AFRL-RI-RS-TM-2008-3
Final Technical Memorandum
January 2008

BENCHMARKS FOR EVALUATION OF DISTRIBUTED DENIAL OF SERVICE (DDOS)

University of Delaware

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TM-2008-3 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
JOHN F. PERRETTA Work Unit Manager	WARREN H. DEBANY, JR. Technical Advisor, Information Grid Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* JAN 2008	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* APR 05 – JUL 07

4. TITLE AND SUBTITLE

BENCHMARKS FOR EVALUATION OF DISTRIBUTED DENIAL OF SERVICE

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-05-2-0197

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Jelena Mirkovic

5d. PROJECT NUMBER
DHSD

5e. TASK NUMBER
MH

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Delaware
103 Smith Hall
Newark DE 19716

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIGB
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TM-2008-3

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-0129

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The main goal of our work was to develop the benchmark suite for evaluation of defenses against distributed denial-of-service (DDoS) attacks. The desired features of the benchmark suite were the following:

1. Realistic topologies, legitimate and attack traffic are represented in the suite
2. A wide variety of attack variants is present in the suite
3. Benchmarks can be used by novice experiments easily
4. There is a common, intuitive and scientifically accurate measure of an attack's impact on network services in any given scenario.

This measure is easily obtained by experimenters and can be used to compare effectiveness of diverse defenses.

15. SUBJECT TERMS
Computer Network Defense, DDoS Mitigation, Denial of Service, Distributed Computing, Distributed Denial of Service

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** John F. Perretta
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UL	12	**19b. TELEPHONE NUMBER (*Include area code)*** N/A

Summary

The main goal of our work was to develop the benchmark suite for evaluation of defenses against distributed denial-of-service (DDoS) attacks. The desired features of the benchmark suite were the following:

1. Realistic topologies, legitimate and attack traffic are represented in the suite
2. A wide variety of attack variants is present in the suite
3. Benchmarks can be used by novice experimenters easily
4. There is a common, intuitive and scientifically accurate measure of an attack's impact on network services in any given scenario. This measure is easily obtained by experimenters and can be used to compare effectiveness of diverse defenses.

We started the work by developing tools to harvest realistic topologies and traffic from the Internet. University of Delaware developed the *LTProf* and *AProf* tools to harvest the legitimate and attack traffic from packet traces. Purdue developed the *NetProf* tool to harvest topologies from the Internet and infer the router configuration from these topologies. These tools are described in the deliverables section in more detail.

We next applied these tools to available packet traces and scanned the Internet to collect representative traffic and topology samples. We were greatly hindered in this step by:

- Lack of publicly available traces, which we needed to collect legitimate and attack traffic samples. We were hoping that the PREDICT project will be opened to researchers during the life of our project, which would have given us an access to significantly larger and more up-to-date traces than are available from public trace archives. Since that did not happen we resorted to using public trace archives and some private traces we were able to obtain and that fitted the needs of our programs. Specifically, *LTProf* program requires a prefix-anonymized trace collected at the edge network. Only CAIDA's DatCat OC48 trace fitted this requirement. *AProf* program requires traces that are consistently anonymized and relatively long. We used AUCK8 traces from http://pma.nlanr.net/Special/auck8.html, and Los Nettos traces that we obtained from USC/ISI, for this purpose.
- Filtering rules at many networks that prohibit ICMP replies to foreign addresses. Without ICMP replies, it is impossible to map a network's interior. We worked around this problem by mapping several AS-level topologies with our tools and inferring the common design of the edge topologies from network design literature instead of direct mapping.

We analyzed the collected samples and designed a comprehensive list of attacks in the wild and variants of these attacks that could be seen in the future. We also analyzed the influence legitimate traffic and topology settings have on an attack's outcome and a defense's effectiveness. We finally converged on a set of legitimate traffic, topologies and attacks that are included in the benchmark suite.

All four parties then worked on developing a common, quantitative and accurate measure of an attack's impact on network services. We met repeatedly, exchanged suggestions, and experimented with various existing measures of network service degradation and converged finally on a collection of novel metrics we developed. These metrics are also described in more detail in the deliverables section.

The benchmark suite was integrated with the DETER testbed by SPARTA and the University of Delaware via the following steps: (1) developing a set of traffic generators and automating defense and traffic collector placement, (2) developing a GUI for experiment visualization and easy traffic generator, statistics collector and defense placement, (3) developing an automated scenario generator that translates benchmark specifications into scripts to be ran on DETER, and (4) developing a GUI for batch running of several experiments sequentially and result summarization. The SPARTA-developed tool is known as SEER and is currently being extended to support a broad range of other experiments, in addition to DDoS.

Deliverables

This section describes the deliverables produced by our project and points to locations containing the source code for various tools we developed.

1. *AProf* Tool

AProf tool processes packet traces in *tcpdump* format and extracts the attack samples. The tool's source code is located at http://www.isi.edu/~mirkovic/bench/.

Attack sample generation is performed in these four steps:

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. Because many applications generate two-way traffic (TCP-based applications, ICMP echo traffic, DNS traffic), some of our attack detection tests use the absence of the reverse 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 one-way traffic by recognizing one-way TCP traffic, and performing some legitimacy tests on this traffic to ensure that it is not part of the attack. Each TCP connection is recorded and initially flagged as one-way and legitimate. The one-way flag is reset if we observe reverse traffic. The connection is continuously tested for legitimacy by checking if its sequence or acknowledgment numbers increase monotonically. Each failed test adds some amount of suspicion points. Connections that collect a sufficient number of suspicion points have their legitimate flag reset. When the connection is terminated (we see a TCP FIN or RST packet or no packets are exchanged during a specified interval), its IP information is written to the `one-way.trc` file, if its one-way and legitimate flags were set. In the second pass, we remove from the original trace all packets between pairs identified in `one-way.trc`, producing the refined trace `distilled.trc`.

2. *Attack detection* is performed by collecting traffic information from the `distilled.trc` 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. Each direction of a connection will generate one connection and one destination record. A packet 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 an attack, which may be detected on the packet's destination. We perform the following checks to identify attack traffic:

- We identify attacks that use aggressive TCP implementations (such as Naptha attack), bypassing the TCP stack, or fabricate junk TCP packets using raw sockets, by checking for a high sent-to-received TCP packet ratio on a destination record, or for mismatched sequence numbers on a TCP connection. If an attack is detected, TCP packets going to the attack's target will all be identified as attack traffic.
- We identify TCP SYN attacks by checking for a high SYN-to-SYNACK packet ratio on a destination record. All TCP SYN packets going to the target will be flagged as attack traffic.
- We identify TCP no-flag attacks by checking for the presence of TCP packets with no flags set. Only no-flag TCP packets will be flagged as attack traffic.
- Some UDP applications require responses from the destination of the UDP traffic (e.g., DNS). The absence of these responses is measured through high sent-to-received UDP packet ratio on a given destination record, and used to identify UDP attacks. In case of one-way UDP traffic (such as media traffic), we will identify an attack if there is no accompanying TCP connection between a given source and destination pair, and there has been a sudden increase in UDP traffic to this destination. In case of an attack, all UDP traffic will be flagged as attack traffic.
- High-rate ICMP attacks using ICMP echo packets are detected by checking for high echo-to-reply ICMP packet ratio on a destination record. All ICMP packets to this destination will be flagged as attack traffic.
- We detect known-malicious traffic carrying invalid protocol numbers, same source and destination IP address, or private IP addresses. All packets meeting this detection criteria will be flagged as attack.
- We check for packet fragmentation rates that are higher than expected for Internet traffic (0.25%) and we identify all fragmented traffic in this case as part of an attack.

3. *Legitimate and attack traffic separation*. Each packet is classified as legitimate or attack as soon as it is read from the trace, using the attack detection criteria described above. If more than one attack is detected on a given target, we apply precedence rules that give priority to high-confidence alerts (e.g., TCP no-flag attack) over low-confidence alerts (high TCP packet-to-ack ratio). 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`. When a new attack is detected, attack type and victim IP are written to a file called `victim.out`.

4. *Attack sample generation.* Attack samples are generated using the `attack.trc` file by first pairing each attack's trace with the information from `victim.out`, and then extracting the attack information such as spoofing type, number of sources, attack packet and byte rates, duration and dynamics from the attack trace and compiling

them into an attack alert. 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.

Our public trace analysis indicated that an overwhelming majority of attacks are TCP SYN attacks. Each attack machine is participating at a very low rate (2-5 packets per second), presumably to stay under the radar of network monitors deployed at the source, and attacks range in duration from several minutes to several hours.

More details about the AProf tool are in the Erinc Arikan's MS thesis, which is at http://www.isi.edu/~mirkovic/bench/.

2. Comprehensive Attack Scenarios

We examined all known DDoS attack variants to populate the benchmark suite with attacks that are challenging to a variety of defenses and commonly seen in the wild. On the other hand, we had to ensure that benchmarks will contain only the necessary tests that can be ran with a moderate investment of experimenter's time (several hours). A comprehensive coverage of all possible attack variations would lead to several-days worth of tests and would preclude benchmarks' use by experimenters.

The list of attacks we examined and our categorization of these attacks can be found at http://www.isi.edu/~mirkovic/bench/.

The benchmarks now contain the attack categories shown in Table 1.

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

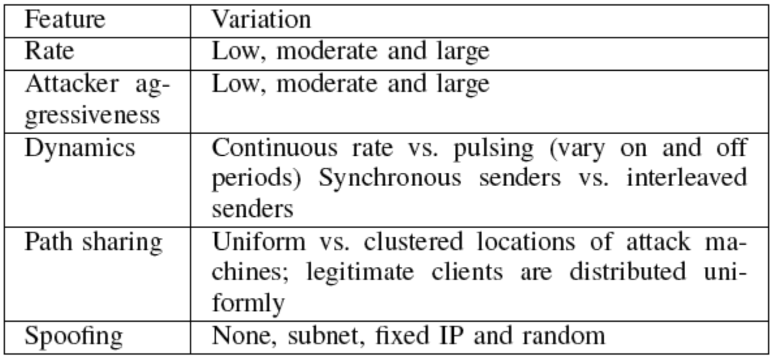
Figure 1: Attack categories in the benchmark suite

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.

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 distribution, (3) Attack traffic on and off periods in case of pulsing attacks, (4) The rate of legitimate traffic relative to the attack, (5) Amount of critical resource - size of connection buffers, fragment tables, link bandwidths, CPU speeds, (6) Path sharing between the legitimate and the attack traffic prior to the critical resource, (7) Legitimate traffic mix at the TCP level --- connection duration, connection traffic volume and sending dynamics, protocol versions at end hosts, (8) 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.

If we assume that the legitimate traffic mix and topological features are fixed by inputs from our legitimate traffic models and topology samples, we must vary the attack rate, distribution, dynamics and path sharing to create comprehensive scenarios. Additionally, presence of IP spoofing can make attacks more challenging to some DDoS defenses.

Figure 2 lists the feature variations included in our benchmark suite for each attack type shown in Figure 1.



Feature	Variation
Rate	Low, moderate and large
Attacker aggressiveness	Low, moderate and large
Dynamics	Continuous rate vs. pulsing (vary on and off periods) Synchronous senders vs. interleaved senders
Path sharing	Uniform vs. clustered locations of attack machines; legitimate clients are distributed uniformly
Spoofing	None, subnet, fixed IP and random

Figure 2: Attack variations in the benchmark suite

3. *LTProf* Tool

LTProf tool processes packet traces in *tcpdump* format and extracts the legitimate traffic samples. The tool’s source code is at http://www.isi.edu/~mirkovic/bench/.

The tool produces legitimate traffic models that describe communication between a set of active clients and a network that is the target of a DDoS attack. It collects legitimate traffic samples from public traces by creating a communication profile for each observed subnet and deriving relevant traffic feature distributions from these profiles. These distributions then serve as an input to the Workbench's traffic generators.

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 received by it and select the largest receivers to act as target networks in our scenarios. We then identify subnets that send a significant percentage of traffic to this target network and model their sending behavior.

We model separately a sender's 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.

The *LTProf* tool automates this traffic modeling and produces for each target network a set of outgoing traffic models for its most active client subnets. These models can be fed directly into the SEER's traffic generators.

4. *NetProf* Tool

For DDoS experimentation, we are interested in modeling topologies of the target network and its Internet Service Provider. We refer to these as *end-network* topology and *AS-level* topology. *NetProf* tool infers AS-level topologies and consists of *NetTopology*, *RocketFuel-to-ns* and *RouterConfig* tools. The source code for these tools and sample topologies collected with these tools can be obtained from http://www.cs.purdue.edu/homes/fahmy/software/rf2ns/index.html.

AS-level topologies consist of router-level connectivity maps of selected Internet Service Providers. They are collected by the *NetTopology* tool, which Purdue developed. The tool probes the topology data by invoking *traceroute* commands from different servers, performing alias resolution, and inferring several routing (e.g., Open Shortest Path First routing weights) and geographical properties. This tool is similar to *RocketFuel*, and was developed because *RocketFuel* is no longer supported.

Purdue further developed tools to generate DETER-compatible input from the sampled topologies: (i) *RocketFuel-to-ns*, which converts topologies generated by the *NetTopology* tool or *RocketFuel* to DETER *ns* scripts, and (ii) *RouterConfig*, a tool that takes a topology as input and produces router BGP and OSPF configuration scripts.

A major challenge in a testbed setting is the scale-down of a large, multi-thousand node topology to a few hundred nodes available on DETER, while retaining relevant topology characteristics. The *RocketFuel-to-ns* tool allows a user to select a subset of a 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. To assign realistic link bandwidths in our topologies, we use information about typical link speed distribution published by the Annual Bandwidth Report.

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. 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.

5. Performance metrics

We have developed several metrics to measure the impact of a DDoS attack on network services. Our goal was to design versatile metrics applicable to many scenarios involving different services, attacks and defenses. At the same time we sought to design simple and intuitive metrics that can be easily used in testbed experiments or in simulation. We observed early on that denial of service is a subjective phenomenon. It depends on a user's perception of service quality and it is this perception that we need to quantify. To reach this goal we investigated quality of service measurements, and how they apply to different network services. This resulted in a set of traffic features that need to be measured, and a set of thresholds corresponding to these features that must not be exceeded for satisfactory service.

We further observed that a user's interaction with a server consists of smaller units we call *transactions*. Each transaction completes a task that is meaningful to a user, such as download of one file or delivery of one instant message. A transaction can *succeed* or *fail* depending on whether its traffic parameters exceeded their thresholds. Our primary attack impact measure is the *percentage of failed transactions (pft)* during an attack, for each service in the network. In addition to this we devised the following secondary measures. *QoS-degrade* shows how many times is a service's quality below the level expected by a user, i.e., it quantifies the severity of service disruption. *DoS-level* is an aggregate measure that shows the mean *pft* for all network services. *Failure ratio* shows *pft* measure over time for a particular service, thus facilitating measurement of the timeliness of a defense's response, e.g., how long it took for a defense to restore service quality in the network. *Life diagram* shows birth and death of each failed and succeeded transaction, grouped per service. It facilitates a fine-grained analysis that can help researchers understand behavior of an attack or of a defense. For example, if a defense erroneously drops all new HTTP connections this is easily visible on a life diagram.

We have developed an automated tool to extract transactions and their traffic features from a tcpdump trace collected during an experiment, compare these features to thresholds and calculate all the above metrics. The tool is called *perf* and can be downloaded from: http://www.isi.edu/~mirkovic/bench/.

6. Integration with DETER

We have integrated DDoS benchmarks we developed, and the performance metrics, with DETER via the following tools: (1) The SEER tool which provides a set of traffic generation tools, topology and defense libraries and a graphical user interface for experiment specification, control and monitoring, and (2) The *Experiment Generator* that receives an input from the benchmark suite, and glues together a set of selected topologies, legitimate and attack traffic into a DETER-ready experiment. This experiment can be deployed and run from the SEER at the click of a button.

SEER is a more general tool that helps users easily deploy traffic generators, monitoring software and several defenses that have been integrated with DETER. It also helps users visualize traffic during a run and dynamically drive their experimentation from a GUI. The *Experiment Generator* integrates benchmarks with SEER by creating experiment scripts SEER understands from benchmark specifications.

The version of SEER with the experiment generator and benchmark specifications can be downloaded from: http://www.isi.edu/~mirkovic/bench/.

7. Promised vs Delivered Items

We now compare the list of deliverables above with what was promised in the proposal:

1. Typical benchmark suite and future benchmark suite (later renamed as "comprehensive benchmark suite" by us) morphed into a single suite integrated with our benchmark implementation in DETER. The reason for this was that we had very few public traces available to us for collection of attack samples. While we discovered some patterns in those samples, and prevalence of flooding TCP SYN attacks, the number of samples was too small to generalize this into the typical benchmark suite.
2. Stress-test scenario benchmarks suite was integrated with the typical and future benchmarks into a unified suite and is in our benchmark implementation in DETER.
3. Specification of performance measures was completed as described in Section 5.
4. Specification of testing methodology was completed and is described in publication [12]
5. LTPRof, AProf and NetProf tools were completed as described in Sections 1, 3 and 4.
6. A partial report on attack trends is contained in publication [1]. As described in item 1 of this list, lack of public traces prevented us in making more general conclusions about attack trends.
7. A partial report on legitimate traffic patterns is contained in publication [13]. As described in item 1 of this list, lack of public traces prevented us in making more general conclusions about legitimate traffic patterns.
8. We could not gather sufficient data about topologies to produce a report on common network topologies and their robustness. This was due to several factors: (1) A lot of organizations filter outgoing ICMP replies so we could not map their topologies; all existing mapping tools require ICMP replies in order to learn about topologies. (2) We had to rewrite the Rocketfuel tool (this became NetTopology tool) that we planned on using for topology mapping, because this tool was not actively maintained and we could not get it to work in its present state. (3) While NetTopology tool automates a lot of scanning process, alias detection and disambiguation require human intervention. This slows down topology sample collection. Overall, we collected 23 topologies, which was too small a sample to draw general conclusions.

We note that our work on benchmarks, understanding DDoS and collection and classification of attack, legitimate traffic patterns and topologies continues in spite of the end of this grant. Since PREDICT project is expected to become open to researchers very soon now, we plan to use its data to complete our traffic studies. In parallel with this, our topology scanning continues, as well as experiments in DETER to understand how DDoS experimentation can be scaled down with fidelity.

8. Publications

This project resulted in the following publications:

[1] E. Arikan, Attack Profiling for DDoS Benchmarks, MS Thesis, University of Delaware, August 2006.

[2] J. Mirkovic, A. Hussain, B. Wilson, S. Fahmy, P. Reiher, R. Thomas, W. Yao, and S. Schwab, Towards User-Centric Metrics for Denial-Of-Service Measurement, Proceedings of the Workshop on Experimental Computer Science, June 2007

[3] J. Mirkovic, S. Wei, A. Hussain, B. Wilson, R. Thomas, S. Schwab, S. Fahmy, R. Chertov and P. Reiher DDoS Benchmarks and Experimenter's Workbench for the DETER Testbed, Proceedings of the Tridentcom 2007, May 2007.

[4] J. Mirkovic, A. Hussain, B. Wilson, S. Fahmy, W. Yao, P. Reiher, S. Schwab and R. Thomas When Is Service Really Denied? A User-Centric DoS Metric, Proceedings of the Sigmetrics 2007, June 2007

[5] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, and P. Reiher Benchmarks for DDoS Defense Evaluation, Proceedings of the Milcom 2006, October 2006

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

[7] J. Mirkovic, B. Wilson, A. Hussain, S. Fahmy, P. Reiher, R. Thomas and S. Schwab, Automating DDoS Experimentation, Proceedings of the DETER Community Workshop on Cyber Security Experimentation, August 2007

[8] J. Mirkovic, S. Fahmy, P. Reiher, R. Thomas, A. Hussain, S. Schwab, and C. Ko, Measuring Impact of DoS Attacks, In Proceedings of the DETER Community Workshop on Cyber Security Experimentation, June 2006.

[9] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, P. Reiher, Benchmarks for DDoS Defense Evaluation, In Proceedings of the DETER Community Workshop on Cyber Security Experimentation, June 2006.

[10] R. Chertov, S. Fahmy, N. B. Shroff, High Fidelity Denial of Service (DoS) Experimentation, In Proceedings of the DETER Community Workshop on Cyber Security Experimentation, June 2006.

[11] R. Chertov, S. Fahmy, P. Kumar, D. Bettis, A. Khreishah, N. B. Shroff, Topology Generation, Instrumentation, and Experimental Control Tools for Emulation Testbeds, In Proceedings of the DETER Community Workshop on Cyber Security Experimentation, June 2006.

[12] A. Hussain, S. Schwab, R. Thomas, S. Fahmy, and J. Mirkovic, DDoS Experiment Methodology, In Proceedings of the DETER Community Workshop on Cyber Security Experimentation, June 2006.

[13] S. Wei, J. Mirkovic and E. Kissel, Profiling and Clustering Internet Hosts, Proceedings of the 2006 International Conference on Data Mining, June 2006