AFRL-IF-RS-TR-2004-333
Final Technical Report
December 2004

DISTRIBUTED DENIAL OF SERVICE-DEFENSE ATTACK TRADEOFF ANALYSIS (DDOS-DATA)

Johns Hopkins University - APL

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. M101

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

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

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-333 has been reviewed and is approved for publication.

APPROVED: /s/

DAVID E. KRZYSIAK
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	DECEMBER 2004	Final Jun 01 – May 03

4. TITLE AND SUBTITLE
DISTRIBUTED DENIAL OF SERVICE-DEFENSE ATTACK TRADEOFF ANALYSIS (DDOS-DATA)

5. FUNDING NUMBERS
C - F30602-01-2-0531
PE - 602310E
PR - FTNP
TA - M1
WU - 01

6. AUTHOR(S)
W. J. Blackert, R. L. Hom,
A. K. Castner, R. M. Jokerst,
D. M. Gregg, and E. M. Kyle

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Johns Hopkins University-APL
11100 Johns Hopkins Road
Laurel Maryland 20723-6099

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFGA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2004-333

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: David E. Krzysiak/IFGA/(315) 330-7454/ David.Krzysiak@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
The project uses modeling and simulation to analyze performance of mitigation technologies to combat Distributed Denial of Service Attacks. The system also determines how an attacker can react to mitigation technologies and how mitigation technologies can be layered to reduce attacker effectiveness.

14. SUBJECT TERMS
Distributed Denial of Service, DDOS Attack, Modeling and Simulation

15. NUMBER OF PAGES
49

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

Distributed Denial of Service (DDOS) attacks disrupt and deny legitimate computer and network resource usage through compromised hosts that monopolize resources. Mitigation technologies have been developed to defend against DDOS attacks, but there is little understanding of the fundamental relationships among DDOS attacks, mitigation strategies, and attacker performance. Without a solid understanding of these fundamental relationships, it is difficult to determine the ability of mitigation technologies to address the DDOS problem or how mitigation technologies can successfully be deployed together. The Johns Hopkins University Applied Physics Laboratory (JHU/APL), under sponsorship of the Defense Advanced Research Projects Agency's Fault-Tolerant Networks Program, has been conducting the DDOS Defense Attack Tradeoff Analysis (DATA). DDOS-DATA is using modeling and simulation (M&S) to analyze mitigation technology performance, determine how an attacker can react to mitigation technologies, and understand ways mitigation technologies can be layered to reduce attacker effectiveness. DDOS-DATA is a 2-year effort; this final report summarizes the first year's results, discusses the analysis methodology, and documents the analysis results for the second year.

1.1 SCOPE

Numerous possible network configurations, attack options, and mitigation strategies can be analyzed for DDOS-DATA. The following decisions were made to scope the analysis effort:

• Target network – DDOS-DATA is analyzing a 500+ node target network that represents an existing JHU/APL network.

• Attacker – The attacker's goal is to deny usage of JHU/APL's internal Web server through server resource monopolization or bandwidth flooding.

• Mitigation technologies – During its first year, DDOS-DATA focused on three mitigation technologies:

  − Active Monitor, which is a system based on Purdue CERIAS' *synkill*, which monitors network traffic and resets invalid traffic (Reference 1)

  − Rate Limiter, which configures a router with Cisco's Committed Access Rate (CAR) to limit how much traffic of a certain type it can pass (Reference 2)

  − Proof-of-Work, which requires clients to "pay" for server usage by solving a puzzle (Reference 3)

The second year effort expanded the mitigation technologies to include D-WARD, which is a system that computes dynamic rate limits based on its classification of traffic flows and connections (Reference 4) and NetBouncer, which is a Network Associates, Inc. (NAI) system that uses legitimacy tests to differentiate legitimate traffic from attack traffic and forwards packets it deems legitimate (Reference 5). Figure 1-1 shows the deployment of these technologies in a notional network.

1.2 ANALYSIS METHODOLOGY

There are a variety of options for analyzing computer network attacks and mitigation strategies. Closed-form analysis may be the most desirable form, but can require many simplifying assumptions. The resulting models provide valuable first-order insights, but detailed analysis

using them is limited. An alternative approach is a real-world testbed, which is an excellent approach to understand attack dynamics. However, a testbed can be limited in its ability

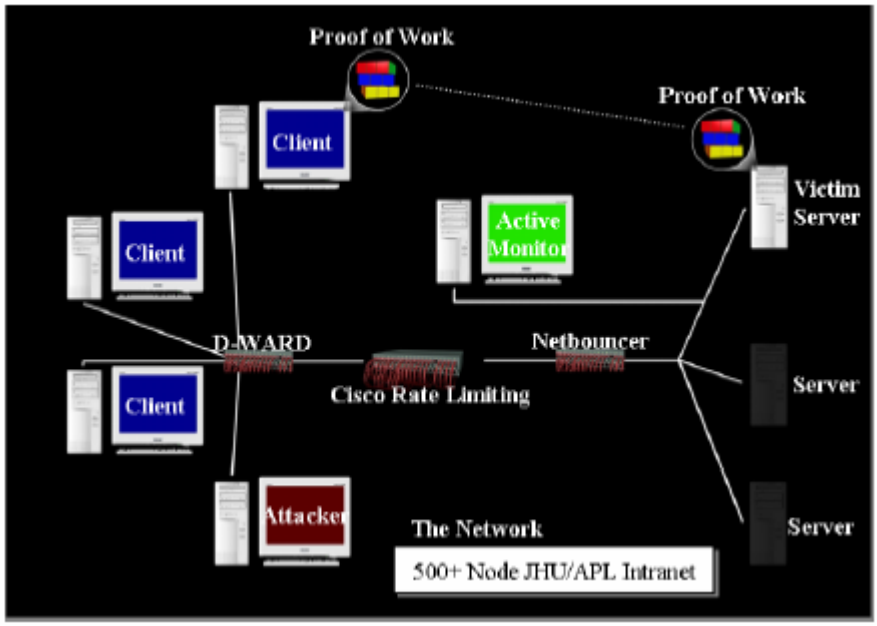

Figure 1-1 Mitigation Technology Deployment

to vary key parameters (e.g., the packet-forwarding speed of a router), and size limitations can restrict the analysis of DDOS attacks that may use hundreds of nodes. M&S provides an approach with several advantages over closed-form and real-world testbed analysis: the ability to vary key parameters that may not be easily modifiable in a testbed, the ability to easily repeat a given analysis scenario, and the use of models without debilitating simplifications. However, successfully using M&S requires model validation, which can be very time consuming. In addition, the tradeoff between model fidelity and model run time must be carefully considered. Because of the flexibility and level of detail it provides, the DDOS-DATA analysis uses M&S. Analysis using M&S requires an in-depth understanding of how attacks and mitigation technologies function. This is accomplished at JHU/APL through literature surveys, code examination, and experimentation in JHU/APL's Information Operations Laboratory. Through this process, key parameters and behaviors are identified that then drive model requirements and design. OPNET Modeler, a commercial discrete event network simulation package, is used for model development. Development requires enhancing existing OPNET models (e.g., to build the target network model) or creating models from scratch (e.g., to build the attack and mitigation models). Because computer network attacks often exploit nuances in protocol implementations and existing OPNET models adhere to the protocol specifications, they are typically not susceptible to attack without enhancements. Model verification and validation are critical to the modeling process. Without them, it is simply not possible to derive value from the results. Verification ensures that the model correctly implements the developer's intent (i.e., "Did I implement it right?"). Validation takes many forms, but often compares the model to the real system ensuring correct model behavior (i.e., "Did I implement the right thing?"). After the models have been constructed, verified, and validated, the analysis (Figure 1-2) begins. To perform the analysis, it is first necessary to determine metrics. These metrics must be quantitative in nature and provide a means to compare system performance across many scenarios. Given the metrics, the target network is examined under benign (i.e., no attack) conditions, under attack conditions, and under attack conditions with mitigation technologies in place.

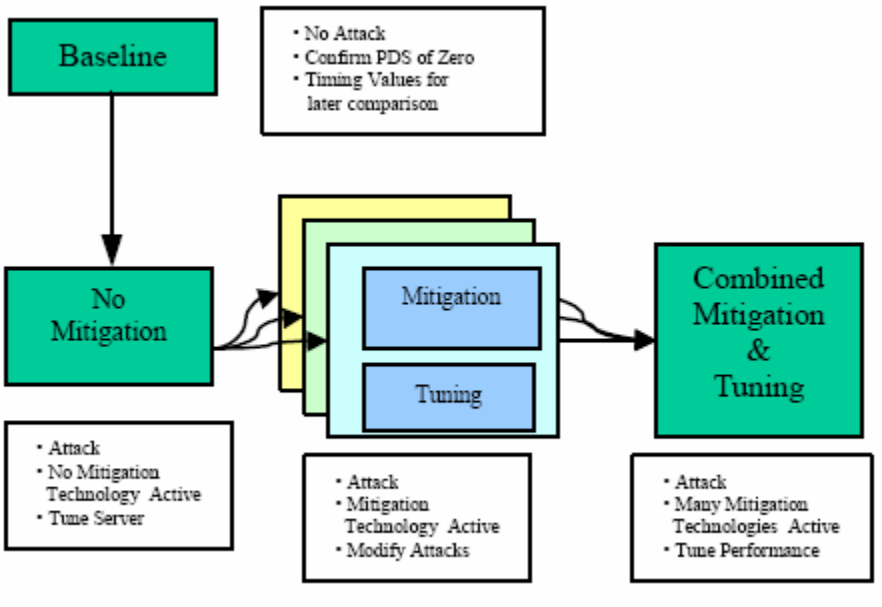

Figure 1-2 Analysis Flow

1.3 YEAR 1 ANALYSIS OVERVIEW (RESULTS SUMMARY)

The DDOS-DATA first-year effort (Reference 6) quantified the relationships between DDOS attacks and the Active Monitor, Rate Limiter, and Proof-of-Work mitigation technologies. These technologies were analyzed individually and, where appropriate, together. This section summarizes the results for individual technology analysis and combined technologies analysis. (Appendix B provides more detailed summary charts extracted from Reference 6.) When individual technologies were analyzed, the findings were as follows:

• Mitigation technologies that classify nodes (e.g., Active Monitor) are effective at freeing resources as long as the classifier behaves correctly. For the classifier to behave correctly, the schemes should account for data loss (e.g., packets lost in the network or SYN-ACKs not being returned by a server). Because an attacker can also use the classifier to the attacker's benefit (e.g., cause legitimate nodes to be misclassified, thereby using the mitigation technology to deny service), the classification algorithms must be robust.

• The rate-limiting technique based on Cisco's CAR restricts a particular attack to using a predetermined amount of network bandwidth. While this will allow other processes to continue, dropping packets blindly will cause an attack's effect to be amplified as the user and attacker compete for the limited bandwidth. This finding suggests that deploying rate limiting closer to the attack source and using more information to make packet discard decisions may be beneficial.

• Requiring a client to make a payment can be an effective countermeasure against Denial of Service (DoS) attackers if a single system cannot afford to make payments at the attack rate. However, a distributed attack that spreads the payment mechanism throughout the network is an effective means of countering these techniques. While this technique was overcome with a relatively small (i.e., 45-node) distributed attacker, it did force the attacker to use the Internet

Protocol (IP) address of the compromised host. This restriction could facilitate attack attribution. The combinations of Rate Limiter/Active Monitor and Active Monitor/Proof-of-Work were also analyzed. These analyses found the following:

• Combining Rate Limiter and Active Monitor and then increasing the bandwidth available to Web connection requests showed that Active Monitor was able to increase availability compared to Rate Limiter alone. While the results were worse than Active Monitor alone, they do demonstrate that the two mitigation technologies can be successfully combined; namely, Rate Limiter restricts network bandwidth while Active Monitor, using its successful classification of connections, resets connections from any passed attack packets.

• When combining Active Monitor and Proof-of-Work, the distributed attack developed against the Proof-of-Work technology was successfully mitigated because Active Monitor was able to reset packets originating from the attackers. Unfortunately, the attacker was able to adapt to this scenario and create an attack that successfully misclassified nodes because Proof-of-Work interfered with the Transport Control Protocol (TCP) three-way handshake. Analyses of these two combined technologies suggest that mitigation technologies can be successfully combined, but only when they either share information (e.g., if Proof-of-Work and Active Monitor shared information, Active Monitor would not have misclassified nodes in the second attack) or when they are designed to not interfere with each other (e.g., the parameters over which the systems operate are independent). Because Rate Limiter preempts the three-way handshake by discarding packets, Active Monitor is able to monitor the remaining handshakes. However, because the Proof-of-Work process, by discarding the SYN packet at the server, interrupted the three-way handshake being monitored by Active Monitor, an attacker was given an opportunity to deny service.

1.4 YEAR 2 ANALYSIS OVERVIEW

Subsections 1.4.1 through 1.4.4 describe the analysis, target network, attack, and measures of effectiveness that are the focus of this report.

1.4.1 ANALYSIS CASES

1.4.1.1 Baseline

This analysis examines the network with no attack or mitigation technology. The user's ability to receive Web pages from a given Web server is monitored, and values needed to tune the mitigation technologies are collected. That is, some mitigation technologies require knowledge of network behavior to operate. Where necessary, these parameters are derived from the baseline analysis. The baseline analysis is used to confirm there is no DoS when no attack is present (i.e., the network is well behaved under normal conditions) and establishes nominal handshake delay, server data rate, and client data rate.

1.4.1.2 System Tuning

The protocols that facilitate network communications use a wide variety of parameters that, in many cases, can be tuned. This analysis experiments with a variety of resources to determine the effect on performance in the face of an attack.

1.4.1.3 One Mitigation Technology

This analysis considers the D-WARD and NetBouncer mitigation technologies individually. Baseline attack effectiveness is calculated and further analyses are performed, as appropriate, to understand the interaction between an adaptive attacker and the mitigation technology.

1.4.1.4 Multiple Mitigation Technologies

This analysis considers the effect of combining mitigation technologies (in addition to the year-two technologies, technologies from the year-one effort are revisited in this case). Baseline attack effectiveness is calculated and further analyses are performed, as appropriate, to understand the interaction between an adaptive attacker and mitigation technology performance.

1.4.2 TARGET NETWORK PARAMETERS

The target network is a 500+ node subset of JHU/APL's intranet. This network is composed of five switches that provide host connectivity. These switches are connected to a central switch that then connects to the core network. The core network is represented as a router that provides connectivity to all JHU/APL servers and the Internet. Figure 1-3 presents the network without mitigation technologies deployed. The hexagons on the right-hand side of the figure each represent an OPNET subnet containing a single switch and associated hosts. Additional details and network validation results are contained in the DDOS-DATA verification and validation reports (References 7 and 8).

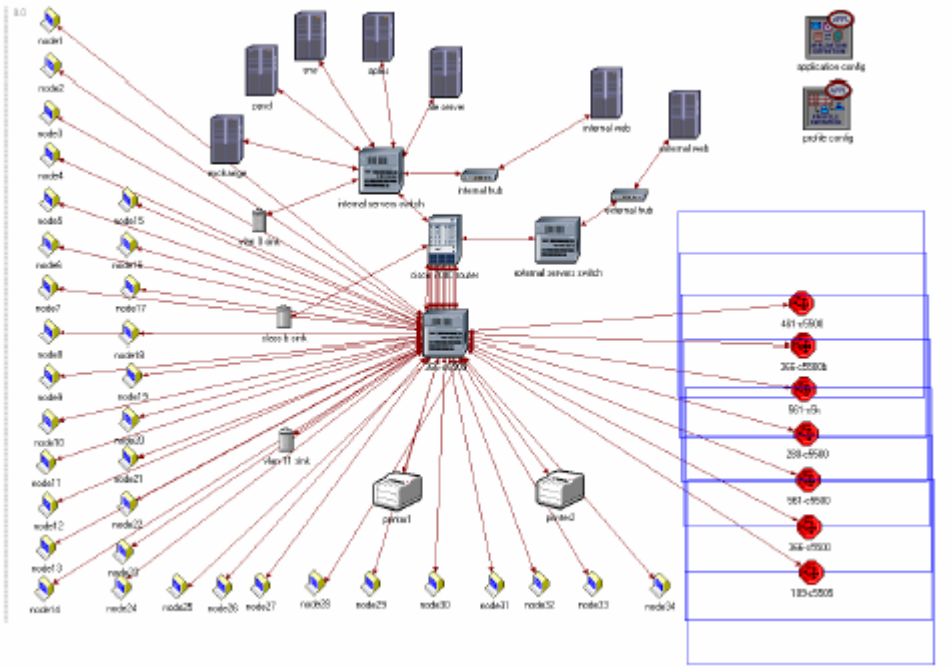

Figure 1-3 Target Network Topologies

A large variety of parameters drive the target network's traffic levels and overall behavior. Traffic is generated by either the continuous Markov models described in Reference 6 or OPNET application models. Table 1-1 presents traffic parameters used in OPNET application models that drive Web client and server behavior. In addition to Web traffic parameters, the underlying target network parameters also influence system behavior. Table 1-2 summarizes key analysis parameters.

Parameter	Value
Internet Web First Object Size	Lognormal (2.17e3,1.59)
Internet Web Other Object Size	Lognormal (2.63e3,1.55)
Internet Number of Objects	Weibull (0.172,0.6763)
Intranet InterRequest Time	Exponential (mean = 1390.1)
Intranet InterRequest Time	Exponential (mean = 231.6)

Table 1-1 Web Traffic Parameters

Parameter	Value
Attack Start Time	4500 sec*
Attack End Time	5500 sec
TCP Pending Connection Queue Size	8192
TCP Connection Retransmission	Attempts based
TCP Connection Retransmission Number of Attempts	Three
TCP Connection Retransmission Timeout (RTO) Multiplier	Two
Network Bandwidth	100 Mbps

*The attack is delayed to 4500 seconds to allow the network to reach steady state.

Table 1-2 Target Network Parameters

As appropriate, these parameters are varied to determine their impact on the analysis results.

1.4.3 THE ATTACK

While DDOS attacks have existed for some time, their evolution has focused on automated deployment processes and enhanced control capabilities and not on the development of new and/or more effective attack methods (Reference 9). DDOS-DATA has focused on the TCP SYN flood attack because it is still a pertinent threat, and many mitigation technologies are designed to defend against TCP SYN flooding. When a TCP connection is initiated, the client[1] begins the connection by sending a TCP packet with the SYN bit set. The server receives this SYN packet, places an entry in the pending connection queue to record this event, and transmits a SYN-ACK packet. If the client is legitimate, it then transmits an ACK packet. The server, upon receiving the ACK packet, considers the connection complete and removes the connection from the pending connection queue.

[1] The terms client and server are being used to describe the two parties involved in the connection. While different terminology would be appropriate in a peer-to-peer data exchange, the concept is the same.

The TCP SYN flood attack relies on the finite length of the TCP pending connection queue. If a SYN packet is received while the pending connection queue is full, no SYN-ACK packet is transmitted and a connection cannot occur. This results in DoS. A TCP SYN flood is used to establish baseline results in all analyses. Subsequent analyses use attackers that have adapted to the deployed mitigation technology. In addition to a TCP SYN flood and its variations (e.g., a specially designed SYN burst), bandwidth flooding is considered. Bandwidth flooding denies service by creating a bandwidth bottleneck in the network. When legitimate traffic attempts to pass through the network, the bottleneck can cause packets to be dropped. Dropped packets can then result in a decreased quality of service (QoS) and potential connection failure.

1.4.4 MEASURES OF EFFECTIVENESS

The DDOS-DATA analysis focuses on connectivity to the internal Web server. Because the internal Web server uses TCP, measures of effectiveness for the attacker and the legitimate client are framed with respect to TCP connections. A TCP connection consists of three parts: the connection establishment, the data transfer, and the connection teardown, as illustrated in Figure 1-4. A TCP connection is established via the TCP three-way handshake, which begins with connection initiation (marked by client SYN sent) and ends with connection establishment acknowledgement (marked by ACK of SYN-ACK received at server). After the TCP connection is established successfully, data is transferred between the client and server using a sliding window protocol that facilitates retransmissions and ordering of received data packets. The TCP connection teardown begins when data transfer is complete (marked by client FIN sent). A TCP connection is completed when the connection teardown ends successfully (marked by ACK of FIN-ACK received at server).

Figure 1-4 TCP Connection

Because the attacker's goal is to deny service to legitimate clients, attack effectiveness is defined as a function of service availability. There are two conditions for a client to successfully receive service. The first is that the TCP connection is established using the three-way handshake. The probability of connection establishment (P_{CE}) is computed as P_{CE} = (Number Initiated Connections) / (Number Established Connections) [1-1]. A P_{CE} of one implies that all legitimate client TCP connection initiations result in successful connection establishment.

The second condition for successful client service is full data transfer. If the legitimate client successfully establishes a TCP connection, network conditions or mitigation technologies can conceivably cause sufficient packet loss to cause the connection to abort. The probability of connection completion given connection establishment ($P_{CC|CE}$) is computed as $P_{CC|CE}$ = (Number Complete Connections) / (Number Established Connections) [1-2]. Given these two metrics, the probability of a complete connection is simply $P_{CC|CE} * P_{CE}$. Because the attacker's goal is to deny service, attack effectiveness is the complement of service availability. The probability of denied service (P_{DS}) is computed as[2] $P_{DS} = 1 - (P_{CC|CE} * P_{CE})$ [1-3]. If a connection is established and completes, QoS metrics are used to compare performance across analysis scenarios. The following metrics consider time to establish a connection, packet loss, data loss, and observed data rate:

[2] In the year-one analysis report, $P_{CC|CE}$ was 1.0.

• Handshake delay[3] measures time elapsed between the beginning of the servicerequest (first SYN sent) and service establishment (SYN-ACK received). In attack conditions, the legitimate client may have to retry multiple times to initiate a connection, resulting in increased waiting time. The calculation of handshake delay requires a successful connection establishment; thus, client delay is a conditional metric requiring P_{CE} greater than 0.

[3] Handshake delay was named client delay in the year-one analysis report.

• Connection quality is a QoS metric for client/server data exchange. A TCP connection that is established and completed successfully does not guarantee that data is transferred as intended (i.e., service availability does not imply connection quality). Connection quality is calculated as the ratio of number of dropped packets to total number of sent packets. A connection quality of one implies that no packets are dropped during data transfer.

• Client ratio and server ratio are QoS metrics that measure data loss in the network. Client ratio[4] measures the number of bytes received by the server from the client vs. the total number of bytes sent to the server by the client. A client ratio of one implies that there is no data lost in the network on the links from the client to the server.

[4] Server ratio is calculated analogously.

• Data rate is another QoS metric for client/server data exchange. Data rate is calculated as the total bytes transferred during the TCP connection divided by the connection duration and is measured in bytes per second (Bps). A decrease in data rate implies degraded service quality.

Mitigation technologies have their own metrics, depending on the system. For example, when multiple mitigation technologies are activated in the network, the contribution of each mitigation technology to preventing the attack is computed as the change in P_{DS} (ΔP_{DS}). Another applicable mitigation technology metric is differential impact (DI), which compares P_{DS} for multiple mitigation techniques with P_{DS} for a single mitigation technique: $DI = \min(P_{DS1}, P_{DS2}) - P_{DS12}$ [1-4]

P_{DS1} and P_{DS2} represent P_{DS} values for single mitigation technologies, and P_{DS12} represents P_{DS} for the combined mitigation technologies. All P_{DS} values are for the same attack. A negative DI indicates that the mitigation technologies interfered with each other, while a positive DI suggests synergy between the two techniques. In all cases, metrics are computed by averaging results over multiple simulation runs. This Monte Carol approach is necessary to account for network performance variation caused by the model's use of probability distributions.

2.0 BASELINE PERFORMANCE

The network is examined without active mitigation technologies to establish baseline performance. First, the network is examined to ensure that the system has no DoS when there is no attack (i.e., the network is well behaved) and to establish baseline performance values. The network is then subjected to an attack, and performance metrics are measured as a basis for comparison of future model runs with one or more mitigation technologies activated.

2.1 NO ATTACK

To ensure that the network is well behaved, the target network is run with no attacks and no active mitigation technology, and data are collected from each host via the internal Web server. The collected data are processed to calculate average P_{DS}, as well as other metrics useful in analyzing model performance. As expected, average P_{DS} is 0 throughout the run. Table 2-1 summarizes the metric values for the no attack scenario.

	Average P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
Normal network behavior	0	0.00	683,170	336,555

Table 2-1 Baseline No Attack Performance

2.2 ATTACK WITH NO MITIGATION PRESENT

The initial attack is a single attacker sending TCP SYN packets from spoofed IP addresses at a rate of 1000 packets per second (pps). These packets are sent to the internal Webserver, which has been configured with a pending connection queue size of 8192.[5] The attack begins at 4500 seconds and ends at 5500 seconds. The other parameters under control of the attacker are the IP address and source port ranges used in the attack. Because of different mitigation technology requirements, two attack variations have been investigated using a source IP from a Class B (i.e., 2_{16} = 65536 possible source addresses) or Class C (i.e., 2_8 = 256 possible source addresses) network (Reference 11). Subsections 2.2.1 through 2.2.7 establish performance parameters for these configurations.

[5] Results on variable queue size are available in Reference 10.

2.2.1 CLASS B NETWORK CONFIGURATION

The attacker configuration that is used unless otherwise noted spoofs the source IP address from the JHU/APL Class B network range. Table 2-2 summarizes the performance metrics averaged over 50 model runs.

	P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
Baseline SYN flood (1000 pps)	0.65	2.39	644,894	177,435

Table 2-2 Baseline SYN flood attack results (50 runs)

2.2.2 CLASS C NETWORK CONFIGURATION

Egress filtering can restrict an attacker's ability to spoof addresses. For example, an attacker could be confined to a Class C network. To investigate this configuration, the previous analysis is repeated with the constraint that the attacker can only spoof from a Class C network. Because many of the Class C addresses are assigned to legitimate hosts in the network model, connection resets will occur when the active host receives a SYN-ACK from the server. When the server receives the reset, it frees the corresponding slot in the pending connection queue, causing the queue to empty more quickly, and the attack's effect to be less severe. Table 2-3 summarizes the performance in this case.

	P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
Class C address SYN flood (1000 pps)	0.41	1.32	653,447	245,574

Table 2-3 Restricted address space results (50 runs)

2.2.3 VARYING LINK BANDWIDTH

Link bandwidth determines the speed at which data can be transferred through a link and therefore the potential for data transfer. To determine the effect of varying the link bandwidth on a network with no attack underway, a normal background traffic load was placed on the network. The link bandwidth for every link in the 500+ node JHU/APL network is decreased from 100 to 10 Mbps. Table 2-4 shows that, as expected, decreasing the link bandwidth results in longer handshake delays and lower data rates.

Link Bandwidth (Mbps)	Average P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
10	0.00	0.001	282,013	114,877
100	0.00	0.0002	683,170	336,555

Table 2-4 Summary of Link Bandwidth Variation (50 runs)

2.2.4 VARYING LINK BANDWIDTH UNDER SYN FLOOD ATTACK CONDITIONS

To determine if link bandwidth influences attack effectiveness, the previous scenario was repeated with the addition of a 1000-pps SYN flood attack. Table 2-5 shows that the link speed does not significantly impact attack effectiveness. The average P_{DS} and handshake delay are not significantly different. The server data rate is comparable to the no attack case, indicating that the attack has little effect on this metric. The overall decrease in client data rate is because of the large increase in handshake delay (2.4 seconds) during the attack. However, the decrease factor of client data rate in the 10-Mbps and 100-Mbps cases is consistent with the no-attack case, suggesting that this attack's impact is independent of link bandwidth.

Link Bandwidth (Mbps)	Average P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
10	0.62	2.45	202,246	51,166
100	0.65	2.39	644,894	177,435

Table 2-5 Variation of Network Bandwidth under attack conditions (50 runs)

2.2.5 VARYING NETWORK DELAY AND PACKET LOSS

The mitigation technologies analyzed for DDOS-DATA exist at the network's edge; that is, the JHU/APL network model is sufficient to exercise the mitigation technologies. However, if the network is expanded to a larger Internet-like system, more loss and delay than currently present

would occur. To determine the impact of increased loss and delay, a node that causes both packet loss and latency is inserted in the link between the router and internal switch. Packets traveling to and from the internal network are dropped with a certain probability [the packet loss rate (PLR)] and delayed by a certain amount of time. The range of PLRs and delays are selected from representative research on Internet dynamics (References 12, 13, and 14). As shown in Table 2-6, varying the network delay and PLR had negligible impact on network availability. Average P_{DS} only increases slightly with increasing PLR and delay. In both cases, the increase in P_{DS} is almost entirely because of an increase in $P_{CC|CE}$, indicating that the handshake completes but service is degraded because of packet loss. As expected, data rates are impacted by the increased connection length that results from dropped packets and delay.

Delay(s)	PLR (%)	P_{DS}	$P_{CC\|CE}$	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
0	0	0.00	1.00	0.0002	683,170	336,555
0	5	0.002	0.99	0.27	180,567	108,638
0	10	0.02	0.98	0.48	91,268	57,917
1	0	0.003	0.99	3.00	1,061	345
1	5	0.02	0.98	3.23	651	228
1	10	0.05	0.95	3.60	480	176

Table 2-6 Variation of network delay and PLR (50 runs)

2.2.6 VARYING NETWORK DELAY AND PLR WITH A SYN FLOOD

To determine the effects of PLR and delay on attack effectiveness, the previous scenario is repeated with the addition of a 1000-pps SYN flood attack. Table 2-7 summarizes the results of this study and shows that the non-zero delay and PLR in the attacked network do not significantly affect the results. Average P_{DS} is statistically the same in this case; the handshake delays both increase from the non-attack case (slightly less in the delayed case) and the data rates are affected similarly.

Delay(s)	PLR (%)	P_{DS}	$P_{CC\|CE}$	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
0	0	0.65	0.99	2.39	644,894	177,435
1	10	0.66	0.95	5.22	367	156

Table 2-7 Network PLR and delay with SYN flood (50 runs)

2.2.7 BANDWIDTH FLOOD ATTACK

The previous attacks overwhelm the TCP pending connection queue with TCP SYN packets, leaving legitimate clients with no available queue slots. A bandwidth flood attack seeks to create a bottleneck in the network that results in increased packet loss. Because of model limitations,[6] the link bandwidth is artificially decreased from 100 to 10 Mbps throughout the network and the router is configured to forward packets at link speed with a 500-packet queue at each interface. Two attackers are configured to send 1400-byte User Datagram Protocol (UDP) packets at various attack rates. By exceeding available network bandwidth, this attack forces the router to

place excess traffic destined for the victim network interface in a queue and drop packets that exceed the queue capacity. Table 2-8 shows initial flood attack results for flood rates between 0 and 30 Mbps.

[6] Model limitations (e.g., available memory, processing speed) preclude the analysis of the default network. However, these results will scale.

Flood Rate (Mbps)	P_{DS}	$P_{CC\|CE}$	Hand-shake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)	Server Packet Loss Ratio	Client Packet Loss Ratio
0	0.00	1.00	0.001	282,013	114,877	1.00	1.00
9	0.03	0.97	0.03	882	425	0.91	0.63
10	0.23	0.86	2.90	358	195	0.94	0.50
11	0.44	0.78	4.00	463	131	0.95	0.44
15	0.66	0.70	6.79	312	74	0.96	0.38
30	0.81	0.56	9.91	732	48	0.98	0.32

Table 2-8 Flood attack results (50 runs)

As shown in Table 2-8, network performance degrades once the attack rate approaches the maximum link bandwidth (10 Mbps). Figure 2-1 shows that the handshake delay increases by four orders of magnitude. The data rate (Figure 2-2) decreases by three orders of magnitude, and $P_{CC|CE}$ decreases (Figure 2-3), indicating degraded service for established connections. Figure 2-3 shows that the average P_{DS} rapidly increases as the attack approaches the link speed, reflecting the large number of interrupted or failed connections to the victim. Because the flood only causes the router to drop incoming packets to the server, the ratio of packets sent to packets received decreases for the clients but remains constant for the server (Figure 2-4). Interestingly, a flood rate of 15 Mbps (i.e., 1339 pps) results in an average P_{DS} nearly identical to the 1000-pps SYN flood attack discussed in Subsections 2.2.4 and 2.2.6 (0.657 vs. 0.625).

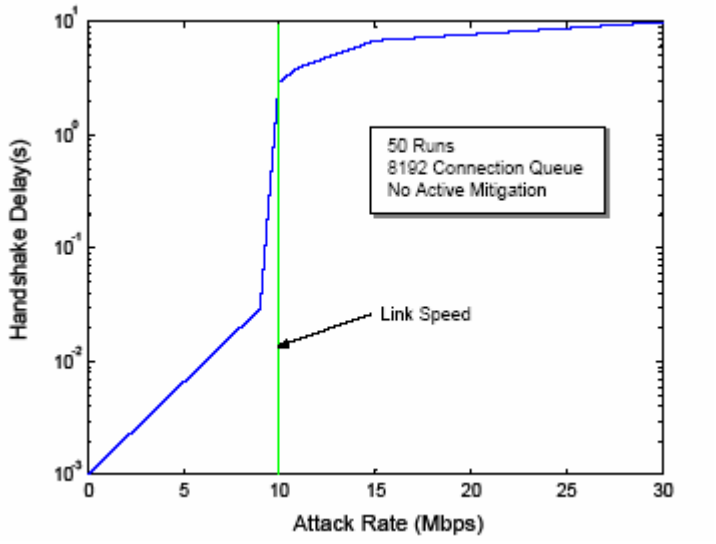

Figure 2-1 Handshake delay during flood attack

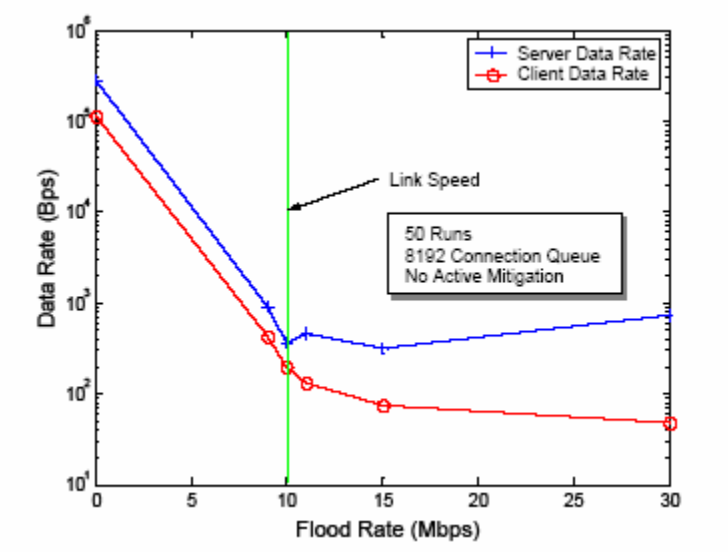

Figure 2-2 Data rate during flood attack

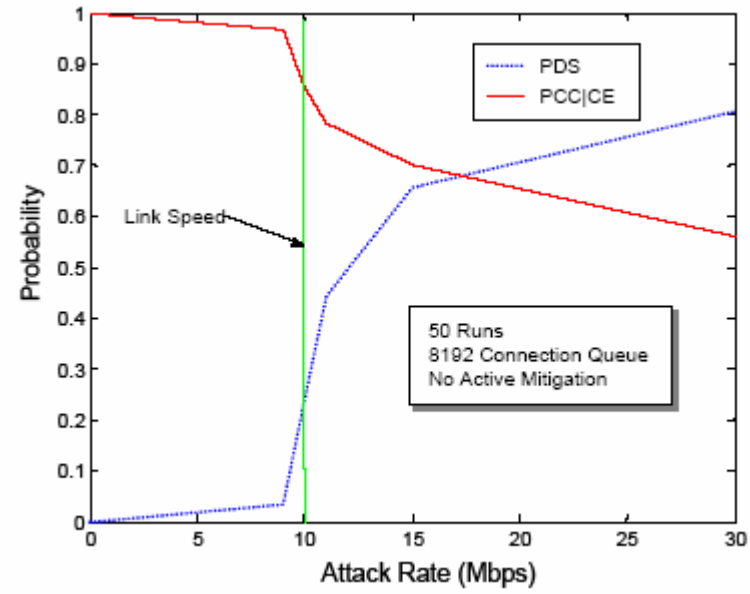

Figure 2-3 PDS and PCC/CE during flood attack

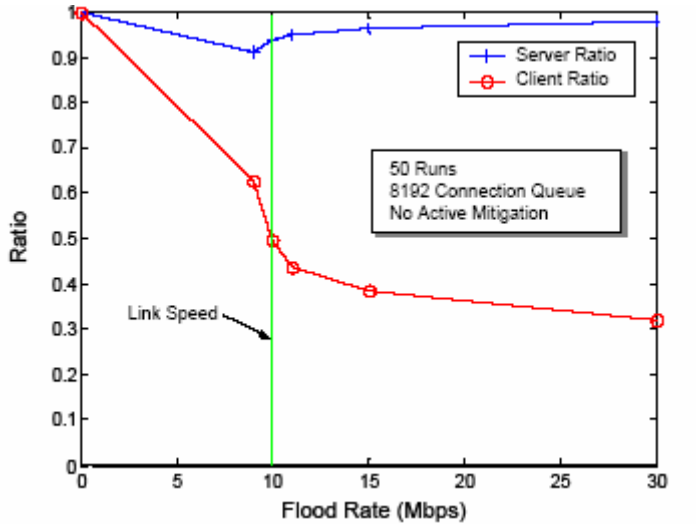

Figure 2-4 Ratio during flood attack

3.0 SYSTEM TUNING

Previous analysis (References 6 and 10) shows that networks can be tuned to better withstand an attack. To further explore this concept, the effects of TCP and network variations are analyzed. This section summarizes the effects of varying these parameters. As discussed in Subsection 1.4.3, the TCP SYN flood attack seeks to deny service by monopolizing TCP pending connection queue resources. The tunable parameters that drive attack performance are the time the attacker can hold a resource, the number and duration of requests a legitimate user makes for the resource, the amount of resources present, and the attack rate (Reference 10). The length of time a malicious entity can hold a resource is represented by the number of SYN-ACK retries the server sends before it resets a connection. The duration and number of requests a legitimate entity attempts to connect is represented by the number of SYN retries. Figure 3-1 graphically depicts these parameters. Sections 3.1, 3.2, and 3.3 describe the results of varying these parameters. Unless otherwise noted, these analyses use the original JHU/APL 500+ node network (i.e., the original 100-Mbps bandwidth allocations are used).

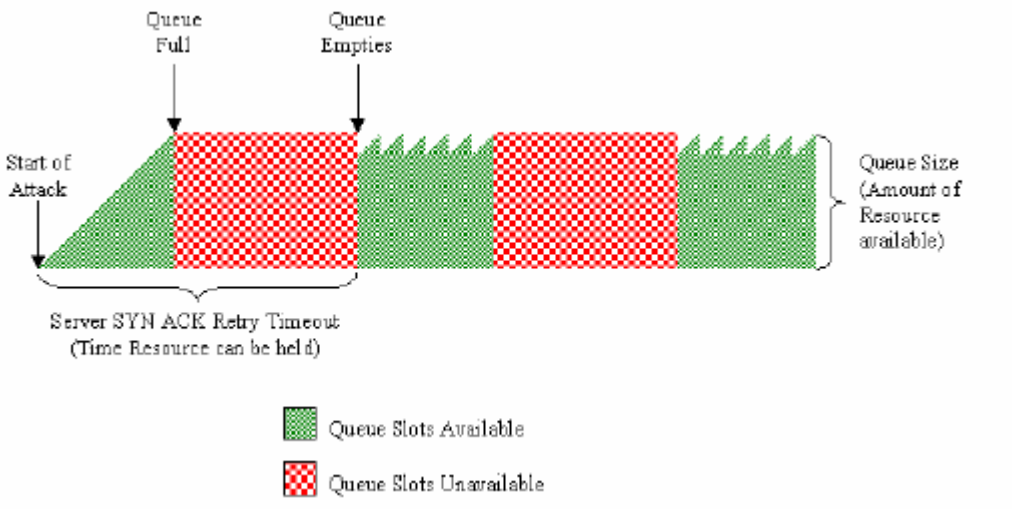

Figure 3-1 TCP Connection queue during an attack

3.1 VARYING THE NUMBER OF SYN RETRIES ATTEMPTED BY THE CLIENT

TCP uses a retransmission timer to ensure data deliver in the absence of any feedback from the data receiver. The duration of this timer is referred to as the RTO (Reference 15). When the retransmission timer expires, the next unacknowledged packet is retransmitted, the RTO (for normal parameter settings) is doubled, and the timer is restarted. Thus, lacking acknowledgements, the time between retries grows exponentially until the maximum RTO is exceeded, at which point the connection is reset. The maximum number of SYN retries a client attempts after its initial connection request determines the user retry duration. As clients make more requests over a longer period of time, it is expected that attack effectiveness will decrease because of the increased probability that a request will arrive when a connection queue slot is available. A TCP SYN flood is launched against the internal Web. The maximum number of TCP SYN retries attempted by the client was varied from one to seven, where the default value for this JHU/APL network is three. As shown in Table 3-1, varying the number of retries attempted by the client has a large effect on both the average PDS and handshake delay. As the number of retries increases, average P_{DS} decreases (Figure 3-2). Unfortunately, there is also a correspondingly large increase in handshake delay (Figure 3-3). This increase is because of the increasing maximum RTO, results in a larger average delay as additional retries are sent at exponentially increasing intervals. These results indicate that although this method can achieve a decrease in P_{DS}, it comes at the cost of increasing user delay.

SYN Retries	P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
1	0.78	0.20	670,647	302,348
2	0.74	0.66	693,926	241,466
3	0.65	2.39	644,894	177,435
4	0.42	5.34	609,095	174,506
5	0.22	10.16	568,398	169,296
6	0.18	12.36	569,500	172,659
7	0.09	22.98	567,431	175,799

Table 3-1 Variation of SYN retries (50 runs)

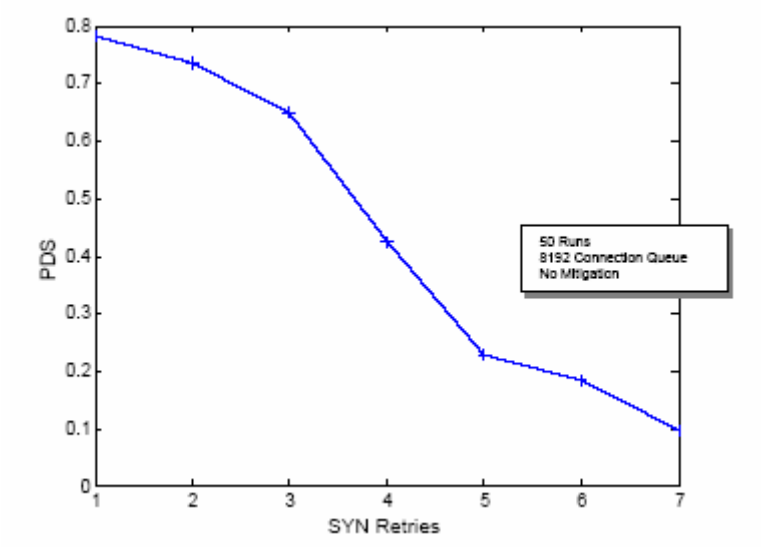

Figure 3-2 Average PDS with SYN retry variation (50 runs)

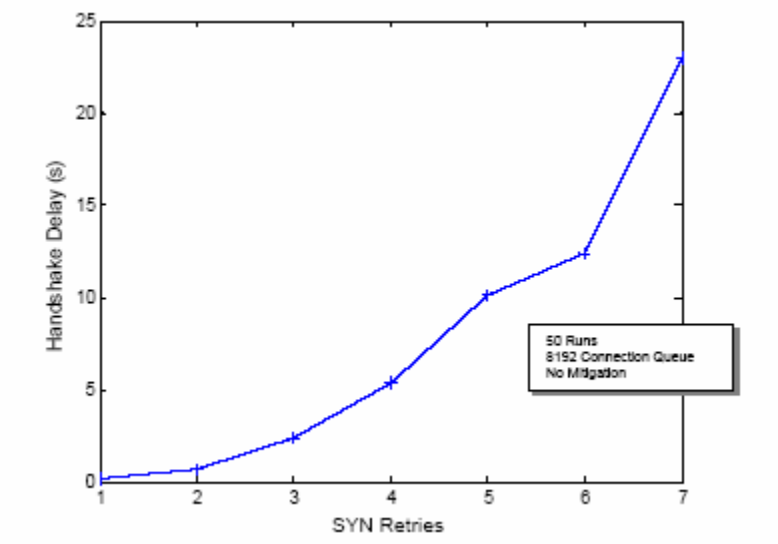

Figure 3-3 Average handshake delay with SYN retry variation (50 runs)

3.2 VARYING THE NUMBER OF SYN-ACK RETRIES ATTEMPTED BY THE SERVER

The maximum number of SYN-ACK retries a server attempts after it receives a connection request determines the amount of time a connection queue slot is held by an attacker. A TCP SYN flood is launched against the internal Web server. The maximum number of TCP SYN-ACK retries attempted by the server is varied from one to four, where the default value for this JHU/APL network is three. Table 3-2 shows that varying the number of SYN-ACK retries attempted by the server affects both the P_{DS} and the handshake delay. By decreasing the number of retries from the default value of three to one, the P_{DS} becomes 0 for this particular attack scenario, as shown in Figure 3-4. Minimizing the number of retries has a positive effect on the handshake delay as well, decreasing it substantially, as shown in Figure 3-5. For larger numbers

of retries, the time the queue is full is much longer than the time it takes to fill. Because the relative probability of getting through on any particular retry is about the same, the average delay is similar for the two-, three-, and four-retry cases. However, in the one-retry case, the time the queue is full is only 1 second longer than the time it takes to fill. In this case, the client rarely needs more than one retry to establish a connection, decreasing the average substantially. If network loading caused packets to be routinely dropped, having a single retry (i.e., two connection attempts) would likely result in denied service regardless of the presence of an attacker.

SYN-ACK Retries	P_{DS}	Handshake Delay(s)	Server Data Rate (Bps)	Client Data Rate (Bps)
1	0.00	0.18	670,064	286,956
2	0.29	2.22	667,234	185,695
3	0.65	2.39	644,894	177,435
4	0.83	2.35	691,886	175,613

Table 3-2 Variation of SYN/ACK retires (50 runs)

3.3 VARYING BOTH SYN AND SYN-ACK RETRIES

Sections 3.1 and 3.2 examined the impact of varying either the number of SYN-ACKs or number of SYNs while holding the other constant. This subsection considers the simultaneous variation of these two parameters. A TCP SYN flood is launched against the internal Web server. The number of TCP SYN retries attempted by the clients is varied from one to seven, where the default value is three. The number of TCP SYN-ACK retries attempted by the server is varied from one to four, where the default value is three. Table 3-3, Table 3-4, and Table 3-5 summarize the average P_{DS}, handshake delay, and server data rate, respectively. Table 3-3 shows that optimum (e.g., minimal denied service) availability is achieved by decreasing the number of SYN-ACK retries (i.e., reducing how long an attacker can hold a resource) and increasing the number of SYN retries (i.e., making the user more persistent). Table 3-4 demonstrates that minimizing either the SYN or SYN-ACK retries is important to minimize handshake duration for those systems receiving service. Finally, Table 3-5 shows that minimizing one type of retry also maximizes that server data rate. Taking these three results together shows that for this scenario, where network losses are not an issue, optimum performance (i.e., low average P_{DS}, low average handshake delay, and high average server data rate) is obtained by maximizing the number of user retries while minimizing the number of server retries

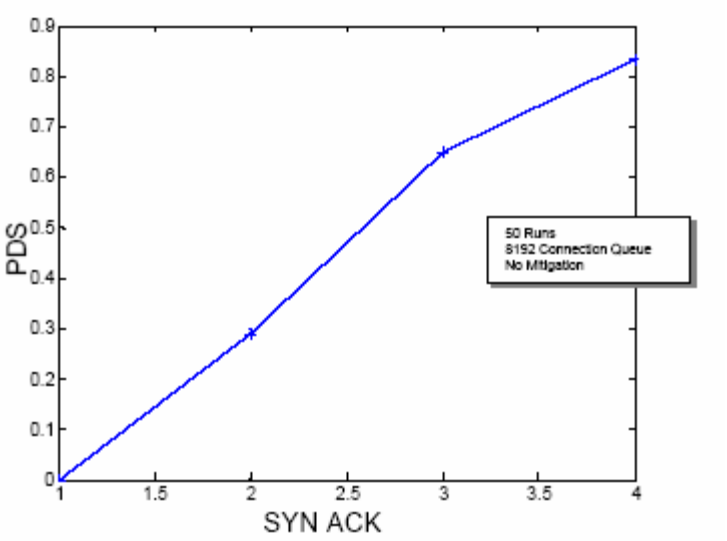

Figure 3-4 Average PDS with SYN-ACK retry variation (50 runs)

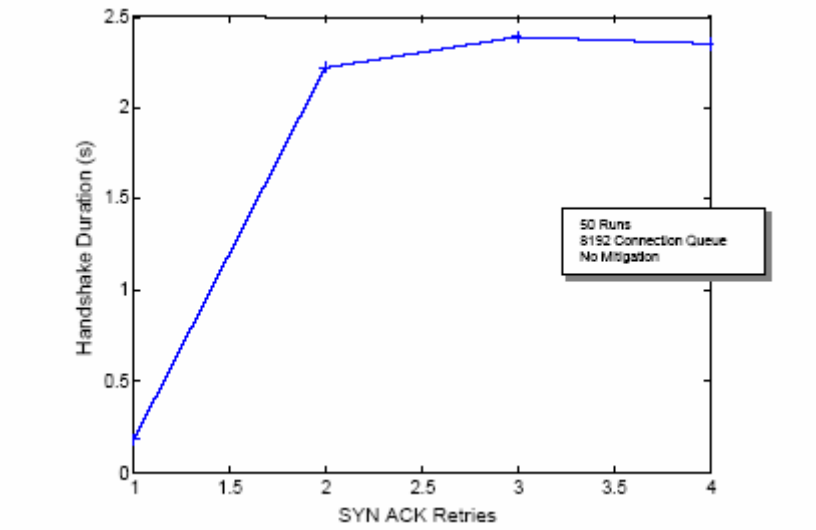

Figure 3-5 Average handshake delay with SYN-ACK variation (50 runs)

	Number of SYN Retries							
		1	2	3	4	5	6	7
Number of SYN-ACK retries	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	2	0.54	0.45	0.29	0.00	0.00	0.00	0.00
	3	0.78	0.74	0.66	0.42	0.22	0.18	0.09
	4	0.89	0.87	0.83	0.71	0.63	0.48	0.42

Table 3-3 Average PDS for number of SYN and SYN-ACK variation (50 runs)

		Number of SYN Retries						
		1	2	3	4	5	6	7
Number of SYN-ACK retries	1	0.18	0.17	0.18	0.18	0.18	0.18	0.18
	2	0.21	0.74	2.22	4.80	4.80	4.80	4.77
	3	0.20	0.66	2.39	5.34	10.16	12.36	22.98
	4	0.17	0.70	2.35	5.28	9.49	19.04	28.66

Table 3-4 Average handshake delay for number of SYN/ACK variation (50 runs)

		Number of SYN Retries						
		1	2	3	4	5	6	7
Number	1	670,312	671,964	670,064	672,760	669,661	670,064	668,780
of SYN-ACK retries	2	669,333	667,469	667,234	622,663	622,893	622,901	625,155
	3	670,647	693,926	644,894	609,095	568,398	569,500	567,431
	4	710,645	685,475	691,866	638,697	591,793	559,283	557,183

Table 3-5 Average server data rate for number of SYN and SYN-ACK variation (50 runs)

3.3.1 VARYING RTO MAXIMUM AND MULTIPLIER

The normal TCP configuration doubles the RTO after each retransmission. This means that the multiplier used to modify the RTO, the backoff multiplier, is normally set to 2.0. In previous sections, the number of retries is varied to modify both the amount of time a resource is be held by an attacker and the client retry duration. Alternatively, the backoff multiplier can be varied to modify the client retry duration and the distribution of retries over time. To investigate the effect of modifying retry timing, the number of SYN retries is set to 3, the default value in the previous sections, and the backoff multiplier is modified. Table 3-6 shows the maximum retry duration obtained by varying the backoff multiplier. The server is then subjected to a TCP SYN flood.

Backoff Multiplier	Retry Duration (sec)
2.0	7.0
1.0	3.0
0.5	1.8

Table 3-6 Maximum retry duration

Table 3-7 shows that decreasing the backoff multiplier increases average P_{DS}. The increase in average P_{DS} is due to the decrease in retry duration of legitimate users as the time between retries is decreased. As the retry duration decreases, retries have a lower probability of arriving when a slot is available in the connection queue. These results suggest that the number of retries is not as important as the duration over which retries occur in avoiding DoS.

Backoff Multiplier	P_{DS}	$P_{CC|CE}$	Handshake Delay (sec)
2.0	0.63	1.00	2.45
1.0	0.72	0.98	1.12
0.5	0.81	0.94	0.19

Table 3-7 RTO Max and Multiplier Variation (50 runs)

3.4 FLOODING ATTACKS

3.4.1 VARYING RTO MULTIPLIER DURING FLOOD ATTACK

In Subsection 3.3.1, the RTO multiplier is modified to make the TCP handshake more aggressive in the case of a TCP SYN flood. In that case, the modifications are not successful because they result in requests spaced more closely in time, decreasing the likelihood that one will arrive when a connection queue slot is open. With a bandwidth flood, the connection queue is not an issue because the server is not flooded with TCP SYN packets. Because a more aggressive TCP client essentially floods the network, the modification is repeated for a network subjected to a bandwidth flood attack. The bandwidth flood scenario uses an 11-Mbps flood rate, and the RTO multiplier is varied from 2.0 to 0.5. The results in Table 3-8 indicate that decreasing the RTO multiplier from 2 to 0.5 increases the average P_{DS}. In the case above where the multiplier is decreased with a TCP SYN flood, the increase in P_{DS} is primarily due to the decrease in P_{CE}, indicating a higher probability that the handshake fails to complete. However, in this case, the decrease in P_{DS} is primarily due to the decrease in $P_{CC|CE}$, indicating that the decreased multiplier is causing additional flooding on the network.

Backoff Multiplier	P_{DS}	$P_{CC|CE}$	Handshake Delay (sec)
2.0	0.43	0.78	4.00
1.0	0.65	0.52	2.13
0.5	0.99	0.02	1.56

Table 3-8 RTO multiplier variation (50 runs)

4.0 ONE ACTIVE MITIGATION TECHNOLOGY

This section presents analysis results for the D-WARD and NetBouncer mitigation technologies. The effectiveness of the baseline attack is determined for each mitigation technology and further analysis is conducted, as appropriate, to better understand attacker and mitigation technology interaction.

4.1 D-WARD

D-WARD is a DDOS mitigation technology deployed at the source router to collect data from network traffic, classify existing traffic flows and connections based on traffic characteristics (e.g., number of packets sent, number of packets received), and compute rate limits. D-WARD applies the calculated rate limits to restrict the volume of outgoing Bps based on their destination. By monitoring TCP, UDP, and Internet Control Message Protocol (ICMP) traffic, D-WARD classifies each connection[7] as good, transient, or bad. D-WARD further classifies each traffic flow, consisting of groups of connections to the same foreign host, as normal, suspicious, or attack.

Rate limits are placed on flows classified as attack or suspicious and applied to transient or bad connections within that flow. After the attack stops, rate limits are relaxed. Classifications and rate limits are recalculated at fixed intervals, defined by the observation interval attribute. For this analysis, D-WARD is deployed in the core router to regulate traffic leaving the client subnets. D-WARD configuration parameters are initialized to their default values (Reference 4). The observation interval, which determines how often flows and connections are reclassified, is set to 1 second; the maximum TCP ratio, the acceptable ratio of TCP packets sent to TCP packets received, is set to 3.0. A complete list of D-WARD configuration attributes is provided in Table 4-1. The router can be configured to perform outbound or inbound D-WARD monitoring. When outbound monitoring is enabled, one D-WARD module monitors, classifies, and rate limits all traffic between the router and the servers (i.e., the client subnets are policed in aggregate). When inbound monitoring is enabled, a collection of D-WARD modules monitors, classifies, and rate limits traffic between each subnet and the router independently. Figure 4-1 depicts this distinction.
[7] D-WARD defines a TCP or UDP connection as traffic between two host port pairs, and an ICMP connection as traffic between two hosts.

Attribute	Value
Classification Parameters	
Observation interval	1 sec
Start time	10 sec
Maximum TCP ratio	3.0
Maximum ICMP ratio	1.1
Maximum UDP connections per flow	100 connections
Minimum UDP packets per UDP connection	3 packets
Rate Limit Calculation Parameters	
Minimum rate limit	2000 Bps
Maximum rate limit	10000000 Bps
Decrease speed	2.0
Increase speed	2.0
Rate allowed	5000 Bps
Slow rate increment	500
Rate increment	500
Penalty period	20 observation intervals
Data Structure Maintenance Parameters	
Maximum flow records	10003 records
Maximum connection records	100003 records
Flow inactive period	360 sec
Good connection inactive period	360 sec
Transient connection inactive period	120 sec
Flow rehash load factor high	0.7
Connection rehash load factor high	0.7
Full alert load factor	0.9
Byte increment	2000 bytes

Table 4-1 D-Ward configuration attributes

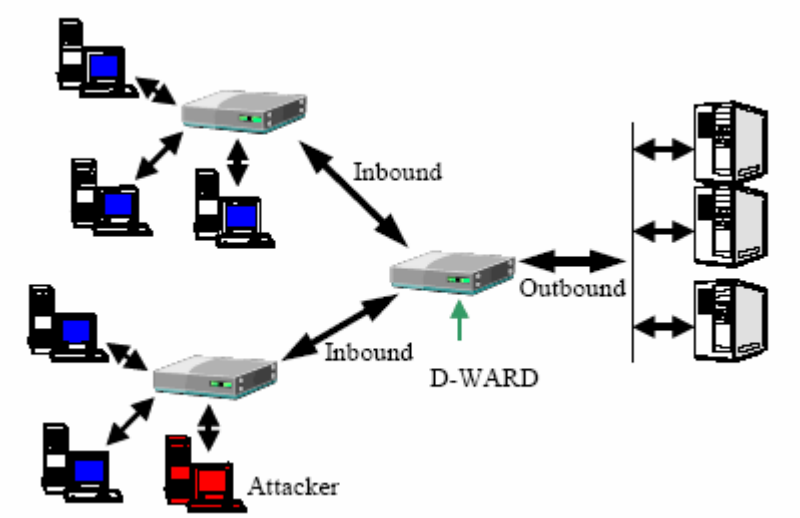

Figure 4-1 Outbound D-WARD vs inbound D-WARD

4.1.1 D-WARD (OUTBOUND MONITORING) AGAINST ONE ATTACKER

The source router is configured to perform outbound D-WARD monitoring, and for 1000 seconds a 1000-pps TCP SYN flood attack is launched against the internal Web server, which has an 8192 TCP pending connection queue. The attacker spoofs addresses within its Class C address space, which are within the D-WARD policed address set. Within 40 seconds of the onset of the attack, D-WARD classifies the flow as attack and begins rate limiting traffic to the internal Web server. An example of the rate limits calculated by D-WARD to combat this attack is shown in Figure 4-2. The rate limit is quickly constricted to 2000 Bps, the minimum rate limit value. As the situation stabilizes, the ratio of TCP packets sent to TCP packets received falls below the maximum TCP ratio. Traffic to the internal Web server is then classified as suspicious, rather than attack, and the rate limit is gradually relaxed to approximately 5000 Bps, the rate allowed parameter. As shown in Figure 4-3, the rate limit causes fewer attack packets to reach the server, which decreases the length of the pending connection queue from 8192 to approximately 1500 in 40 seconds. When D-WARD outbound monitoring is activated, average P_{DS} is 0.46, a negligible increase (ΔP_{DS} = +0.06) from the unmitigated attack. During the unmitigated attack, service is denied to legitimate clients because the connection queue at the internal Web server is full. When D-WARD outbound monitoring is enabled, service is denied during the first 40 seconds of the attack because the connection queue is full. However, after the first 40 seconds, D-WARD rate limits cause legitimate client SYN packets to the server to be dropped, resulting in DoS. The $P_{CC|CE}$ decreases from 1.0 in the unmitigated case to 0.98 when D-WARD outbound monitoring is enabled. This occurs because D-WARD rate limits drop data packets from legitimate connections, and if enough data loss occurs, the connection is reset.

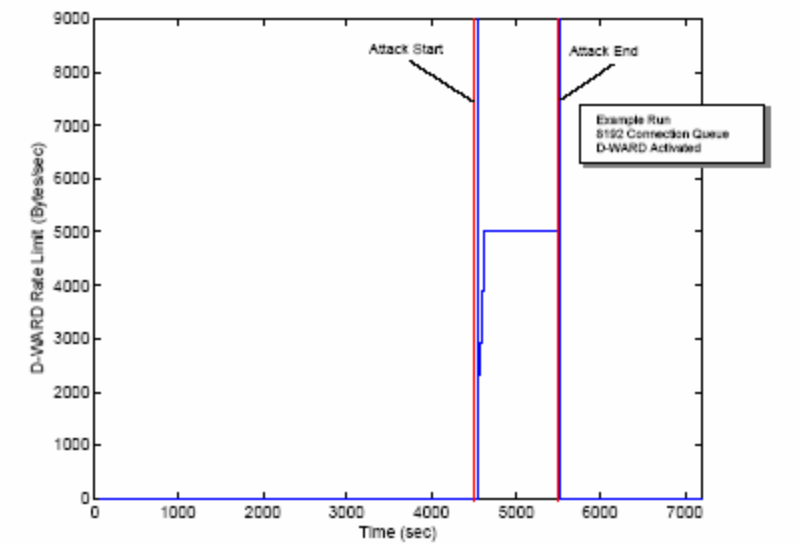

Figure 4-2 D-WARD rate limit enforced on flow to internal web server (single attacker against D-WARD outbound)

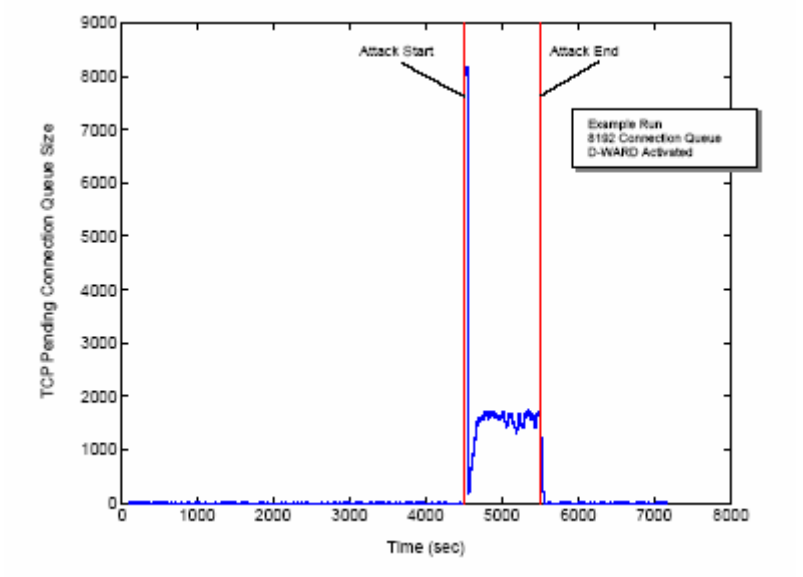

Figure 4-3 Victim server pending connection queue (single attacker vs D-WARD outbound)

Legitimate clients experience an average data rate of 29,000 Bps during the attack, decreased from 250,000 Bps when the attack is unmitigated. The average client data rate decreases because legitimate client data packets en route to the internal Web server are dropped by D-WARD rate limits.

4.1.2 D-WARD (INBOUND MONITORING) AGAINST ONE ATTACKER

The router is reconfigured to perform inbound D-WARD monitoring. While outbound D-WARD monitoring occurs on the internal Web server interface, inbound D-WARD monitoring occurs independently on each of the client subnet interfaces. When the 1000-pps attack is repeated, average P_{DS} drops to 0.25 (from 0.46), and average client data rate increases to

230,000 Bps, nearly the unmitigated attack value. This improvement occurs because only the subnet containing the attacker is subject to D-WARD rate limits. Because the remaining subnets contain no attackers, the D-WARD modules that police them perform no rate limiting. Average P_{DS} for the subnet containing the attacker is 0.42, while average P_{DS} for all other subnets is 0.01. Similarly, the average client data rate for the subnet including the attacker is 20,000 Bps, while the average client data rate for all other subnets is 350,000 Bps.

4.1.3 D-WARD (INBOUND MONITORING) AGAINST SEVEN ATTACKERS

Because D-WARD calculates rate limits based on observed traffic statistics, one potential attack adaptation is to distribute the attack. To examine the effect of a distributed attack, D-WARD is configured to perform inbound monitoring, and seven attackers are distributed throughout the network, each on a different subnet and thus policed by a different D-WARD module, as shown in Figure 4-4. Each attacker sends one packet every 0.007 second. The attack start times are staggered such that the cumulative effect at the server is a 1000-pps attack, with a constant packet inter-arrival time and the origin of each packet alternating between the seven attackers.

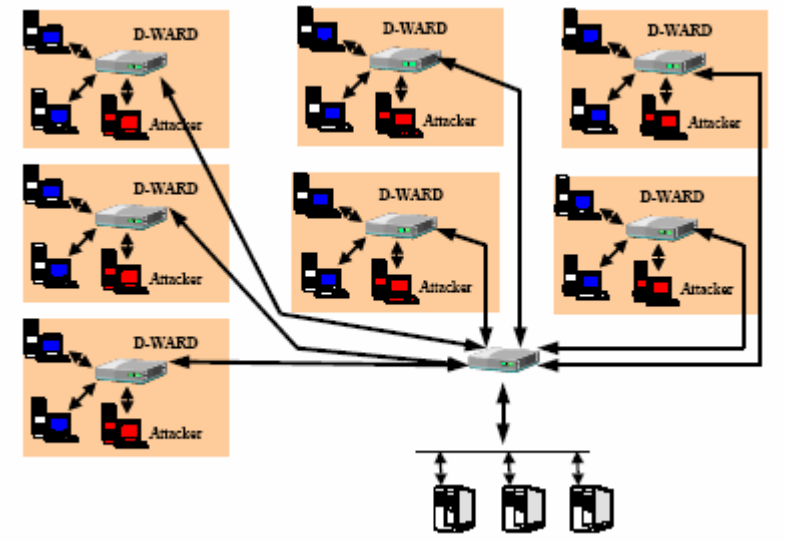

Figure 4-4 Seven attackers distributed throughout network

Distribution of the attack increases average P_{DS} from 0.25, in the one attacker against inbound monitoring case, to 0.49. Average client data rate decreases from 230,000 to 120,000 Bps. After approximately 20 seconds, the D-WARD modules detect the attack and restrict traffic to the internal Web server to the minimum rate limit. As these rate limits are applied, the number of attack SYNs that reach the server decreases. The server tries to empty its pending connection queue, and the number of SYN-ACKs from the server to D-WARD policed subnets increases. This causes the ratio of TCP packets sent to TCP packets received to fall below the maximum TCP ratio, and D-WARD gradually relaxes the rate limit to the rate allowed for suspicious flows. As the rate limit is relaxed, more attack SYN packets are passed and the pending connection queue is filled again, which causes the TCP ratio to increase and triggers detection of another attack. The result is a periodic restriction and relaxation of D-WARD rate limits, which causes similar oscillation in the length of the pending connection queue. Examples of the effect of this attack on D-WARD rate limits and the pending connection queue are shown in Figure 4-5 and Figure 4-6, respectively.

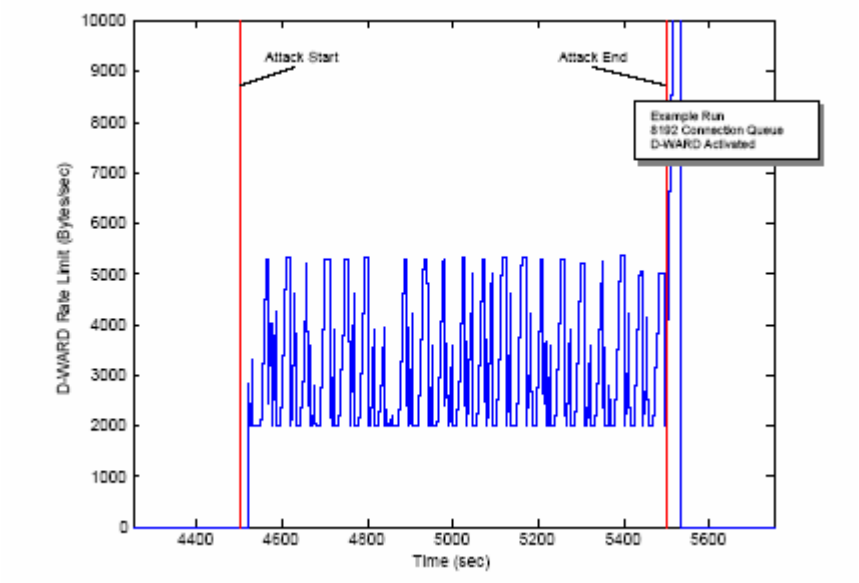

Figure 4-5 D-WARD Rate limit on traffic to internal web server from attacker 2 subnet (distributed attack against D-WARD inbound)

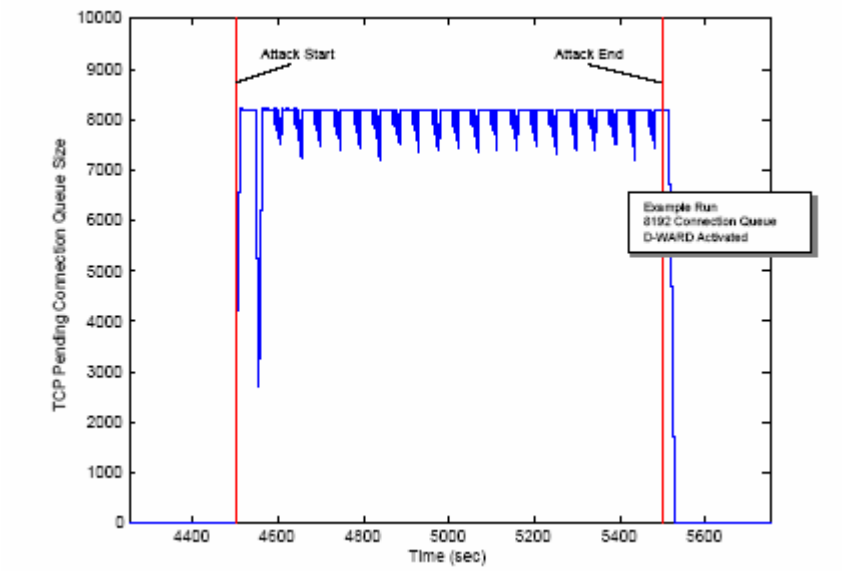

Figure 4-6 Victim server pending connection queue (distribute attack against D-WARD inbound)

4.1.4 VARYING MINIMUM RATE LIMIT PARAMETER (OUTBOUND MONITORING)

The minimum rate limit parameter determines the minimum rate to which an attacking flow can be restricted. To analyze the effect of the minimum rate limit parameter on average P_{DS}, D-WARD is subjected to the same single-attacker 1000-pps SYN flood with minimum rate limit values ranging from 500 to 5000 Bps. As shown in Figure 4-7, the minimum rate limit causes minimal variance in average P_{DS}. This occurs because soon after the attack begins and traffic to the internal Web server is rate limited, the ratio of TCP packets sent to TCP packets received decreases below the

maximum TCP ratio, and the flow is classified as suspicious. At this point, the rate limit is relaxed to approximately the rate allowed parameter, where it remains until the attack stops, as shown in Figure 4-2. Because this attack is classified as suspicious for most of its existence, variation of the minimum rate limit parameter has minimal impact on average P_{DS}.

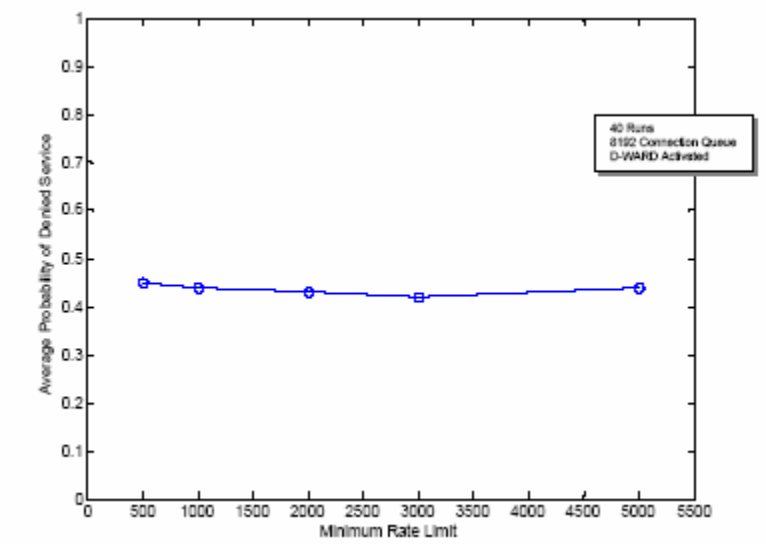

Figure 4-7 D-WARD minimum rate limit study (single attacker against D-WARD outbound)

4.1.5 VARYING RATE ALLOWED PARAMETER (OUTBOUND MONITORING)

The rate allowed parameter determines the maximum rate limit that can be applied to a suspicious flow. To determine the effect of the rate allowed parameter on average P_{DS} and average client data rate, the router is configured to perform outbound D-WARD monitoring with the rate allowed parameter varying from 3000 to 10000 Bps. The single-attacker 1000-pps SYN flood is repeated for each rate allowed value. As shown in Table 4-2, average P_{DS} decreases and average client data rate improves as the rate allowed value increases. Even though a relaxed rate increases the flow of attack packets to the internal Web server during the 950 seconds, the flow is classified as suspicious, causing P_{DS} to decrease and the average client data rate to improve. Fortunately, as shown in Figure 4-8, even at the increased highest allowed rate, the pending connection queue does not fill, allowing average P_{DS} to remain low.

Rate Allowed (Bps)	Average P_{DS}	Average Client Data Rate (Bps)
3000	0.49	19000
5000 (default)	0.46	29000
10000	0.39	43000

Table 4-2 Effect of rate allowed parameter on average PDS and average client data rate

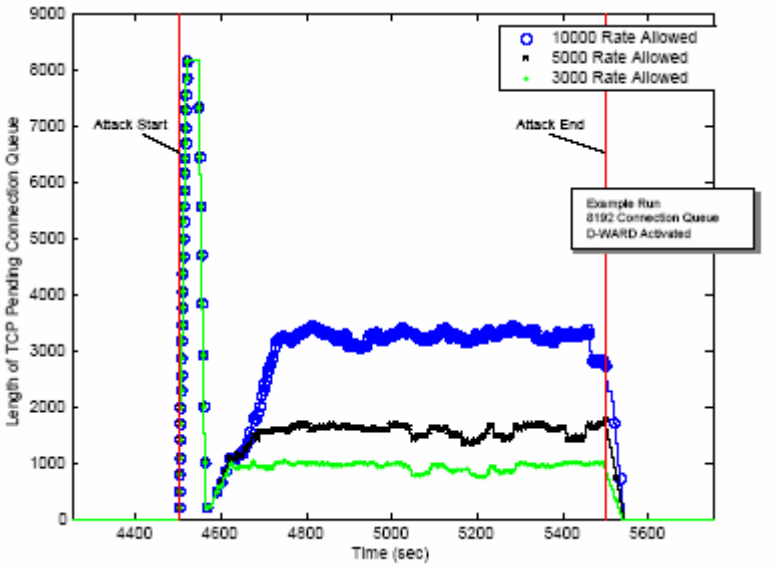

Figure 4-8 Victim server pending connection queue for each rate allowed value

Figure 4-9 displays the rate limits applied to the internal Web server traffic flow for each value of the rate allowed parameter. In each case, behavior for the first 50 seconds of attack is identical. The attack is detected 40 seconds after its onset, and then classified as attack for 10 seconds. Fifty seconds after the attack begins, it is classified as suspicious, and the rate limit is gradually increased to the rate allowed value. This causes the length of the connection queue to stabilize, as shown in Figure 4-8. Connection queue length stabilizes at a higher value if the rate allowed parameter is increased. It is important to note that if the rate allowed parameter is allowed to continue to increase, the result would be that at some point, the SYN flood is allowed to proceed normally and the connection queue is filled, resulting in DoS. On the other hand, lowering the rate allowed parameter results in increased denied service, as shown in Table 4-2. Tuning D-WARD parameters requires careful balance between these two concerns.

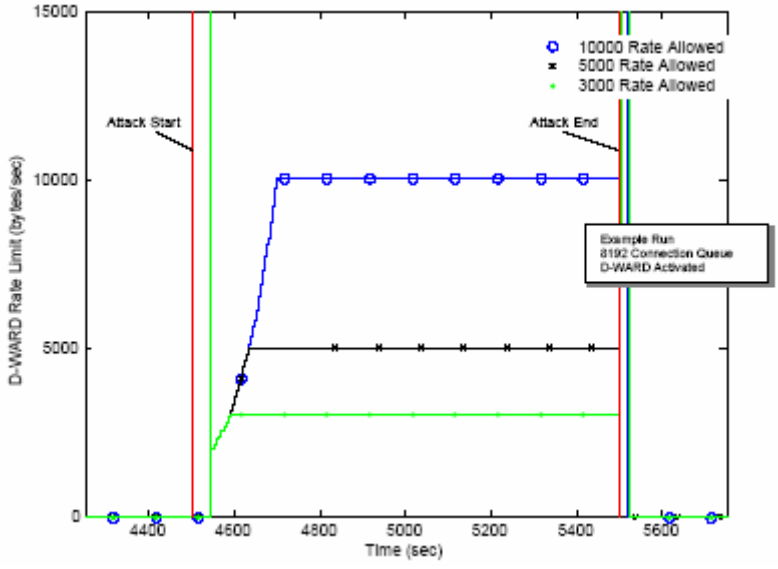

Figure 4-9 Rate limit enforced on flow to internal web server for each rate allowed value

4.1.6 VARYING CONNECTION LENGTH (OUTBOUND MONITORING)

As noted by its creators, D-WARD is optimized for long legitimate connections (Reference 16). To study the impact of long client connections on D-WARD performance during an attack, the amount of data downloaded from the internal Web server during each legitimate client connection is increased by a constant amount. The average connection length and average client data rate when D-WARD is enabled and no attack is present are shown in Table 4-3. Because there is no attack, D-WARD does not enforce rate limits on these legitimate client connections.

Data Downloaded During Connection	Average Connection Length (sec)	Average Client Data Rate (Bps)
Default	0.03	350,000
Additional 2.5 MB	3.08	64,000
Additional 5 MB	6.74	60,000

Table 4-3 Properties of legitimate client connections to internal web server in Absence of attack

The single 1000-pps attacker is enabled, and the router is configured to perform outbound D-WARD monitoring. During the attack, the client connection experiences an increase in connection length and a decrease in average data rate. However, as the volume of data transferred across the connection increases, the decreases in average client data rate become less significant.

Data Downloaded During Connection	Average Connection Length (sec)	Average Client Data Rate (Bps)
Default	5.50	29,000
Additional 2.5 MB	7.58	32,000
Additional 5 MB	11.8	39,000

Table 4-4 Properties of legitimate client connections to internal web servers during 1000 pps attack

Table 4-4 shows the connection length and average client data rate during the 1000-pps attack for each of the three volumes of data downloaded.

Figure 4-10 depicts the fraction of the average client data rate preserved when the attack occurs (i.e., the ratio of data rate during attack to data rate in absence of attack). As the volume of data transferred increases, the decreases in average client data rate are less significant. This occurs because as the behavior of attack and legitimate connections diverges, D-WARD is able to distinguish between them for better application of rate limits.

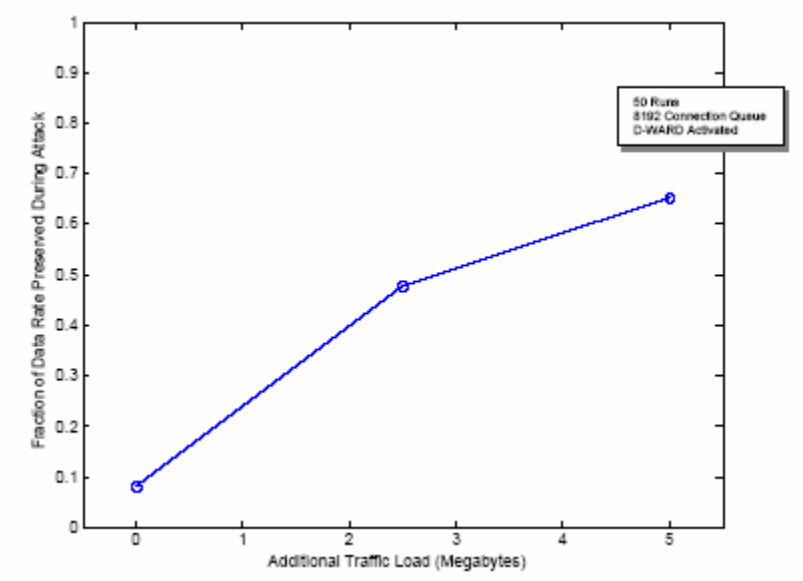

Figure 4-10 Fraction of average client data rate preserved during attack for various traffic loads (single attacker against D-WARD outbound)

4.1.7 D-WARD (OUTBOUND MONITORING) AGAINST BANDWIDTH FLOOD ATTACK

The attacks examined thus far have attempted to overwhelm a TCP pending connection queue with TCP SYN packets. A bandwidth flood attack seeks to create a bottleneck in the network, where legitimate clients must compete with a flood of attack traffic to pass through the bottleneck. By creating more than 10 Mbps of traffic across a 10-Mbps link, the attacker forces the router to place excess traffic in a queue and drop packets if the queue is full. In this case, the attacker is configured to send an 11-Mbps flood to the internal Web server and D-WARD is enabled. The attacker spoofs addresses within its Class C subnet, which is included in the D-WARD policed address set. Average P_{DS} is 0.51 when the attack is not mitigated (i.e., D-WARD is inactive). When D-WARD outbound monitoring is enabled, average P_{DS} increases to 0.63. This occurs because $P_{CC|CE}$ decreases from 0.66, in the unmitigated case, to 0.50, when D-WARD is enabled. P_{CE} remains constant at 0.75 in each case. In this case, the attacker is configured to send 1428-byte UDP packets (i.e., a 28-byte header with a 1400-byte payload). Because D-WARD calculates a rate limit of 2000 Bps, as shown in Figure 4-11, at most one attack packet will be forwarded each second. The remaining bandwidth for that second is used to transmit legitimate client traffic. If legitimate clients transmit more than 572 bytes before the attacker transmits a packet, all 2000 bytes allotted for that second will be available to legitimate clients. Thus, D-WARD rate limits drop a high percentage of attacker UDP packets and a low percentage of legitimate client SYN packets, resulting in an increased P_{CE}, and limiting the amount of attack traffic that reaches the bottleneck. $P_{CC|CE}$ decreases because D-WARD rate limits drop acknowledgment packets, which are large compared to SYN packets, en route to the internal Web server during data transmission. When enough packet loss occurs, the connection is reset. Strict D-WARD rate limits, coupled with packet loss caused by the bottleneck, cause $P_{CC|CE}$ to decrease. This suggests that rate-limiting technologies, in the face of a flooding attack, can cause additional packet loss, further degrading performance.

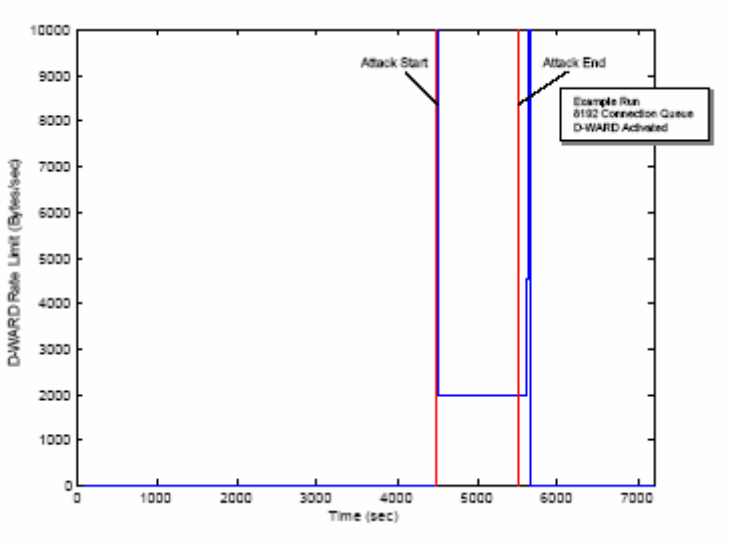

Figure 4-11 D-WARD rate limit enforced on flow to internal web server (bandwidth flood Attack against D-WARD outbound)

4.1.8 VARYING ATTACK PACKET SIZE DURING BANDWIDTH FLOOD ATTACK

One parameter under control of the attacker is the size of a flood packet. Because D-WARD determines whether to pass or drop a packet based on the amount of bandwidth the rate limit allots for each second, it may be advantageous to select different packet sizes. Table 4-5 shows the effect of varying UDP packet size during a bandwidth flood attack. Attack rate is held constant at approximately 60,000 pps.

Packet Size (bytes)	Average P_{DS}	Average $P_{CC|CE}$	Average P_{CE}
200	0.44	0.57	0.998
1428	0.63	0.50	0.75
2000	0.68	0.45	0.71

Table 4-5 Relationship between packet size and average PDS (50 runs) (bandwidth flood attack against D-WARD outbound)

If the packet size is increased to 2000 bytes, average P_{DS} slightly increases to 0.68 from 0.63. Average P_{CE} decreases from 0.75, in the baseline bandwidth flood attack, to 0.71, when attack packet size is increased. Simultaneously, $P_{CC|CE}$ decreases from 0.50 to 0.45. Because D-WARD enforces a 2000-bps rate limit on traffic to the internal Web server, the attacker is able to claim all the bandwidth each second with a single 2000-byte attack packet. If D-WARD receives an attack packet first, the attacker claims all bandwidth for that second; however, if D-WARD receives a client packet first, all attack packets received during that second will be dropped because they exceed the 2000-bps rate limit.

A 200-byte attack packet reduces average P_{DS} to 0.44 (ΔP_{DS} = -0.19). In this case, the average P_{CE} is 0.998 (ΔP_{CE} = +0.24), while the average $P_{CC|CE}$ increases slightly to 0.57. Because the attack packets have decreased in size, the legitimate client stands a better chance of sharing the available bandwidth with the attacker. D-WARD passes most legitimate client SYN packets, but still drops a percentage of acknowledgement packets during data transmission, which, when coupled with packet loss due to the bottleneck, accounts for the 0.56 $P_{CC|CE}$.

4.1.9 SUMMARY

D-WARD mitigates attacks by monitoring traffic flows and calculating custom rate limits for each attacking flow. While the D-WARD algorithm often decreases the length of the TCP pending connection queue at the server under attack, it does so at the expense of clients sharing the attacker's bandwidth. Deploying D-WARD closer to the source limits denied service, but may permit a low-rate distributed attack. Overall, as a situation-aware rate limiter, D-WARD provides benefits over situation-unaware rate limiters, such as Cisco's CAR analyzed in year one (Reference 6), because rate limiting is restricted to packets en route to destinations perceived to be under attack, and long-standing connections exhibiting good behavior are not rate limited. However, careful configuration of the system is necessary to ensure that D-WARD operates correctly with the network.

4.2 NETBOUNCER

NetBouncer is a mitigation technology from NAI that distinguishes between legitimate and illegitimate packets into a protected network (Reference 5). NetBouncer consists of a variety of legitimacy tests that can be applied to incoming packets. If a client passes the appropriate test or tests, it is added to a list of clients that have been proven to be legitimate. Incoming packets from clients on the legitimacy list are forwarded to their destination; otherwise, a challenge is initiated to give the client the opportunity to gain legitimacy. The NetBouncer model is incorporated into the DDOS-DATA network between the internal server's switch and the Cisco 7000 router (Figure 4-12). In this configuration, NetBouncer controls client access to the internal servers. The model incorporates three legitimacy tests proposed by NAI: Anti-Smurf, TCP SYN Cookie, and WWW Turing. The Anti-Smurf test provides a defense against a flood attack of ICMP echo replies. Because traffic models for the simulation network are TCP-based, NetBouncer analysis focuses on the effectiveness of the TCP SYN Cookie and WWW Turing tests.

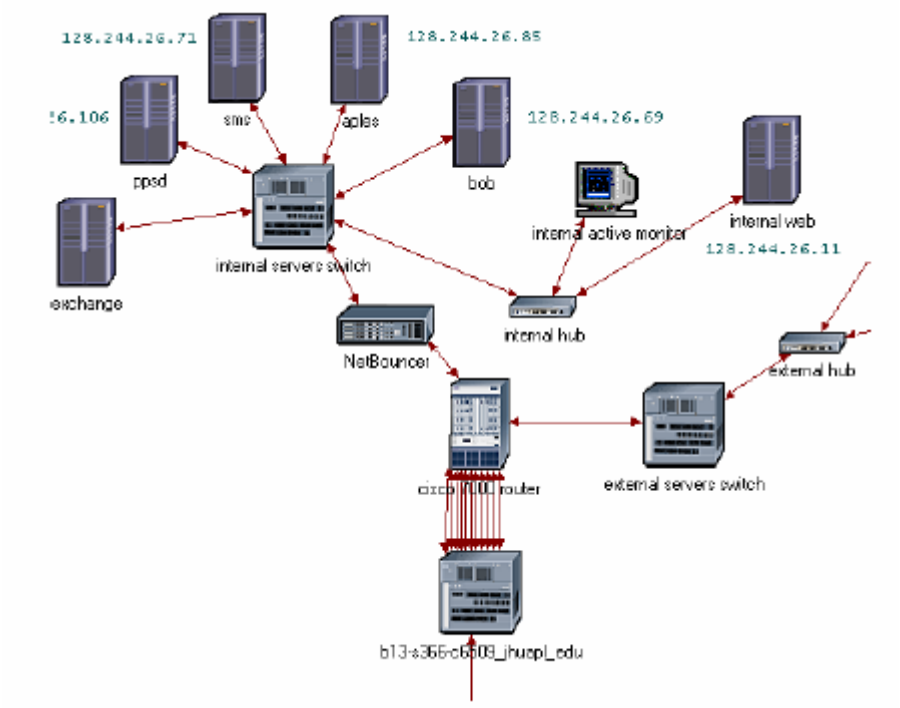

Figure 4-12 Netbouncer placement in DDos-DATA network

The TCP SYN Cookie test is a transport layer test that passes client packets to the protected network as long as NetBouncer is able to establish a connection with the client. When NetBouncer establishes the connection with the client, the client's IP address is placed on the legitimacy list so subsequent TCP connection requests are forwarded by NetBouncer. The TCP SYN Cookie test is invoked when NetBouncer receives a TCP SYN packet from a client that is not on the legitimacy list. After establishing the connection with the client, NetBouncer opens a TCP connection with the intended server. NetBouncer then acts as an intermediary between the client and server by translating sequence numbers of incoming packets to make the two connections seamless and forwarding these packets until the connection is closed. The WWW Turing test is an application layer test on Hypertext Transfer Protocol (HTTP) requests that requires the client to correctly solve a puzzle to be deemed legitimate. NetBouncer invokes the Turing test when it receives a TCP SYN packet with a destination port of 80 (indicating an HTTP application) from a client that is not on the legitimacy list. The WWW Turing test consists of two TCP connections between the client and NetBouncer. On the client's original HTTP request, NetBouncer responds with a puzzle for the client to solve. The client initiates a second TCP connection with its response to the puzzle. If the solution is correct, NetBouncer places the client's IP address on the legitimacy list and responds to the client with an HTTP refresh command, causing the originally requested page to be reloaded.

4.2.1 BASELINE ANALYSIS

When the attacker attempts the SYN flood attack against NetBouncer with the TCP SYN Cookie test enabled, NetBouncer successfully defends the protected network from the SYN flood attacker (i.e., average $P_{DS} = 0$). Figure 4-13 shows the number of network packets NetBouncer receives and forwards to the protected network. Packets received by NetBouncer are forwarded until the attack starts. At this point, the two curves deviate as attack packets entering NetBouncer are not able to complete the TCP SYN Cookie challenges. NetBouncer continues forwarding packets from legitimate clients as it rejects attack packets.

4.2.2 ATTACK MODIFICATION

To adapt to a TCP SYN Cookie test, the attacker can first generate an Octopus attack (i.e., complete the three-way handshake) to have NetBouncer add an IP address to the legitimacy list and then performs a SYN flood attack, using this IP address, on the protected internal Web server. When NetBouncer's TCP SYN Cookie test is subjected to this attack, the initial Octopus attack succeeds in causing NetBouncer to classify the attacker IP address as legitimate, allowing NetBouncer to forward subsequent SYN flood attack packets to the internal Web server. SYN flood packets use up server resources, resulting in DoS when legitimate clients try to access the server. P_{DS} for the NetBouncer TCP SYN Cookie multiple attack case is 0.52, which is lower than the P_{DS} (0.65) for the baseline SYN flood case without NetBouncer in the network. This result is because of a difference in how the SYN packets are generated in each case. For the NetBouncer case, the attack packets have the attacker's IP address and random port numbers, while the attack packets for the baseline case contains spoofed IP addresses and random port numbers. The attack space is smaller in the NetBouncer case, resulting in fewer packets accessing the internal Web server's connection queue and therefore fewer packets being dropped from legitimate clients.

4.2.3 WWW TURING TEST APPLICATION

Because it is a more sophisticated test and requires responses from the client, the WWW Turing test is able to protect the internal Web server from the phased attack. During the Octopus attack, NetBouncer establishes a connection with the attacker, but the WWW Turing test never receives the HTTP request it expects. As a result, the challenge remains active and the IP address is never placed on the legitimacy list. Because subsequent SYN flood attack packets arriving at NetBouncer are from a source not on the legitimacy list, NetBouncer does not forward these but responds with a SYN-ACK back to the attacker.

4.2.4 ADAPTATION TO THE WWW TURING TEST

Because the WWW Turning test defeats the previous attack, an adversary could create a Turing Aware attacker. This attacker is cognizant of the WWW Turing test and tries to gain legitimacy by correctly responding to NetBouncer's puzzle. The attack model includes a Turing Response Accuracy (TRA) attribute that indicates the accuracy at which the attacker correctly responds to a WWW Turing test puzzle. Once NetBouncer is compromised, the Turing Aware attacker begins a SYN flood attack on the internal Web server. The first set of model runs examines NetBouncer performance when the attacker is immediately able to solve the puzzle (TRA of 1.0). The second case examines performance when the attacker has a more difficult time correctly solving the puzzle (TRA of 0.02).

4.2.4.1 Immediate Puzzle Solution

If the WWW Turing test puzzle is very simple with a limited number of solutions, an attacker would be able to quickly solve the puzzle and compromise NetBouncer's ability to mitigate an attack. By setting the TRA attribute to 1.0, the attacker solves the Turing puzzle on its first attempt. Because NetBouncer immediately legitimizes the Turing Aware attacker's IP address, subsequent SYN flood attack packets are forwarded by NetBouncer (Figure 4-14), degrading network performance. Legitimate clients are competing with the attacker for resources, causing denied service at the internal Web server. The resultant P_{DS} of 0.51 is similar to the SYN Cookie/Octopus attacker average P_{DS} of 0.52.

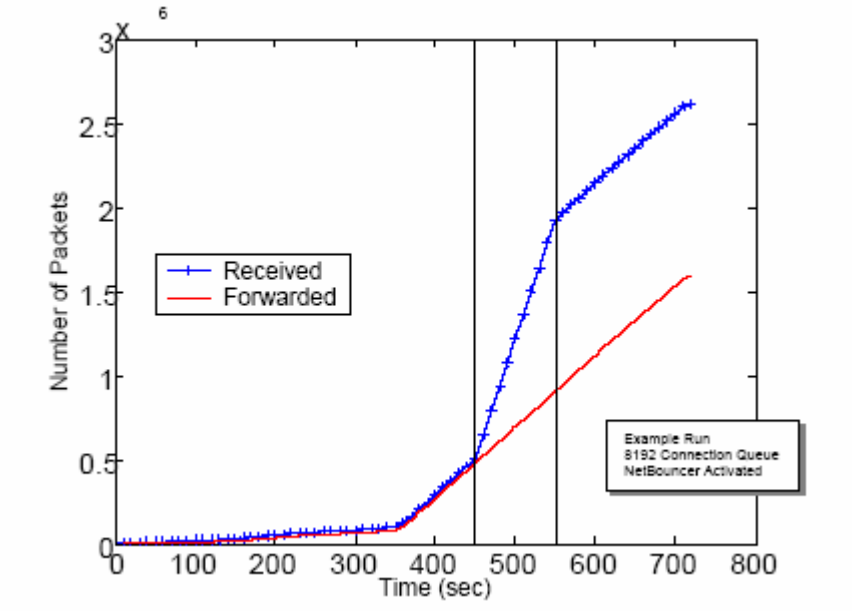

Figure 4-13 TCP SYN cookie test packet statistics – SYN flood attack (50 runs)

4.2.4.2 Delayed Puzzle Solution

By making the Turing puzzle more complicated with a greater number of possible solutions, NetBouncer can potentially delay or completely mitigate the effect of a Turing Aware attacker. Figure 4-15 shows the number of packets forwarded by NetBouncer, comparing the case in which the attacker immediately solves NetBouncer's puzzle (TRA = 1.0), to several runs in which the attacker has a lower probability (TRA = 0.02) of responding correctly to a NetBouncer puzzle. The resultant curves from various simulation runs show the effectiveness of NetBouncer to an attacker randomly trying to solve the Turing puzzle. The attacker attempts to break NetBouncer every 10 seconds until it successfully solves the Turing puzzle. Sometimes the attacker fails to solve the puzzle during the attack, as indicated by the run 12 curve. The run 22 and run 48 curves are examples where the Turing Aware attacker solves the puzzle during the attack, allowing NetBouncer to forward subsequent attack packets. Average P_{DS} is 0.24, smaller than the P_{DS} for the case in which the attacker is immediately successful in solving the Turing puzzle (average P_{DS} = 0.51), indicating that NetBouncer's WWW Turing test is more robust with a more complicated puzzle.

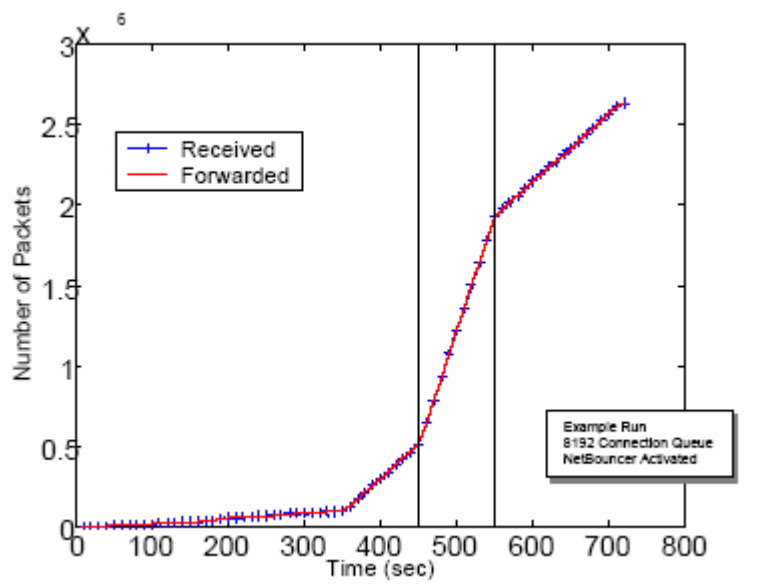

Figure 4-14 Turing test packet statistics – turing aware attack with TRA of 1.0

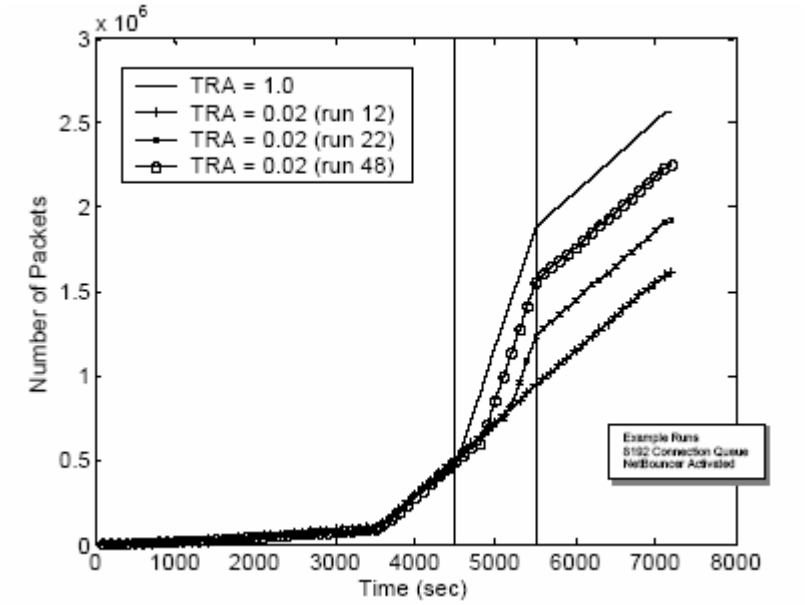

Figure 4-15 Packets forwarded by netbouncer – turing aware attack (example runs)

4.2.5 BANDWIDTH FLOOD ATTACK

One of the disadvantages of the WWW Turing test is that it requires more transactions than normal Web communications. Because flooding results in increased packet loss, examining whether performance decreases because of a flood is of interest. For this analysis, the network is configured as described in Subsection 2.2.7, with the WWW Turing test activated. Two sets of runs for different legitimacy list timeout values are analyzed. Table 4-6 shows the WWW Turing test results in the presence of a UDP flooding attack compared to the baseline results to the network flooding attack without NetBouncer. A legitimacy list entry is considered obsolete if

there is no activity from this address in the period specified by the legitimacy list timeout parameter. For longer timeout values, the WWW Turing test is less likely to be invoked because legitimacy list addresses are valid for a longer time period. During the attack period, average P_{DS} is greater if the WWW Turing test is invoked because it requires three successful TCP connections for a successful HTTP session. Network performance with the larger 3600-second legitimacy list timeout is similar to the baseline case without NetBouncer because the WWW Turing test is invoked infrequently during the attack. For the case with the 600-second legitimacy list timeout, entries to the legitimacy list are removed more quickly, causing the WWW Turing test to be invoked more often during the attack period. As indicated by Table 4-6, P_{CE} and $P_{CC|CE}$ are smaller, resulting in a larger P_{DS} for the 600-second legitimacy list timeout.

	Average P_{CE}	Average $P_{CC|CE}$	Average P_{DS}
Baseline without NetBouncer	0.725	0.782	0.433
WWW Turing Test (legitimacy list timeout = 3600 sec)	0.713	0.770	0.451
WWW Turing Test (legitimacy list timeout = 600 sec)	0.650	0.678	0.559

Table 4-6 Netbouncer turing test performance – UDP flooding attack (50 runs)

4.2.6 SUMMARY

The TCP SYN Cookie test is able to recognize and protect against a SYN flood attack on a targeted server. However, this test can be adapted to by using an Octopus attack to compromise NetBouncer. The WWW Turing test model successfully rejects the Octopus/SYN flood attack. For the Turing Aware attack scenario, the WWW Turing test performance is dependent on the complexity of the puzzle and the ability of the attacker to correctly solve the puzzle in a timely manner. NetBouncer proves ineffective if the puzzle is easily solved, allowing subsequent SYN flood packets to reach their target. Mitigation technologies such as NetBouncer seek to protect a network by performing tests on traffic. However, attackers can, and will, adapt to such tests. The results presented previously suggest that the only way to successfully use such an approach is to make the test sufficiently unpredictable that an attacker cannot automate a response to them. If a human-in-the loop is truly required, a DDOS attack can be mitigated.

5.0 COMBINED MITIGATION TECHNOLOGIES

This section presents analysis results for four scenarios where mitigation technologies are combined. Year-one analysis examined the Rate Limiter/Active Monitor and Active Monitor/ Proof-of-Work mitigation combinations. (Appendix B provides more detailed summary tables extracted from Reference 6). Year-two analysis examines D-WARD combined with server tuning, Proof-of-Work, Active Monitor, and finally NetBouncer. Attack effectiveness relative to each combination is analyzed and further analysis is conducted, as appropriate, to better understand attacker and mitigation technology combination interaction. The analysis also examines mitigation effectiveness by computing DI (defined in Subsection 1.4.4) for each mitigation combination.

5.1 D-WARD (INBOUND MONITORING) AND SERVER TUNING AGAINST A DISTRIBUTED ATTACK

D-WARD detects a TCP SYN flood based on the smoothed ratio of TCP packets sent to TCP packets received. In accordance with the protocol, the server sends, at most, four SYN-ACK packets while attempting to establish a connection for each entry in its pending connection queue. If the maximum number of server SYN-ACK packets is decreased, this will affect the TCP packet ratio. To study the effect of varying the number of server retries during a TCP SYN flood, seven distributed attackers are enabled, and the maximum number of server SYN-ACK packets is decreased from four to three. The attackers spoof addresses within their Class C subnets, which are included in the D-WARD policed address set. When the distributed attack is launched against the tuned server without D-WARD, each attack packet holds its slot in the pending connection queue for a shorter period of time. This causes average P_{DS} to decrease to 0.22 and average client data rate to stabilize at 82,000 Bps because the server can accept more SYN packets over the course of the attack. As discussed in Subsection 4.1.3, the distributed attack against inbound D-WARD monitoring results in an average P_{DS} of 0.49 and an average client data rate of 120,000 Bps. By enabling inbound D-WARD monitoring and tuning the server, average P_{DS} drops to 0.04 and average client data rate stabilizes at 93,000 Bps. This results in a DI of 0.18. During combined mitigation, the server removes connection attempts from its pending connection queue after three failed attempts to complete the connection, instead of four, which causes slots in the pending connection queue to be freed more quickly. While combined mitigation does not prevent the connection queue from filling, server tuning and inbound D-WARD monitoring cooperate to provide empty slots in the pending connection queue more frequently than in either of the single mitigation cases. During the single mitigation scenarios, the connection queue is full for more than 15 seconds at a time; however, during the combined mitigation case, queue length stabilizes at capacity for less than 5 seconds at a time before several hundred slots are freed. Because legitimate clients transmit up to four SYN packets over several seconds while attempting to establish a connection, it is more likely that one of those packets will be admitted to the pending connection queue in the combined mitigation scenario. P_{CE} is 0.97 in the combined mitigation case, an increase from 0.78 in the server tuning only case, and 0.52 in the inbound D-WARD only scenario.

5.2 D-WARD (OUTBOUND MONITORING) AND PROOF-OF-WORK

Proof-of-Work protocols attempt to mitigate DoS attacks by requiring payment for service. The implementation analyzed here (Reference 3) uses central processing unit (CPU) time as the payment method. If the server determines an attack is underway, clients are required to solve a cryptologic puzzle before the TCP handshake can occur. When the Proof-of-Work protocol is enabled, a client must request and receive a cryptologic puzzle via UDP. The client returns the solved puzzle, and upon verification of the solution, the TCP handshake proceeds. In this analysis, each puzzle requires 0.45 second to solve. To test the synergy of D-WARD outbound monitoring and Proof-of-Work, 450 distributed attackers are enabled, resulting in a net attack rate of 1000 pps. D-WARD recognizes the attack after 1 second and restricts traffic to the internal Web server to the minimum rate limit, 2000 Bps. During the attack, both attacker and client puzzle requested UDP packets are dropped because of D-WARD rate limits. The average probability that the UDP Proof-of-Work packet from a legitimate client will be dropped before it reaches the internal Web server is 0.97. This results in an average P_{DS} of 0.98. However, these rate limits are applied only to packets transmitted to the internal Web server. Average P_{DS} for a legitimate client connecting to the external Web server is 0. Figure 5-1 shows an example of this attack's effect on the internal Web server pending connection queue. D-WARD and Proof-of-Work recognize the attack immediately, preventing the queue from filling during the attack. Unfortunately, the loss of UDP packets results in DoS.

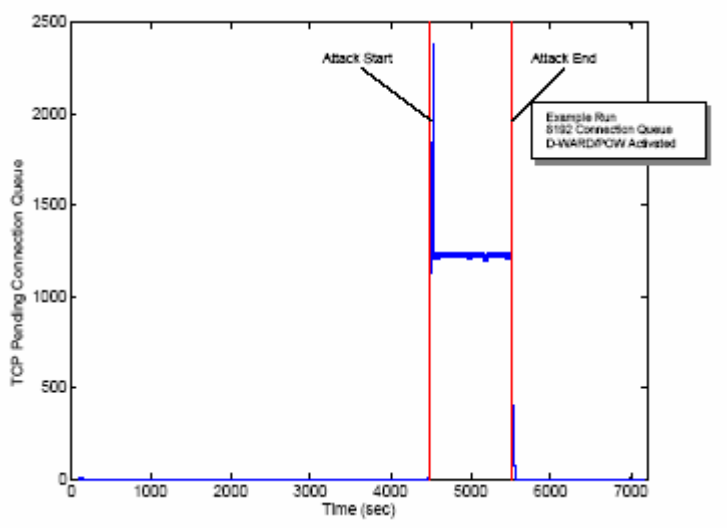

Figure 5-1 Victim server pending connection queue (single attacker against D-WARD and Proof-of-work

5.3 D-WARD (OUTBOUND MONITORING) AND ACTIVE MONITOR

Active Monitor (Reference 1) seeks to decrease the time an attacker can hold server resources by classifying each host as GOOD or BAD, based on observed traffic. Traffic from GOOD hosts (i.e., hosts that successfully complete a TCP three-way handshake) is allowed to proceed normally. However, connections from BAD hosts (i.e., hosts that fail to complete the TCP three-way handshake) will be reset. By resetting connections, Active Monitor frees entries in the victim server's pending connection queue. The interaction between D-WARD outbound monitoring and Active Monitor is tested by enabling the 1000-pps attacker. The attacker spoofs addresses within its Class C address space, which is in the D-WARD policed address set. When Active Monitor alone defends against this attack, average P_{DS} is 0.41 and average client data rate is 240,000 Bps. As discussed in Subsection 4.1.1, when D-WARD combats this attack, average P_{DS} is 0.46 and average client data rate is 29,000 Bps. When both mitigation technologies are enabled, average client data rate stabilizes at 34,000 Bps, and average P_{DS} is 0.45. In the combined mitigation case, D-WARD detects an attack and begins rate limiting identical to the D-WARD only case discussed in Subsection 4.1.1. Active Monitor observes connections close to the internal Web server after D-WARD rate limits are applied. Legitimate clients may be misclassified by Active Monitor during the first 40 seconds of the attack, when D-WARD has not constricted its rate limits and the pending connection queue is full, as shown in Figure 5-2. An examination of Tcpdump data collected during simulations shows that misclassification affects connection establishment infrequently throughout the remainder of the attack. It is possible for D-WARD to sporadically drop legitimate client ACK packets transmitted to the internal Web server, denying connection establishment. Active Monitor misclassifies these clients as BAD because the absence of an ACK packet indicates the handshake has failed to complete. This does not make average P_{DS} worse than the D-WARD only case or cause denied service after the attack ends. In fact, the presence of D-WARD rate limits in the combined mitigation scenario dominates the Active Monitor algorithm and causes average P_{DS} and client data rate to behave much like the D-WARD only case. Average P_{DS} in the combined case is 0.45, similar to 0.46 encountered during D-WARD only mitigation, while average client data rate is 34,000 Bps, similar to 29,000 Bps occurring during D-WARD mitigation.

5.4 D-WARD (OUTBOUND MONITORING) AND NETBOUNCER AGAINST A SINGLE ATTACKER

To examine the interaction between D-WARD and NetBouncer, a single 1000-pps attacker is enabled to send a SYN flood to the internal Web server. The attacker spoofs addresses within its Class C subnet, which is included in the D-WARD policed address set. NetBouncer is enabled to perform the TCP SYN Cookie test. When this attack is launched against NetBouncer alone, average P_{DS} is 0. NetBouncer is able to differentiate between legitimate and attack traffic through use of the TCP SYN Cookie test. As described in Subsection 4.1.1, when D-WARD defends against this attack, average P_{DS} is 0.46.

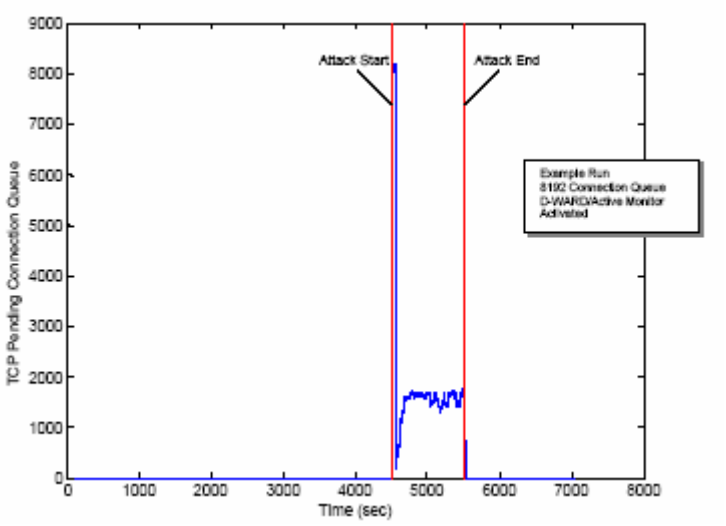

Figure 5-2 Victim server pending connection queue length (single attacker against D-WARD and active monitor.

Average P_{DS} is 0 when both technologies are enabled. This occurs because for each attack SYN packet, NetBouncer replies with a SYN-ACK packet in an effort to complete the connection for the TCP SYN Cookie test. This causes the ratio of TCP packets sent to TCP packets received to remain close to one, and thus D-WARD does not detect an attack or enforce rate limits. Because the NetBouncer TCP SYN Cookie test does not receive SYN-ACK packet responses, the attack SYN packets are not forwarded to the protected network by NetBouncer. This represents an improvement over the Active Monitor/Rate Limiter mitigation combination explored in Reference 6. Because D-WARD monitors both incoming and outgoing traffic at the policed network router, it does not classify the well-balanced traffic created by NetBouncer's TCP SYN Cookie test as attack. This situation awareness allows D-WARD to make informed decisions regarding traffic classification and rate limiting, resulting in an average P_{DS} of 0.

5.5 D-WARD (OUTBOUND MONITORING) AND NETBOUNCER AGAINST AN OCTOPUS ATTACK

To further test the synergy between D-WARD and NetBouncer, an attacker is enabled to complete one TCP handshake (i.e., Octopus attack) and then send a flood of TCP SYN packets from its own address. The attacker address is included in the D-WARD policed address set, and NetBouncer is enabled to perform the TCP SYN Cookie test. An attack of this type launched against NetBouncer results in an average P_{DS} of 0.52. D-WARD handles this attack similar to the SYN flood described in Subsection 4.1.1, resulting in an average P_{DS} of 0.53. Average P_{DS} is 0.52 when both technologies are enabled. Because the attacker's address is on the legitimacy list, NetBouncer allows all attack SYN packets to proceed to the internal Web server. This causes the

pending connection queue at the server to fill, and the DoS that results prompts the ratio of TCP packets sent to TCP packets received to increase. As the ratio passes the maximum value permitted, D-WARD detects the attack and rate limits traffic to the internal Web server. Although NetBouncer is unable to detect this attack, D-WARD recognizes the attack and enforces rate limiting on all traffic to the server.

6.0 CONCLUSIONS

The deployment of multiple mitigation technologies simultaneously can improve overall mitigation performance or introduce additional vulnerabilities. Combined mitigation performance hinges on the algorithmic details of the specific mitigation technologies selected, rather than the general classes to which their algorithms belong. For example, while some combinations of rate limiting and classification algorithms eliminate an attack (e.g., Section 5.4, D-WARD and NetBouncer), other such combinations do not decrease attack effectiveness (e.g., Section 5.3, D-WARD and Active Monitor). Thus, combined mitigation performance is determined by the details of algorithmic interaction between mitigation technologies, and may improve or inhibit overall mitigation performance.

DDOS-DATA has analyzed DDOS attacks and mitigation technologies to develop a solid understanding of the fundamental relationships between them. This understanding is necessary to determine the ability of mitigation technologies to address the DDOS problem and to understand how they can be successfully deployed together. To develop this understanding, JHU/APL has developed a systems analysis approach that uses M&S to develop quantitative metrics of attack, mitigation technology, and network performance. Such metrics are needed to develop a methodology for rigorously comparing and assessing information assurance systems. The conclusions for this analysis are as follows:

- By taking advantage of the numerous tunable system parameters, a system can be better positioned to defend itself from an attack. For example, maximizing the number of client retries while minimizing the time the server holds resources best defeated a SYN flood.

- The analysis of two rate limiting schemes, Cisco's CAR and D-WARD, quantitatively demonstrates that rate limiting is most effective if deployed as near to the attack source as possible. Not only does this architecture throttle the attack before it reaches the target network, it also minimizes the collateral damage caused by the rate limiter. Furthermore, smarter schemes, such as D-WARD, are more effective because they can make better decisions.

- Technologies that force users to perform some work or pass a test to restrict their ability to obtain resources will be defeated if the test response is easily automated or distributed. However, if difficult nondeterministic aspects can be added to a test, the ability of an attacker to adapt will be limited. The development of such tests will benefit from the application of formal techniques to ensure that an attacker cannot exploit the test protocols.

- Combining mitigation technologies must be done with caution. In several cases, mitigation technologies interfered with each other, causing an attack to become more effective. It is noteworthy that conclusions with regard to interference do not necessarily hold true across mitigation technology classes. For example, NetBouncer and D-WARD together reduced P_{DS} while Active Monitor and D-WARD together showed little improvement over the individual attacks.

Finally, it is important to note that it is typically straightforward to modify attackers to either bypass or exploit the mitigation technology. If the information assurance community is to successfully develop attack countermeasures, it is imperative that they consider the appropriate threat model. That is an adversary that will quickly adapt to mitigation technologies by developing new attacks, not one that simply relies on known attacks.

APPENDIX A - LIST OF REFERENCES

1. C. L. Schuba, et al., “Analysis of Denial of Service Attack on TCP,” IEEE, 1997.

2. Committed Access Rate, http://www.cisco.com/warp/public/732/Tech/car/index.html, 13 September 2001.

3. A. Juels and J. Brainard, “Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks,” RSA Laboratories.

4. J. Mirkovic, G. Prier, and P. Reiher, “Attacking DDOS at the Source,” University of California Los Angeles.

5. E. O’Brien and R. Thomas, “NetBouncer: A Practical Client-legitimacy-based DDoS Defense via Ingress Filtering,” September 2002.

6. D. M. Gregg, W. J. Blackert, R. M. Jokerst, E. M. Kyle, R. L. Hom, and A. K. Castner, “Distributed Denial of Service Defense Attack Tradeoff Analysis Report,” JHU/APL Report VS-02-129, February 2003.

7. D. M. Gregg, W. J. Blackert, E. M. Kyle, and R. M. Jokerst, “Distributed Denial of Service Defense Attack Tradeoff Analysis Verification and Validation Report,” JHU/APL Report VS-02-014, September 2002.

8. W. J. Blackert, A. K. Castner, R. L. Hom, E. M. Kyle, and R. M. Jokerst, “Distributed Denial of Service Defense Attack Tradeoff Analysis Year 2 Verification and Validation Report,” JHU/APL Report VS-03-068, September 2003.

9. “CERT/CC Overview Incident and Vulnerability Trends,” CERT Coordination Center, Pittsburgh, PA, 2002.

10. D. M. Gregg, W. J. Blackert, D. C. Furnage, and D. J. Heinbuch, “Denial of Service Attack Assessment Analysis Report,” JHU/APL Report VS-01-071, July 2001.

11. D. Comer, Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture, Prentice Hall, New Jersey, 1991, p. 62.

12. R. Cáceres and S. Floyd, “Measurement Studies of End-to-End Congestion Control in the Internet”, http://www.icir.org/floyd/ccmeasure.html.

13. Internet Traffic Report, http://www.internettrafficreport.com/main.htm.

14. The PingER Project, http://www-iepm.slac.stanford.edu/pinger

15. V. Paxson and M. Allman, “Computing TCP’s Retransmission Timer,” RFC 2988, November 2000.

16. Conversation with Jelena Mirkovic, April 2003.

APPENDIX B - SUMMARY OF PREVIOUS RESULTS

Technology	Objective	System Performance	Attacker Restrictions
Active Monitor	Classify nodes based on observed traffic. Reset connections from nodes deemed BAD to free resources.	P_{DS} decreases when compared to the no mitigation technology case, but misclassified nodes may be temporarily denied service. System will correct misclassification upon observing valid traffic. Tuning of expire timeout can be used to further lower P_{DS}.	Attacker can use a large number of IP addresses to overcome classification. Attacker can deny service with fewer packets by exploiting misclassification.
Rate Limiter	Restrict bandwidth used by a given type of traffic (e.g., TCP SYN on Port 80).	Blind rate limiting can increase P_{DS} because of dropped legitimate packets. Legitimate packets are affected when attacker and clients share a path to the server that is rate limited.	Bandwidth available to the attacker is limited, allowing other traffic to continue.
Proof-of-Work	Require clients to pay for CPU resources before being given service.	P_{DS} is 0 unless attacker adapts to change in protocol. Once this occurs, the attacker can cause DoS if sufficient resources are applied.	Attacker must apply significant resources (e.g., many CPUs) to maintain attack. Protocol exchange limits attackers ability to spoof IP addresses.

Table B-1 Single mitigation technology performance

Scenario	Attack	Mitigation Technology Behavior		Outcome
Rate Limiter/ Active Monitor	1000-pps SYN flood	Rate Limiter	Blindly filters TCP SYN packets, causing P_{DS} to exceed the no mitigation technology value; bandwidth becomes relatively unconstrained.	Negative DI and larger P_{DS} are found for Rate Limiter/Active Monitor combination compared to Active Monitor alone. However, Active Monitor can reduce P_{DS} relative to Rate Limiter alone while still maintaining bandwidth limits.
		Active Monitor	Classifies and resets attack connections causing P_{DS} to drop below the Rate Limiter only case.	
Proof-of-Work/ Active Monitor	1000-pps distributed Proof-of-Work	Proof-of-Work	Requires attacker to distribute effort to meet CPU payment requirements. Puzzle exchanges requirements force attacker to abandon IP spoofing.	Positive DI and a reduced P_{DS} (0.81 to 0.06) result from the lack of spoofed packets.
		Active Monitor	Classifies nodes and resets connections as designed.	
	Short-duration SYN flood spoofing legitimate nodes	Proof-of-Work	Drops SYN packets because Proof-of-Work protocol has not been followed by attacker.	Lack of communication between the two mitigation technologies results in P_{DS} being approximately the same as no mitigation technology for substantially less attacker effort.
		Active Monitor	Classifies spoofed IP addresses as BAD and resets next connection attempt by legitimate client using that address.	

Table B-2 Mitigation combination performance

APPENDIX C - LIST OF ACRONYMS AND ABBREVIATIONS

ACK	Acknowledgement
Bps	Bytes per Second
CAR	Committed Access Rate
CPU	Central Processing Unit
DATA	Defense Attack Tradeoff Analysis
DDOS	Distributed Denial of Service
DI	Differential Impact
DoS	Denial of Service
FIN	Finish
ICMP	Internet Control Message Protocol
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
JHU/APL	The Johns Hopkins University Applied Physics Laboratory
Mbps	Megabits per Second
M&S	Modeling and Simulation
NAI	Network Associates, Inc.
$P_{CC\|CE}$	Probability of Connection Completion given Connection Establishment
P_{CE}	Probability of Connection Establishment
P_{DS}	Probability of Denied Service
PLR	Packet Loss Rate
pps	Packets per Second
QoS	Quality of Service
RTO	Retransmission Timeout
sec	Second
SYN	Synchronization
TCP	Transmission Control Protocol
TRA	Turing Response Accuracy
UDP	User Datagram Protocol
WWW	World Wide Web