fahmy
Roman Chertov
Purdue University

Peter Reiher
University of California Los Angeles

***Abstract*—While the DETER testbed provides a safe environment and basic tools for security experimentation, researchers face a significant challenge in assembling the testbed pieces and tools into realistic and complete experimental scenarios. In this paper, we describe our work on developing a set of *sampled* and *comprehensive* benchmark scenarios, and a workbench for experiments involving denial-of-service (DoS) attacks. The benchmark scenarios are developed by sampling features of attacks, legitimate traffic and topologies from the real Internet. We have also developed a measure of DoS impact on network services to evaluate the severity of an attack and the effectiveness of a proposed defense.**

The benchmarks are integrated with the testbed via the *experimenter's workbench* — a collection of traffic generation tools, topology and defense library, experiment control scripts and a graphical user interface. Benchmark scenarios provide inputs to the workbench, bypassing the user's selection of topology and traffic settings, and leaving her only with the task of selecting a defense, its configuration and deployment points. Jointly, the benchmarks and the experimenter's workbench provide an easy, point-and-click environment for DoS experimentation and defense testing.

I. Introduction

Various network security threats plague today's communication and undermine the Internet's stability and reliability. The DETER testbed [1] was funded by the Department of Homeland Security and the National Science Foundation, and developed by USC Information Sciences Institute and UC Berkeley, with the goal of providing an infrastructure for safe, repeatable and versatile security experimentation. DETER allows security researchers to replicate threats of interest in a secure environment and to develop, deploy and evaluate potential solutions. The testbed has a variety of hardware devices and supports many popular operating systems. Researchers obtain exclusive use of a portion of a testbed, configured into a user-specified topology, and shielded from the outside world via a firewall. DETER's hardware infrastructure was enhanced by a collection of software tools for traffic generation, statistics collection, analysis and visualization, developed in its sister project EMIST [2]. Jointly, DETER and EMIST facilitate reconstruction of numerous security scenarios, where every element of the scenario is customizable by the researcher. However, the task of choosing realistic traffic and topology settings for experimentation remains an open problem.

In this paper, we describe our work on developing benchmarks for denial of service (DoS) and distributed DoS (DDoS) defense evaluation[1], that are integrated with the DETER testbed. We identify three components of a DoS attack scenario that jointly determine attack impact and defense effectiveness: the legitimate traffic, the attack traffic and the topology. We further identify key features of each component that interact with an attack or with a defense and provide two benchmark suites: a *sampled* suite, where key features are sampled from the Internet to provide a set of commonly observed scenarios for DoS experimentation, and a *comprehensive* suite, where features are varied within some predetermined range to thoroughly test a defense against current and future threats. These two suites provide complete and realistic scenarios for DoS experimentation. The benchmark architecture is described in detail in [3]. In this paper, we provide a brief explanation of key design issues, and focus on presenting the scenarios contained in the benchmarks, and the integration of the benchmarks with the DETER testbed.

The benchmark scenarios are reproduced on the DETER testbed via the *security experimenter's workbench*. The workbench provides a set of traffic generation tools, topology and defense libraries and a graphical user interface for experiment specification, control and monitoring. The benchmarks interface with the workbench by providing additions to the topology library, traffic generators and post-analysis via metric calculations. The necessary input from an experimenter is limited to the selection of the defense system (if any) and the specification of its deployment pattern and configuration. In addition to benchmarks, we have developed a novel measure of the effectiveness of a DoS defense, in order to compute the impact of an ongoing attack on network traffic and services [4]. The visual representation of this measure is also integrated with the workbench. Figure 1 illustrates the benchmark components and their integration with the DETER testbed through the experimenter's workbench.

[1]This material is based on research sponsored by the Department of Homeland Security under agreement number FA8750-05-2-0197. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security or the U.S. Government.

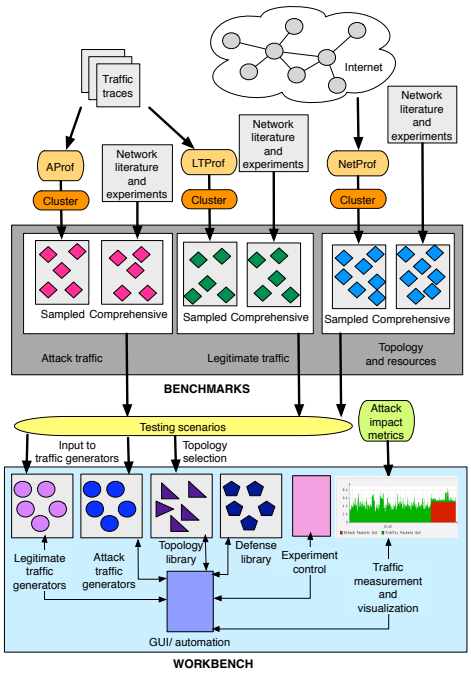


Fig. 1. Benchmark components and their generation

II. Benchmarks for DoS Experimentation

DoS defense benchmarks must specify all the elements of an attack scenario that influence its impact on a network's infrastructure and a defense's effectiveness. We consider these elements along three dimensions:
(1) ***DoS attack*** — features describing a malicious packet mix arriving at the victim, and the distribution and activities of machines involved in the attack.
(2) ***Legitimate traffic*** — features describing communication patterns of the target network.
(3) ***Network topology and resources*** — features describing the target network architecture.

Our first step was to evaluate the attacks, legitimate traffic patterns and topologies that are prevalent in today's Internet. To this end, we designed a collection of tools that harvest traffic and topology samples from the Internet and cluster them to reveal common features. The *AProf* tool collects attack samples from publicly available traffic traces. It uses a variety of detection criteria to discover attack traffic, and then separates it from the other trace traffic and extracts or infers relevant attack features. The *LTProf* tool collects legitimate traffic samples from public traces by creating a communication profile for each observed subnet and host, and deriving relevant traffic feature distributions from these profiles. Topology samples are collected by the *NetProf* tool, which harvests router-level topologies of Internet Service Providers through active probing. We report the results of sampling Internet traffic from three traffic trace collections and sampling topologies of several Internet Service Providers in Section III.

Our next step was to identify those features of the attack traffic, the legitimate traffic and the topology that influence the impact of an attack or the effectiveness of a defense, and to determine how to vary these features to comprehensively test a given defense. This work relies on research papers in the DoS field, the network design literature, and experimentation on the DETER testbed, and we describe it in Section IV.

III. Sampled Scenarios

In this section, we describe our work on sampling traffic and topologies from the Internet.

A. Attack Traffic

Attack traffic samples are obtained from public traffic traces, using our *AProf* tool. For space reasons, we omit the details of this tool but refer the reader to [5].

We have collected attack samples from the following traces:
(1) ***Auckland-VIII*** traffic trace set from NLANR-PMA [6]. This is a two-week random-anonymized trace captured in December 2003 at the link between the University of Auckland and the rest of the Internet. Each daily trace is divided into 24 one-hour traces. So far, we have sampled attacks from 62 one-hour long traces.
(2) ***MAWI traces***, collected at a trans-Pacific backbone link by the WIDE project [7]. The traces contain 15-minute long daily samples of random-anonymized traffic. We have collected attack samples from the first 10 days of June 2006; however, we do not discuss the results in the paper due to space constraints.
(3) ***CAIDA's OC48 traffic trace***, collected in both directions of an OC48 link at the AMES Internet Exchange (AIX) on April 24, 2003. The collected trace is one hour long, and anonymized in a prefix-preserving manner.

Table I shows the characteristics of attacks we have detected in the Auckland-VIII traces. There were 1048 attacks, out of which an overwhelming majority (1024 attacks or 97.7%) were TCP SYN floods. 8 attacks were ICMP floods and 16 were UDP floods. 84 of the SYN floods were performed via randomly spoofed packets, while we could not detect any spoofing in ICMP and UDP floods. In all attacks, we could only observe 1 or 2 sources in the trace, sending at a rate below 5 packets per second. 91% of attacks lasted 5 minutes or less, but 9% lasted one hour, which means that they were present throughout the trace. These long-lasting attacks were low-rate SYN floods. Looking across traces, we identified two victims that were the targets of 42 and 44 hour-long attacks. The victims were targeted continuously for 3-6 hours, and the attack was resumed after 1-3 hours.

Table II shows the characteristics of attacks we have detected in the OC48 traces. There were 35 attacks in the trace, out of which 21 were ICMP floods, 7 were SYN floods and 7 were UDP floods. We detected random spoofing in 2 UDP attacks. For 22 of the attacks, we could only observe 1 source

Type	Count	Percentage	Spoofed
SYN flood	1024	97.7%	8.2% random
ICMP flood	8	0.7%	no
UDP flood	16	1.5%	no

TABLE I
Attacks in 62 hours of the Auckland-VIII trace

port	traffic feature	distribution
53	Requests per second	Poisson(1.828)
	Requests per host	Pareto(1.1,2.17)
	Requests size	Pareto(32.74,2.5)
	Reply size	Pareto(117.5,3.1)
80	Requests per second	Poisson(94.147)
	Requests per host	Pareto(10,2.315)
	Requests size	Pareto(287,2.35)
	Reply size	Pareto(259,2.028)

TABLE III
OUTGOING TRAFFIC MODELS FOR 0.3.117.0/24

in the trace; 6 had 2–10 sources; 4 had 10-20 sources and 2 had 25-30 sources. The packet rate for 32 attacks was below 8 pps and for the remaining three it was less than 29 pps. 54.3% of attacks lasted 5 minutes or less, 25.7% lasted between 5 and 15 minutes, and the remaining 20% lasted between 15 minutes and one hour, which was the duration of the trace. The long-lasting attacks were low-rate UDP floods.

These results of attack sampling agree with our expectations that contemporary attacks deploy more sophisticated means than just simple packet flooding. TCP SYN packets are critical for many businesses; thus attacks that deploy these packets cannot be easily filtered. Hosts that do not deploy TCP SYN cookies can be overwhelmed with as few as 100 TCP SYN packets per second. ICMP attacks deployed short packets, targeting CPU resources. UDP attacks mostly targeted DNS port 53. The low attack rate and low number of sources in the samples indicate that a large part of an attack was not visible in the traces, because they were collected near a few attack sources.

B. Legitimate Traffic

The legitimate traffic dimension of the benchmarks consists of subnet and host models that describe their sending and receiving behavior. Together, these models are used to drive traffic generation during testing. Each edge network in an experiment is assigned a role of a subnet from the trace, and its traffic to/from the rest of the Internet is generated using this subnet's model. Host models are used to produce the input for the traffic generation tools.

We build subnet models by first identifying /24 and /16 subnets in a traffic trace anonymized in a prefix-preserving manner. For each subnet, we identify the total traffic sent from and received by this subnet, and select subnets that either receive or produce more than a certain amount of traffic for further modeling.

We model separately a subnet's incoming and outgoing traffic for each well-known port number. Within the selected traffic mix, we identify individual sessions between two IP addresses, and extract the distributions of the number and length of service requests, the reply length and the request inter-arrival time. These distributions are used during an experiment to drive the traffic generation. For example, the outgoing traffic from the anonymized network 0.3.117.0/24 consists of traffic to port 53 and port 80, with the characteristics shown in Table III. The *LTProf* tool automates this traffic modeling.

A related traffic modeling approach is deployed in the Swing tool [8], which models traffic over a single network link, using custom client and server code. Our goal was to develop traffic models for communication between a given subnet and the rest of the Internet, possibly over multiple links. Another difference is that we use real applications for traffic generation, to obtain more accurate traffic dynamics.

Type	Count	Percentage	Spoofed
SYN flood	7	20%	no
ICMP flood	21	60%	no
UDP flood	7	20%	28.5% random

TABLE II
ATTACKS IN OC48 DATA

C. Topology

It is important to have representative topologies for DDoS experimentation both at the Internet level and at the enterprise level. To reproduce topologies containing multiple autonomous systems (ASes) at the router level, we developed the *NetTopology* tool. The tool invokes traceroute commands from different servers, performs alias resolution, and infers several routing (e.g., Open Shortest Path First (OSPF) routing weights) and geographical (e.g., location) properties. This tool is similar to *RocketFuel* [9], and was developed because *RocketFuel* is no longer supported.

To generate topologies that can be used on a testbed like DETER, we have developed two additional tools, combined into the *NetProf* toolkit: (i) *RocketFuel-to-ns*, which converts topologies generated by the *NetTopology* tool or *RocketFuel* to DETER-compliant configuration scripts, and (ii) *RouterConfig*, a tool that takes a topology as input and produces router (software or hardware) BGP and OSPF configuration scripts, according to the router relationships in the specified topology.

A major challenge in reproducing realistic Internet-scale topologies in a testbed setting is the scale-down of a large, multi-thousand node topology to a few hundred nodes available on DETER [1], while retaining relevant topology characteristics. The *RocketFuel-to-ns* tool allows a user to select a subset of large topology, specifying a set of Autonomous Systems or performing a breadth-first traversal from a specified point, with specified degree and number-of-nodes bounds. The *RouterConfig* tool operates both on (a) topologies based on real Internet data, and on (b) topologies generated from the GT-ITM topology generator [10]. To assign realistic link bandwidths in our topologies, we use information about typical link speed distribution published by the Annual Bandwidth Report [11].

We have ported to DETER six topologies that are subsets of the original *RocketFuel* topologies (Telstra (Australia), Sprintlink (US), Ebone (Europe), Verio (US), Tiscali (Europe), Level3 (US), Exodus (US), VSNL (India), Abovenet (US), and AT&T (US)), containing 20, 20, 21, 50, and 79 nodes. The topology for AS 1239 (Sprintlink) contains 79 nodes and is shown in Figure 2. To further facilitate experiments that investigate the impact of topology on attacks and defenses, we have also ported a number of GT-ITM topologies to DETER.

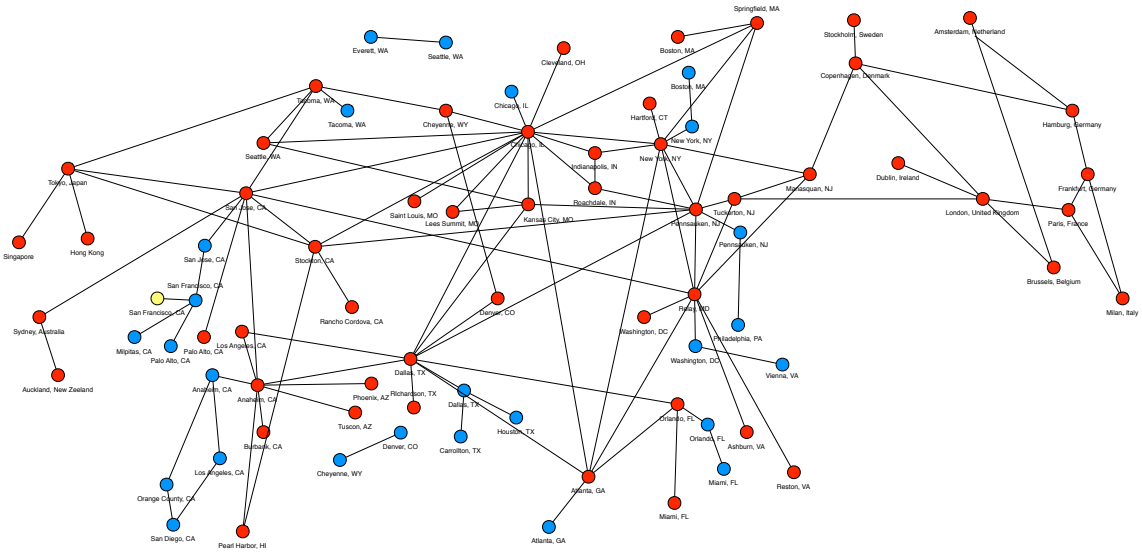


Fig. 2. Rocketfuel topology of the AS 1239 — Sprintlink (red nodes) with some routers from APNIC (blue) and Exodus (yellow)

These topologies have varying degrees of inter-domain and intra-domain connectivity but they all contain 36 nodes.

Since many end-networks filter outgoing ICMP traffic, the *NetTopology* tool cannot collect end-network topologies. To overcome this obstacle, we analyzed enterprise network design methodologies typically used in the commercial marketplace to design and deploy scalable, cost-efficient production networks. An example of this is Cisco's classic three-layer model of hierarchical network design that is part of Cisco's Enterprise Composite Network Model [12], [13]. This consists of the topmost core layer which provides Internet access and ISP connectivity choices, and a middle distribution layer that connects the core to the access layer and serves to provide policy-based connectivity to the campus. Finally, the bottom access layer addresses the design of the intricate details of how individual buildings, rooms and work groups are provided network access, and typically involves the layout of switches and hubs. We used these design guidelines to produce end-network topologies with varying degrees of complexity and redundancy.

IV. Comprehensive Scenarios

While sampled scenarios could be used by researchers to reproduce traffic and topologies typically seen in the Internet, they are insufficient for thorough evaluation of a proposed DoS defense. Our goal in defining comprehensive scenarios was to understand which features of the attack, the legitimate traffic and the topology interact with each other and with the defense. Once isolated, these features are varied in comprehensive scenarios to thoroughly test a defense.

We first collected information about all the known DoS attacks and categorized them based on the mechanism they deploy to deny service. We focused on attacks where denial of service is a primary goal not a secondary effect, i.e., attacks that target a specific resource or a service as opposed to DoS created by worms or e-mail viruses. We consider the following attack categories:

(1) ***Packet floods*** — service is denied because some key resource is exhausted. This resource could be bandwidth (if the flood volume is large), router or end host CPU (if packet rate of the flood is high) or tables in memory, created by the end host operating system or application (if each attack packet creates a new record in some table). Packets in bandwidth and CPU exhaustion floods can belong to any transport and application protocol, as long as they are numerous, and may contain legitimate transactions, e.g., flash crowd attacks. An attacker can use amplification effects such as reflector attacks, to generate large-volume floods. Examples of memory exhaustion floods are TCP SYN floods and random fragment floods.

(2) ***Unexpected header values*** — service is denied because some device (router, switch, firewall, end host) en route to the destination or the end host application cannot process malformed packets and crashes. The anomalies can be in the IP header, IGMP header, transport or application headers.

(3) ***Invalid application inputs*** — the attacker generates invalid packet content that causes an application to freeze or crash.

(4) ***Invalid fragments*** — the end host cannot properly handle the case of overlapping fragments (teardrop attack) or fragments that are reassembled into too large a packet (boink attack), and crashes.

(5) ***Large packets*** — usually lead to a buffer overflow at the end host (in the OS or the application), causing a crash of the host or the application. One example of large packet attacks is the ping-of-death attack.

(6) ***Congestion control exploits*** — the attacker creates the impression at a sender that there is congestion on the path. If the sender deploys a congestion control mechanism, it reduces its sending rate. Examples of such attacks are TCP ECE floods, shrew attacks and ICMP source quench floods.

(7) ***Impersonation attacks*** — the attacker spoofs a host's identity to take over its traffic, to blackhole its traffic or to kill its ongoing communications via fake messages. Examples

Feature	Variation
Rate	Low, moderate and severe.
Dynamics	Continuous rate vs. pulsing (vary on and off periods). Synchronous senders vs. interleaved senders
Legitimate traffic rate	Light, moderate and high traffic load on the bottleneck link
Critical resource	Covered by attack rate variations
Path sharing	Uniform vs. log-normal location of attack machines. Legitimate clients are distributed uniformly. Several topologies with various degrees of path sharing.
TCP traffic mix	80%/15%/5% mixes of traffic, choosing from: data transfers, Telnet-like communication and single-message request/reply exchanges.
Application mix	Create a mix of all supported applications and vary the contribution of each application to the mix.

TABLE V
FEATURE VARIATIONS THAT INFLUENCE DOS IMPACT

of such attacks are DNS and ARP poisoning, and ICMP unreachable message floods.

Out of the listed categories, only packet floods and congestion control exploits require continuous generation of attack messages, and will benefit from distributed attackers, so only these two categories are included as DDoS benchmarks. Table IV lists all the attack types in the comprehensive benchmark suite, and their denial-of-service mechanisms. Although there are a few attack categories, they can invoke a large variety of DoS conditions and challenge defenses, by varying attack features such as sending dynamics, spoofing and rates. All packet flood attacks can be converted into congestion control exploits by sending the flood in pulses.

Attack type	DoS mechanism
UDP/ICMP packet flood	Large packets consume bandwidth, while small packets consume CPU
TCP SYN flood	Consume end-host's connection table
TCP data packet flood	Consume bandwidth or CPU
HTTP flood	Consume Web server's CPU or bandwidth
DNS flood	Consume DNS server's CPU or bandwidth
Random fragment flood	Consume end-host's fragment table
TCP ECE flood	Invoke congestion control
ICMP source quench flood	Invoke congestion control

TABLE IV
ATTACK TYPES IN THE COMPREHENSIVE BENCHMARK SUITE

Attack traffic generated by the listed attacks interacts with legitimate traffic by creating real or perceived contention at some critical resource. The level of service denial depends on the following traffic and topology features: (1) Attack rate, (2) Attack traffic on and off periods in case of pulsing attacks, (3) The rate of legitimate traffic, (4) Amount of critical resource — size of connection buffers, fragment tables, link bandwidths, CPU speeds, (5) Path sharing between the legitimate and the attack traffic prior to the critical resource, (6) Legitimate traffic mix at the TCP level — connection duration, connection traffic volume and sending dynamics, protocol versions at end hosts, (7) Legitimate traffic mix at the application level — since different applications have different quality of service requirements, they may or may not be affected by a certain level of packet loss, delay or jitter. Table V lists the feature variations included in our benchmark suite for each attack type listed in Table IV. A single feature is varied during a test, while other features are kept at a specific default value.

Our next step was to collect information about all the proposed countermeasures that could apply to packet floods and congestion control exploits, and to categorize them based on their defense mechanism. We categorize the specific mechanisms that help detect, prevent, or counter attacks; a single defense could embody several mechanisms.

(1) ***Path isolation*** — routers mark, sample or record packets to isolate traffic paths. Path information can be used to deploy filters on the path, or to perform fair sharing of resources.

(2) ***Privileged customer*** — some customers obtain "passes" that allow privileged access to the critical resource, in form of capabilities, authorization to enter a dedicated overlay, knowledge of the server's identity, good classification, etc. A defense prioritizes traffic with "passes."

(3) ***Traffic baselining*** — many traffic parameters are observed over time to learn their valid value ranges. During attacks, some parameter values will exceed their predicted range, which can be used to devise fine-grain filters or to isolate attack packets.

(4) ***Resource multiplication*** — distributed resources are deployed (statically or dynamically) to sustain large attacks.

(5) ***Legitimate traffic inflation*** — legitimate traffic is multiplied to enhance its chances to win in the fight for the limited resource.

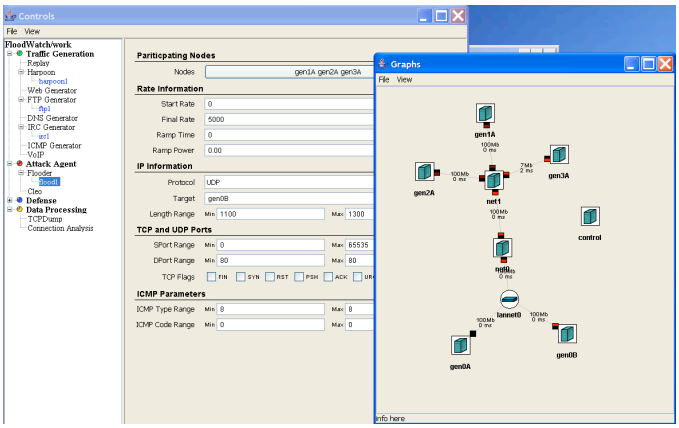
Table VI lists features that should be varied to stress-test a given defense, and their range of variation.

V. PERFORMANCE METRICS

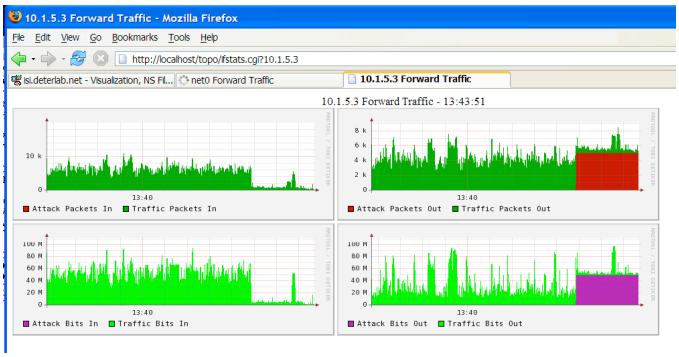
The DoS experimentation field lacks a metric that accurately captures if service has been denied and to what extent. We have proposed such a metric in [4], and we have devised tools to compute this metric from traffic traces. Briefly, we categorize

Defense	Feature	Variation
Path isolation and resource multiplication	Path sharing	(1) Uniform vs. log-normal distributed attackers. (2) Pulsing, interleaved attacks
Privileged customer	Resource access pattern	Attacker mimics legitimate client behavior (1) prior to the attack or (2) throughout the experiment
Traffic baselining	Legitimate vs. attack traffic parameters	(1) Randomized attack packets, (2) Attacker mimics legitimate client traffic, (3) Attack with slowly increasing rate
Traffic inflation	Legitimate user vs. attacker network resources	Vary attackers' locations
All	Attacker aggressiveness	Vary number of attack machines while keeping attack rate constant.
All	Attacker dynamics	Engage new attackers during the attack and retire old ones.
All	Legitimate client dynamics	Engage new legitimate clients during the attack and retire old ones.

TABLE VI
FEATURE VARIATIONS THAT INFLUENCE DEFENSE EFFECTIVENESS



(a) Experimentation palette and topology visualization



(b) Visualization of legitimate and attack traffic at a chosen interface

Fig. 3. Experimentation palette

all applications based on their quality of service requirements. We then observe user "transactions" in traffic traces captured during the experiment and extract transaction features. Each transaction is considered as "succeeded" if it meets all the QoS requirements of its application or "failed" otherwise. The QoS requirements take into account the application semantics (some packets are more important than others) and contain application-specific delay, jitter and loss bounds. While this measurement approach is likely too simplistic to capture the complex human user experience of service quality, it provides an objective and simple metric of service quality for testbed experimentation. Our main DoS impact measure — *DoS-hist* — expresses the percentage of transactions, in each application category, that have failed during an experiment. An effective defense should quickly reduce the percentage of failed transactions to zero.

VI. Integration with DETER

The methodology and tools for DoS experimentation are integrated with the DETER testbed [1] through the experimenter's workbench. The workbench enables even a novice experimenter to reproduce complex scenarios by selecting various experimental elements from a pre-defined palette. The palette provides a range of canonical and realistic scaled-down topologies that can be used for the experiments. These topologies allow the experimenter to perform evaluations with a range of routing, bandwidth, and delay configurations. The palette also provides support for interactive and bulk background traffic generators that allow creation of traffic that resembles Internet traffic. The palette supports traffic generation using off-the-shelf servers such as Apache, vsftpd and sshd. It also supports definitions of statistical distributions, such as Pareto, Gamma, or Exponential, for the frequency of client service requests, file sizes and connection durations. Host models developed using *LTProf* provide a direct input to these traffic generators. A single physical host can act as multiple IP addresses using virtualization; the address range is specified via the palette. This enables us to reproduce low-rate communications between a large number of hosts, with appropriate address diversity. The palette further allows generation of a wide range of sophisticated attacks, including both the attacks observed in the Internet to date, and attacks proposed in research papers. Finally, the palette provides support for trace collection and visualization at all locations in the topology.

Figure 3 shows the workbench palette in operation on a small canonical topology. A stand-alone Java application runs locally on the experimenter desktop, and communicates with the DETER server using xmlrpc to send commands to each node in the topology. The palette allows the user to simply click at an interface of any node in the topology to visualize the incoming and outgoing packet and byte rates. The traffic is color-coded so that legitimate and attack traffic can be visually distinguished from each other. This significantly reduce the barrier for experimentation for a novice user. The DETER workbench also provides support for experiment automation and repeatability in addition to the user interface, through a Perl-based intuitive scripting interface allowing an experienced user to rapidly execute a large set of experiments in batch mode.

Using the guidelines outlined in Section III and Section IV, our testing methodology frames systematic questions that guide an experimenter in selecting and combining the appropriate experimental elements. A preliminary round of defense system evaluation usually dictates using several sampled scenarios that permit rapid testing of the system with a pre-defined topology, legitimate traffic, and attack traffic combination. After a successful preliminary evaluation, the experimenter will likely need to perform detailed stress testing of the system with a range of comprehensive scenarios. We are currently modifying the workbench interface to allow scheduling of a series of comprehensive tests that are automatically selected based on defense mechanisms present in the defense being tested. These mechanisms will be specified by a user via the palette interface.

VII. Related Work

The Center for Internet Security has developed benchmarks for evaluation of operating system security [14], and large security bodies such as CERT and SANS maintain checklists of known vulnerabilities that can be used by software developers to test the security of their code. However, much remains to be done to define rigorous, clear and representative tests for various security threats. This is especially difficult in the DoS

field as there are many ways to deny service and many variants of attacks, while the impact of a given attack on a target network further depends on various network characteristics including its traffic and resources.

In [15], the authors propose a traffic generation method for online Intrusion Detection System (IDS) evaluation. They extract selected legitimate traffic features from a full packet trace (including packet contents), and generate packets with these features using a custom "Harpoon" tool and state automata for various protocols. This traffic is less realistic than traffic generated by real client and server applications, as implemented in our benchmarks. The authors further develop a list of representative attacks for IDS evaluation. Their list contains a few DoS and DDoS attacks, but their features are kept at default values during evaluation.

Selecting representative benchmark topologies with realistic routing parameters and realistic resources and services is an extremely challenging problem [16]. Internet topology characterization has been the subject of significant research for over a decade (e.g., see [10], [17]) and we draw on this work in our benchmark topology generation.

Several Internet researchers have attempted to characterize Internet denial-of-service activity [18], [19]. Compared to our work on attack benchmarks, they used more limited observation approaches and a single traffic trace collection. Moreover, both of these studies were performed several years ago, and attacks have evolved since then.

Finally, there is a significant body of work on traffic modeling [20], [21], [22], but there is a lack of unifying studies that observe communication patterns across different networks and the interaction of this traffic with denial-of-service attacks. Our work aims to fill this research space.

VIII. Conclusions

Live testbed experimentation is usually preceded by a lengthy setup of topology, routing and traffic generators. Researchers are further hindered by the necessity to evaluate multiple topology and traffic settings to select a few that are complex and realistic enough for experimentation. We have developed a set of *benchmark scenarios* and a *security experimenter's workbench* that simplify and significantly speed up the setup for DoS experimentation.

The sampled suite in our benchmarks contains traffic and topology samples from the Internet, and allows for rapid testing. The comprehensive suite provides more complete tests, customized to challenge a proposed DoS defense. Additionally, our attack impact metric facilitates accurate measurement of the success of an attack and the effectiveness of a defense.

The benchmarks are integrated with the DETER testbed via the experimenter's workbench, which provides an easy interface for experiment customization, control and analysis. The benchmarks free the user from the low-level tasks of configuring tools and selecting realistic scenarios, so her attention can be focused on analyzing experimental data and improving a proposed defense. Jointly, the experimenter's workbench and the benchmarks greatly reduce the barrier for DoS experimentation and defense evaluation.

IX. Acknowledgments

We are grateful to David Bettis, Pankaj Kumar, and Abdallah Khreishah who developed the *NetTopology*, *Rocketfuel-to-ns* and *RouterConfig* tools. We are also grateful to Erinc Arikan who developed the *APprof* tool.

References

[1] 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 TridentCom*, March 2006.

[2] EMIST project. Evaluation methods for internet security technology. http://www.isi.edu/deter/emist.temp.html.

[3] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, and P. Reiher. Benchmarks for DDoS Defense Evaluation. In *MILCOM*, 2006.

[4] J. Mirkovic, P. Reiher, S. Fahmy, R. Thomas, A. Hussain, S. Schwab, and C. Ko. Measuring Denial-of-Service. In *Proceedings of the 2006 Quality of Protection Workshop*, October 2006.

[5] E. Arikan. Attack profiling for ddos benchmarks. MS thesis, University of Delaware, August 2006.

[6] Nlanr pma special traces archive. http://pma.nlanr.net/Special.

[7] Mawi traffic archive. http://tracer.csl.sony.co.jp/mawi/.

[8] K. V. Vishwanath and A. Vahdat. Realistic and Responsive Network Traffic Generation. In *Proceedings of ACM SIGCOMM*, September 2006.

[9] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with RocketFuel. In *Proceedings of ACM SIGCOMM*, 2002.

[10] E. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proc. of IEEE INFOCOM*, volume 2, pages 594 –602, March 1996.

[11] Websiteoptimization.com. *The Bandwidth Report*. http://www.websiteoptimization.com/bw/.

[12] Priscilla Oppenheimer. *Top-Down Network Design*. CISCO Press, 1999.

[13] Russ White, Alvaro Retana, and Don Slice. *Optimal Routing Design*. CISCO Press, 2005.

[14] The Center for Internet Security. CIS Standards Web Page. http://www.cisecurity.org/.

[15] J Sommers, V Yegneswaran, and P Barford. Toward Comprehensive Traffic Generation for Online IDS Evaluation. Technical report, Dept. of Computer Science, University of Wisconsin, August 2005.

[16] K. Anagnostakis, M. Greenwald, and R. Ryger. On the Sensitivity of Network Simulation to Topology. In *Proc. of MASCOTS*, 2002.

[17] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, Univ. of Michigan, 2002.

[18] D Moore, G Voelker, and S Savage. Inferring Internet Denial-of-Service Activity. Proceedings of the 2001 USENIX Security Symposium, 2001.

[19] 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.

[20] Bell Labs. Bell Labs Internet Traffic Research. http://stat.bell-labs.com/InternetTraffic/index.html.

[21] ICSI Center for Internet Research. Traffic Generators for Internet Traffic. http://www.icir.org/models/trafficgenerators.html.

[22] Hei Xiaojun. Self-Similar Traffic Modelling in the Internet. http://www.ee.ust.hk/~heixj/publication/comp660f/comp660f.html.