Inferring Internet Denial-of-Service Activity

David Moore
CAIDA
San Diego Supercomputer Center
University of California, San Diego
dmoore@caida.org

Geoffrey M. Voelker and Stefan Savage
Department of Computer Science and Engineering
University of California, San Diego
{voelker,savage}@cs.ucsd.edu

Abstract

In this paper, we seek to answer a simple question: "How prevalent are denial-of-service attacks in the Internet today?". Our motivation is to understand quantitatively the nature of the current threat as well as to enable longer-term analyses of trends and recurring patterns of attacks. We present a new technique, called "backscatter analysis", that provides an estimate of *worldwide* denial-of-service activity. We use this approach on three week-long datasets to assess the number, duration and focus of attacks, and to characterize their behavior. During this period, we observe more than 12,000 attacks against more than 5,000 distinct targets, ranging from well known e-commerce companies such as Amazon and Hotmail to small foreign ISPs and dial-up connections. We believe that our work is the only publically available data quantifying denial-of-service activity in the Internet.

1 Introduction

In February of 2000, a series of massive denial-of-service (DoS) attacks incapacitated several high-visibility Internet e-commerce sites, including Yahoo, Ebay, and E*trade. Next, in January of 2001, Microsoft's name server infrastructure was disabled by a similar assault. Despite attacks on high-profile sites, the majority of attacks are not well publicized. Many other domestic and foreign sites have also been victims, ranging from smaller commercial sites, to educational institutions, public chat servers and government organizations.

While it is clear from these anecdotal reports that denial-of-service attacks continue to be a problem, there is currently not much quantitative data about the prevalence of these attacks nor any representative characterization of their behavior. Unfortunately, there are multiple obstacles hampering the collection of an authoritative denial-of-service traffic dataset. Service providers and content providers consider such data sensitive and private. Even if it were allowed, monitoring traffic at enough sites to obtain a representative measure of Internet-wide attacks presents a significant logistical challenge. Consequently, the only contemporary public data we are aware of is a CSI/FBI survey study [8][1].

We believe that a strong quantitative foundation is necessary both for understanding the nature of today's threat and as a baseline for longer-term comparison and analysis. Our paper seeks to answer the simple question: "How prevalent are denial-of-service attacks in the Internet today?". As a means to this end, we describe a traffic monitoring technique called "backscatter analysis" for estimating the *worldwide* prevalence of denial-of-service attacks. Using backscatter analysis, we observe 12,805 attacks on over 5,000 distinct Internet hosts belonging to more than 2,000 distinct organizations during a three-week period. We further are able to estimate a lower-bound on the intensity of such attacks – some of which are in excess of 600,000 packets-per-second (pps) – and characterize the nature of the sites victimized.

The remainder of this paper is organized as follows: Section 2 describes the underlying mechanisms of denial-of-service attacks, Section 3 describes the backscatter technique, and limitations arising from its assumptions, and Section 4 explains our techniques for classifying attacks from monitored backscatter traffic. In Section 5 we describe our experimental platform, and present our results in Section 6. Finally, in Sections 7 and 8 we cover related work and summarize our find-

[1] The primary result from this report is that 27 percent of security professionals surveyed detected denial-of-service attacks during the year 2000.

ings.

2 Background

Denial-of-service attacks consume the resources of a remote host or network that would otherwise be used for serving legitimate users. There are two principal classes of attacks: *logic* attacks and *flooding* attacks. Attacks in the first class, such as the "Ping-of-Death", exploit existing software flaws to cause remote servers to crash or substantially degrade in performance. Many of these attacks can be prevented by either upgrading faulty software or filtering particular packet sequences, but they remain a serious and ongoing threat. The second class, flooding attacks, overwhelm the victim's CPU, memory, or network resources by sending large numbers of spurious requests. Because there is typically no simple way to distinguish the "good" requests from the "bad", it can be extremely difficult to defend against flooding attacks. For the purposes of this study we will focus solely on flooding attacks.

2.1 Attack types

There are two related consequences to a flooding attack – the network load induced and the impact on the victim's CPU. To load the network, an attacker generally sends small packets as rapidly as possible since most network devices (both routers and NICs) are limited not by bandwidth but by packet processing rate. Therefore, packets-per-second are usually the best measure of network load during an attack.

An attacker often simultaneously attempts to load the victim's CPU by requiring additional processing above and beyond that required to receive a packet. For example, the best known denial-of-service attack is the "SYN flood" [6] which consists of a stream of TCP SYN packets directed to a listening TCP port at the victim. For each such SYN packet received, the host victim must search through existing connections and if no match is found, allocate a new data structure for the connection. Moreover, the number of these data structures may be limited by the victim's operating system. Consequently, without additional protection, even a small SYN flood can overwhelm a remote host. There are many similar attacks that exploit other code vulnerabilities including TCP ACK, NUL, RST and DATA floods, IP fragment floods, ICMP Echo Request floods, DNS Request floods, and so forth.

2.2 Distributed attacks

While a single host can cause significant damage by sending packets at its maximum rate, attackers can (and do) mount more powerful attacks by leveraging the resources of multiple hosts. Typically an attacker compromises a set of Internet hosts (using manual or semi-automated methods) and installs a small attack daemon on each, producing a group of "zombie" hosts. This daemon typically contains both the code for sourcing a variety of attacks and some basic communications infrastructure to allow for remote control. Using variants of this basic architecture an attacker can focus a coordinated attack from thousands of zombies onto a single site.

Packet sent	Response from victim
TCP SYN (to open port)	TCP SYN/ACK
TCP SYN (to closed port)	TCP RST (ACK)
TCP ACK	TCP RST (ACK)
TCP DATA	TCP RST (ACK)
TCP RST	no response
TCP NULL	TCP RST (ACK)
ICMP ECHO Request	ICMP Echo Reply
ICMP TS Request	ICMP TS Reply
UDP pkt (to open port)	protocol dependent
UDP pkt (to closed port)	ICMP Port Unreach
...	...

Table 1: A sample of victim responses to typical attacks.

2.3 IP spoofing

To conceal their location, thereby forestalling an effective response, attackers typically forge, or "spoof", the IP source address of each packet they send. Consequently, the packets appear to the victim to be arriving from one or more third parties. Spoofing can also be used to "reflect" an attack through an innocent third party. While we do not address "reflector attacks" in this paper, we describe them more fully in Section 3.3.

3 Basic methodology

As noted in the previous section, attackers commonly spoof the source IP address field to conceal the location of the attacking host. The key observation behind our technique is that for direct denial-of-service attacks, most programs select source addresses at random for each packet sent. These programs include all of the most popular distributed attacking tools: Shaft, TFN, TFN2k, trinoo, all variants of Stacheldraht, mstream and Trinity). When a spoofed packet arrives at the victim, the victim usually sends what it believes to be an appropriate response to the faked IP address (such as shown in Table 1). Occasionally, an intermediate network device (such as a router, load balancer, or firewall) may issue its own reply to the attack via an ICMP message [21].

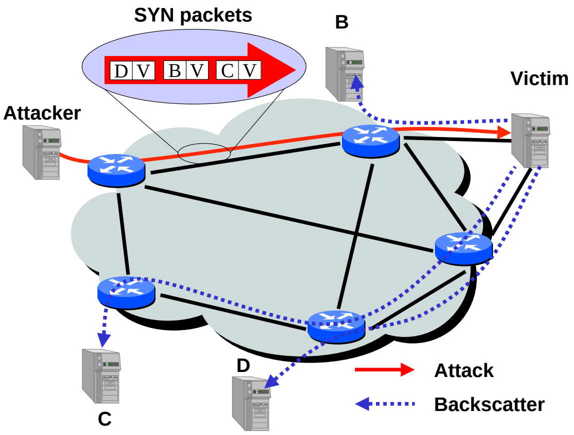

Figure 1: An illustration of backscatter in action. Here the attacker sends a series of SYN packets towards the victim V, using a series of random spoofed source addresses: named C, B, and D. Upon receiving these packets the victim responds by sending SYN/ACKs to each of spoofed hosts.

Again, these ICMP messages are sent to the randomly spoofed source address.

Because the attacker's source address is selected at random, the victim's responses are equi-probably distributed across the entire Internet address space, an inadvertent effect we call "backscatter"[2]. This behavior is illustrated in Figure 1.

3.1 Backscatter analysis

Assuming per-packet random source addresses, reliable delivery and one response generated for every packet in an attack, the probability of a given host on the Internet receiving at least one unsolicited response from the victim is $\frac{m}{2^{32}}$ during an attack of m packets. Similarly, if one monitors n distinct IP addresses, then the expectation of observing an attack is:

$$E(X) = \frac{nm}{2^{32}}$$

By observing a large enough address range we can effectively "sample" all such denial-of-service activity on the Internet. Contained in these samples are the identity of the victim, information about the kind of attack, and a timestamp from which we can estimate attack duration. Moreover, given these assumptions, we can also use the average arrival rate of unsolicited responses directed at the monitored address range to estimate the actual rate of the attack being directed at the victim, as follows:

$$R \geq R' \frac{2^{32}}{n}$$

where R' is the measured average inter-arrival rate of backscatter from the victim and R is the extrapolated attack rate in packets-per-second.

3.2 Address uniformity

The estimation approach outlined above depends on the spoofed source addresses being uniformly distributed across the entire IP address space. To check whether a sample of observed addresses are uniform in our monitored address range, we compute the Anderson-Darling (A2) test statistic [9] to determine if the observations are consistent with a uniform distribution. In particular, we use the implementation of the A2 test as specified in RFC2330 [19] at a 0.05 significance level.

3.3 Analysis limitations

There are three assumptions that underly our analysis:

- *Address uniformity*: attackers spoof source addresses at random.
- *Reliable delivery*: attack traffic is delivered reliably to the victim and backscatter is delivered reliably to the monitor.
- *Backscatter hypothesis*: unsolicited packets observed by the monitor represent backscatter.

We discuss potential biases that arise from these assumptions below.

Key among our assumptions is the random selection of source address. There are three reasons why this assumption may not be valid. First, some ISPs employ *ingress filtering* [12, 5] on their routers to drop packets with source IP addresses outside the range of a customer's network. Thus, an attacker's source address range may not include any of our monitored addresses and we will underestimate the total number of attacks.

"Reflector attacks" pose a second problem for source address uniformity. In this situation, an attacker "launders" the attack by sending a packet spoofed with the victim's source address to a third party. The third party responds by sending a response back towards the victim. If the packets to the third partie are addressed using a broadcast address (as with the popular smurf or fraggle attacks) then third parties may further amplify the attack. The key issue with reflector attacks is that the source address is specifically selected. Unless an IP address in the range we monitor is used as a reflector, we will be unable

[2] We did not originate this term. It is borrowed from Vern Paxson who independently discovered the same backscatter effect when an attack accidentally disrupted multicast connectivity by selecting global multicast addresses as source addresses [20].

to observe the attack. We have detected no instances of a monitored host involved in this sort of attack. Our inability to detect, "reflector attacks" cause us to underestimate the total number of denial-of-service attacks.

Finally, if the distribution of source addresses is not random, then any attempt to extrapolate the attack rate via the arrival rate of responses will produce an arbitrarily biased result. This particular problem can be mitigated by verifying that the distribution of observed source addresses is indeed uniform within the set of n addresses we observe.

Another limitation arises from our assumption that packets are delivered reliably and that every packet generates a response. During a large attack it is likely that packets from the attacker may be queued and dropped. Those packets that *do* arrive may be filtered or rate-limited by firewall or intrusion detection software [4] and moreover some forms of attack traffic (e.g., TCP RST messages) do not typically elicit a response. Finally, the responses themselves may be queued and dropped along the path back to our monitored address range. In particular, our estimate of the attack rate is necessarily limited to the capacity of smallest bottleneck link between the victim and our monitor. As with our random distribution assumption, these limitations will cause us to *underestimate* the number of attacks and the attack rate. However, they may also bias our characterization of victims (e.g., if large e-commerce sites are more likely to have rate-limiting software than educational sites, then we may disproportionately underestimate the size of attacks on this class of victim).

The final limitation of our technique is that we assume unsolicited responses represent backscatter from an attack. Any server on the Internet is free to send unsolicited packets to our monitored addresses, and these packets may be misinterpreted as backscatter from an attack. It is possible to eliminate accidental errors by choosing a quiescent address range for monitoring, filtering those packet flows consistently destined to a single host in the range and by high-pass filtering to only record sufficiently long and voluminous packet flows. However, a concerted effort by a third-party to bias our results would be difficult to detect and correct automatically. The most likely source of such bias arises from misinterpretation of random port scans as backscatter. While it is impossible to eliminate this possibility in general, we will show that it is extremely unlikely to be a factor in the vast majority of attacks we observe.

In spite of its limitations, we believe our overall approach is sound and provides at worst a conservative estimate of current denial-of-service activity.

4 Attack Classification

After collecting a large trace of backscatter packets, the first task is post-processing the trace. For this we group collections of related packets into clusters representing attacks. The choice of a specific aggregation methodology presents significant challenges. For example, it is often unclear whether contemporaneous backscatter indicating both TCP and ICMP-based attacks should be classified as a single attack or multiple attacks. More difficult still is the problem of determining the start and end times of an attack. In the presence of significant variability, too lenient a threshold can bias the analysis towards fewer attacks of longer duration and low average packet rates, while too strict an interpretation suggests a large number of short attacks with highly variable rates.

Without knowledge of the intent of the attacker or direct observation of the attack as it orchestrated by the attacker, it is impossible to create a synthetic classification system that will group all types of attacks appropriately for all metrics. Instead, we have chosen to employ two distinct classification methods: a flow-based analysis for classifying individual attacks – how many, how long and what kind – and an event-based method for analyzing the severity of attacks on short time scales.

4.1 Flow-based classification

For the purpose of this study, we define a flow as a series of consecutive packets sharing the same target IP address and IP protocol. We explored several approaches for defining flow lifetimes and settled on a fixed timeout approach: the first packet seen for a target creates a new flow and any additional packets from that target are counted as belonging to that flow if the packets are received within five minutes of the most recent packet in this flow. The choice of parameters here can influence the final results, since a more conservative timeout will tend to suggest fewer, longer attacks, while a shorter timeout will suggest a large number of short attacks. We chose five minutes as a human-sensible balance that is not unduly affected by punctuated attacks or temporary outages.

To reduce noise and traffic generated due to random Internet misconfiguration (for instance, one NetBIOS implementation/configuration sends small numbers unsolicited packets to our monitored address range) we discard all flows that do not have at least 100 packets and a flow duration of at least 60 seconds. These parameters are also somewhat arbitrary, but we believe they represent a reasonable baseline – below such thresholds it seems unlikely that an attack would cause significant damage. Finally, flows must contain packets sent to more than one of our monitored addresses.

We examine each individual flow and extract the following information:

- *TCP flag settings*: whether the flow consists of SYN/ACKs, RSTs, etc.
- *ICMP payload*: for ICMP packets that contain copies of the original packet (e.g. TTL expired) we break out the enclosed addresses, protocols, ports, etc.
- *Address uniformity*: whether the distribution of source addresses within our monitored range passes the Anderson-Darling (A2) test for uniformity to the 0.05 significance level.
- *Port settings*: for source and destination ports (for both UDP and TCP) we record whether the port range is fixed, is uniform under the A2 test, or is non-fixed and non-uniform.
- *DNS information*: the full DNS address of the source address – the victim.
- *Routing information*: the prefix, mask and origin AS as registered in our local BGP table on the morning of February 7th.

We generate a database in which each record characterizes the properties of a single attack.

4.2 Event-based classification

Because the choice of flow parameters can impact the estimated duration of an attack, the flow-based method may obscure interesting time-domain characteristics. In particular, attacks can be highly variable – with periodic bursts of activity – causing the flow-based method to vastly underestimate the short-term impact of an attack and overestimate the long-term impact.

We use an event-based classification method keyed entirely on the victim's IP address over fixed time-windows for examining time-domain qualities, such as the number of simultaneous attacks or the distribution of attack rates, For these analyses we divide our trace into one minute periods and record each *attack event* during this period. An attack event is defined by a victim emitting at least ten backscatter packets during a one minute period. We do not further classify attacks according to protocol type, port, etc, as the goal is to estimate the instantaneous impact on a particular victim. The result of this classification is a database in which each record characterizes the number of victims and the intensity of the attacks in each one minute period.

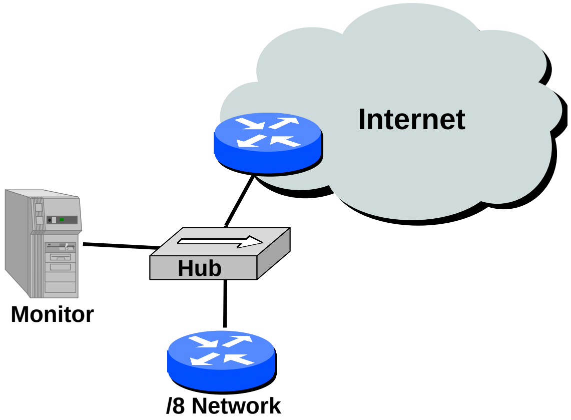

Figure 2: Our experimental backscatter collection platform. We monitor all traffic to our /8 network by passively monitoring data as it is forwarded through a shared hub. This monitoring point represents the only ingress into the network.

5 Experimental platform

For our experiments monitored the sole ingress link into a lightly utilized /8 network (comprising 2^{24} distinct IP addresses, or 1/256 of the total Internet address space). Our monitoring infrastructure, shown in Figure 2, consisted of a PC configured to capture all Ethernet traffic, attached to a shared hub at the router terminating this network. During this time, the upstream router did filter some traffic destined to the network (notably external SNMP queries) but we do not believe that this significantly impacted our results. We also have some evidence that small portions of our address prefix are occasionally "hijacked" by inadvertent route advertisements elsewhere in the Internet, but at worst this should cause us to slightly underestimate attack intensities. We collected three traces, each roughly spanning one week, starting on February 1st and extending to February 25th, and isolated the inbound portion of the network.

6 Results

Using the previously described flows-based approach (Section 4.1), we observed 12,805 attacks over the course of a week. Table 2 summarizes this data, showing more than 5,000 distinct victim IP addresses in more than 2,000 distinct DNS domains. Across the entire period we observed almost 200 million backscatter packets (again, representing less than $\frac{1}{256}$ of the actual attack traffic during this period).

In this section, we first show the overall frequency of attacks seen in our trace, and then characterize the attacks according to both the type of attack and the type of victim.

	Trace-1	Trace-2	Trace-3
Dates (2001)	Feb 01 – 08	Feb 11 – 18	Feb 18 – 25
Duration	7.5 days	6.2 days	7.1 days
Flow-based Attacks:			
Unique victim IPs	1,942	1,821	2,385
Unique victim DNS domains	750	693	876
Unique victim DNS TLDs	60	62	71
Unique victim network prefixes	1,132	1,085	1,281
Unique victim Autonomous Systems	585	575	677
Attacks	4,173	3,878	4,754
Total attack packets	50,827,217	78,234,768	62,233,762
Event-based Attacks:			
Unique victim IPs	3,147	3,034	3,849
Unique victim DNS domains	987	925	1,128
Unique victim DNS TLDs	73	71	81
Unique victim network prefixes	1,577	1,511	1,744
Unique victim Autonomous Systems	752	755	874
Attack Events	112,457	102,204	110,025
Total attack packets	51,119,549	78,655,631	62,394,290

Table 2: Summary of backscatter database.

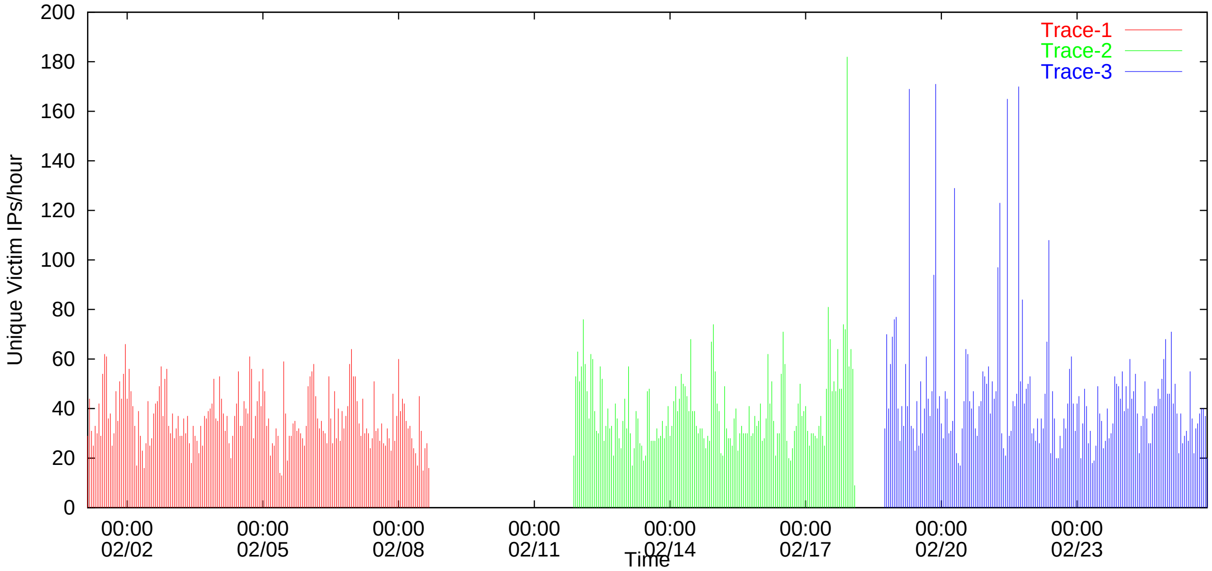

Figure 3: Estimated number of attacks per hour as a function of time (UTC).

Kind	Trace-1		Trace-2		Trace-3	
	Attacks	Packets (k)	Attacks	Packets (k)	Attacks	Packets (k)
TCP (RST ACK)	2,027 (49)	12,656 (25)	1,837 (47)	15,265 (20)	2,118 (45)	11,244 (18)
ICMP (Host Unreachable)	699 (17)	2,892 (5.7)	560 (14)	27,776 (36)	776 (16)	19,719 (32)
ICMP (TTL Exceeded)	453 (11)	31,468 (62)	495 (13)	32,001 (41)	626 (13)	22,150 (36)
ICMP (Other)	486 (12)	580 (1.1)	441 (11)	640 (0.82)	520 (11)	472 (0.76)
TCP (SYN ACK)	378 (9.1)	919 (1.8)	276 (7.1)	1,580 (2.0)	346 (7.3)	937 (1.5)
TCP (RST)	128 (3.1)	2,309 (4.5)	269 (6.9)	974 (1.2)	367 (7.7)	7,712 (12)
TCP (Other)	2 (0.05)	3 (0.01)	0 (0.00)	0 (0.00)	1 (0.02)	0 (0.00)

Table 3: Breakdown of response protocols.

6.1 Time series

Figure 3 shows a time series graph of the estimated number of actively attacked victims throughout the three traces, as sampled in one hour periods. There are two gaps in this graph corresponding to the gaps between traces. In contrast to other workloads, such as HTTP, the number of active attacks does not appear to follow any diurnal pattern (at least as observed from a single location). The outliers on the week of February 20th, with more than 150 victim IP addresses per hour, represent broad attacks against many machines in a common network. While most of the backscatter data averages one victim IP address per network prefix per hour, the ratio climbs to above five for many outliers.

6.2 Attack classification

In this section we characterize attacks according to the protocols used in response packets sent by victims, the protocols used in the original attack packets, and the rate and durations of attacks.

6.2.1 Response protocols

In Table 3 we decompose our backscatter data according to the protocols of responses returned by the victim or an intermediate host. For each trace we list both the number of attacks and the number backscatter packets for the given protocol. The numbers in parentheses show the relative percentage represented by each count. For example, 1,837 attacks in Trace 2 (47% of the total), were derived from TCP backscatter with the RST and ACK flags set.

We observe that over 50% of the attacks and 20% of the backscatter packets are TCP packets with the RST flag set. Referring back to Table 1 we see that RST is sent in response to either a SYN flood directed against a closed port or some other unexpected TCP packet. The next largest protocol category is ICMP host unreachable, comprising roughly 15% of the attacks. Almost all of these ICMP messages contain the TCP header from a packet directed at the victim, suggesting a TCP flood of some sort. Unfortunately, the TCP flags field cannot be recovered, because the ICMP response only includes the first 28 bytes of the original IP packet. ICMP host unreachable is generally returned by a router when a packet cannot be forwarded to its destination. Probing some of these victims we confirmed that a number of them could not be reached, but most were accessible, suggesting intermittent connectivity. This discontinuous reachability is probably caused by explicit “black holing’ on the part of an ISP.

We also see a number of SYN/ACK backscatter packets (likely sent directly in response to a SYN flood on an open port) and an equivalent number of assorted ICMP messages, including ICMP echo reply (resulting from ICMP echo request floods), ICMP protocol unreachable (sent in response to attacks using illegal combinations of TCP flags), ICMP fragmentation needed (caused by attacks with the “Dont Fragment” bit set) and ICMP administratively filtered (likely the result of some attack countermeasure). However, a more surprising finding is the large number of ICMP TTL exceeded messages – comprising between 36% and 62% of all backscatter packets observed, yet less than 15% of the total attacks. In fact, the vast majority of these packets occur in just a few attacks, including three attacks on @Home customers, two on China Telecom (one with almost 9 million backscatter packets), and others directed at Romania, Belgium, Switzerland and New Zealand. The attack on the latter was at an extremely high rate, suggesting an attack of more than 150,000 packets per second. We are unable to completely explain the mechanism for the generation of these time-exceeded messages. Upon examination of the encapsulated header that is returned, we note that several of them share identical “signatures” (ICMP Echo with identical sequence number, identification fields, and checksum) suggesting that a single attack tool was in use.

6.2.2 Attack protocols

We refine this data in Table 4 to show the distribution of *attack protocols*. That is, the protocol which must

Kind	Trace-1		Trace-2		Trace-3	
	Attacks	Packets (k)	Attacks	Packets (k)	Attacks	Packets (k)
TCP	3,902 (94)	28,705 (56)	3,472 (90)	53,999 (69)	4,378 (92)	43,555 (70)
UDP	99 (2.4)	66 (0.13)	194 (5.0)	316 (0.40)	131 (2.8)	91 (0.15)
ICMP	88 (2.1)	22,020 (43)	102 (2.6)	23,875 (31)	107 (2.3)	18,487 (30)
Proto 0	65 (1.6)	25 (0.05)	108 (2.8)	43 (0.06)	104 (2.2)	49 (0.08)
Other	19 (0.46)	12 (0.02)	2 (0.05)	1 (0.00)	34 (0.72)	52 (0.08)

Table 4: Breakdown of protocols used in attacks.

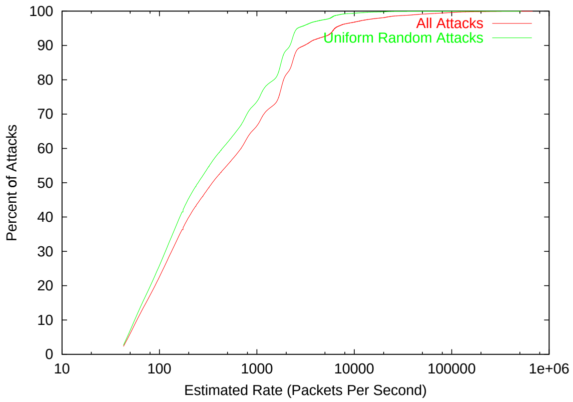

Figure 4: Cumulative distributions of estimated attack rates in packets per second.

have been used by the attacker to produce the backscatter monitored at our network. We see that more than 90% of the attacks use TCP as their protocol of choice, but a smaller number of ICMP-based attacks produce a disproportionate number of the backscatter packets seen. Other protocols represent a minor number of both attacks and backscatter packets. This pattern is consistent across all three traces.

In Table 5 we further break down our dataset based on the service (as revealed in the victim's port number) being attacked. Most of the attacks focus on multiple ports, rather than a single one and most of these are well spread throughout the address range. Many attack programs select random ports above 1024; this may explain why less than 25% of attacks show a completely uniform random port distribution according to the A2 test. Of the remaining attacks, the most popular static categories are port 6667 (IRC), port 80 (HTTP), port 23 (Telnet), port 113 (Authd). The large number of packets directed at port 0 is an artifact of our ICMP categorization – there are fewer than ten TCP attacks directed at port 0, comprising a total of less than 9,000 packets.

6.2.3 Attack rate

Figure 4 shows two cumulative distributions of attack event rates in packets per second. The lower curve shows the cumulative distribution of event rates for all attacks, and the upper curve shows the cumulative distribution of event rates for uniform random attacks, i.e., those attacks whose source IP addresses satisfied the A2 uniform distribution test described in Section 3.2. As described earlier, we calculate the attack event rate by multiplying the average arrival rate of backscatter packets by 256 (assuming that an attack represents a random sampling across the entire address space, of which we monitor $\frac{1}{256}$). Almost all attacks have no dominant mode in the address distribution, but sometimes small deviations from uniformity prevent the A2 test from being satisfied. For this reason we believe that there is likely some validity in the extrapolation applied to the complete attack dataset. Note that the attack rate (x-axis) is shown using a logarithmic scale.

Comparing the distributions, we see that the uniform random attacks have a lower rate than the distribution of all attacks, but track closely. Half of the uniform random attack events have a packet rate greater than 250, whereas half of all attack events have a packet rate greater than 350. The fastest uniform random event is over 517,000 packets per second, whereas the fastest overall event is over 679,000 packets per second.

How threatening are the attacks that we see? Recent experiments with SYN attacks on commercial platforms show that an attack rate of only 500 SYN packets per second is enough to overwhelm a server [10]. In our trace, 38% of uniform random attack events and 46% of all attack events had an estimated rate of 500 packets per second or higher. The same experiments show that even with a specialized firewall designed to resist SYN floods, a server can be disabled by a flood of 14,000 packets per second. In our data, 0.3% of the uniform random attacks and 2.4% of all attack events would still compromise these attack-resistant firewalls. We conclude that the majority of the attacks that we have monitored are fast enough to overwhelm commodity solutions, and a small fraction are fast enough to overwhelm even optimized countermeasures.

Of course, one significant factor in the question of threat posed by an attack is the connectivity of the victim. An attack rate that overwhelms a cable modem victim may be trivial a well-connected major server installation. Victim connectivity is a difficult to ascertain with-

Kind	Trace-1		Trace-2		Trace-3	
	Attacks	Packets (k)	Attacks	Packets (k)	Attacks	Packets (k)
Multiple Ports	2,740 (66)	24,996 (49)	2,546 (66)	45,660 (58)	2,803 (59)	26,202 (42)
Uniformly Random	655 (16)	1,584 (3.1)	721 (19)	5,586 (7.1)	1,076 (23)	15,004 (24)
Other	267 (6.4)	994 (2.0)	204 (5.3)	1,080 (1.4)	266 (5.6)	410 (0.66)
Port Unknown	91 (2.2)	44 (0.09)	114 (2.9)	47 (0.06)	155 (3.3)	150 (0.24)
HTTP (80)	94 (2.3)	334 (0.66)	79 (2.0)	857 (1.1)	175 (3.7)	478 (0.77)
0	78 (1.9)	22,007 (43)	90 (2.3)	23,765 (30)	99 (2.1)	18,227 (29)
IRC (6667)	114 (2.7)	526 (1.0)	39 (1.0)	211 (0.27)	57 (1.2)	1,016 (1.6)
Authd (113)	34 (0.81)	49 (0.10)	52 (1.3)	161 (0.21)	53 (1.1)	533 (0.86)
Telnet (23)	67 (1.6)	252 (0.50)	18 (0.46)	467 (0.60)	27 (0.57)	160 (0.26)
DNS (53)	30 (0.72)	39 (0.08)	3 (0.08)	3 (0.00)	25 (0.53)	38 (0.06)
SSH (22)	3 (0.07)	2 (0.00)	12 (0.31)	397 (0.51)	18 (0.38)	15 (0.02)

Table 5: Breakdown of attacks by victim port number.

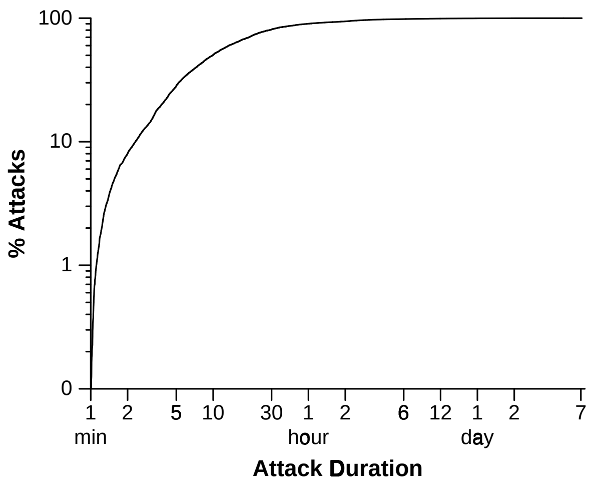

Figure 5: Cumulative distribution of attack durations.

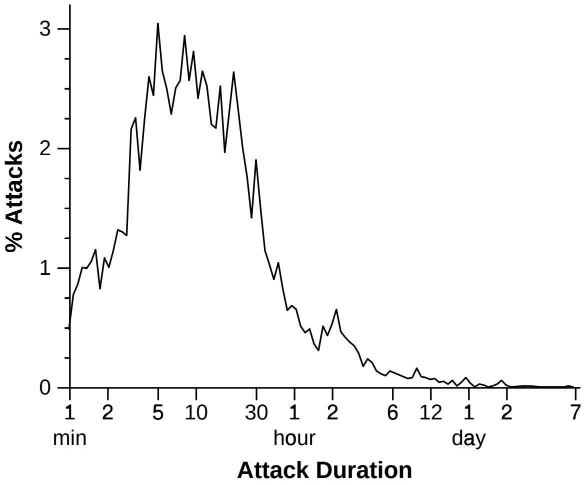

Figure 6: Probability density of attack durations.

out flooding the victim's link. Consequently, we leave correlation between attack rates and victim connectivity as an open problem.

6.2.4 Attack duration

While attack event rates characterize the intensity of attacks, they do not give insight on how long attacks are sustained. For this metric, we characterize the duration of attacks in Figures 5 and 6 across all three weeks of trace data. In these graphs, we use the flow-based classification described in Section 4 because flows better characterize attack durations while remaining insensitive to intensity. We also combine all three weeks of attacks for clarity; the distributions are nearly dentical for each week, and individual weekly curves overlap and obscure each other.

Figure 5 shows the cumulative distribution of attack durations in units of time; note that both the axes are logarithmic scale. In this graph we see that most attacks are relatively short: 50% of attacks are less than 10 minutes in duration, 80% are less than 30 minutes, and 90% last less than an hour. However, the tail of the distribution is long: 2% of attacks are greater than 5 hours, 1% are greater than 10 hours, and dozens spanned multiple days.

Figure 6 shows the probability density of attack durations as defined using a histogram of 150 buckets in the log time domain. The x-axis is in logarithmic units of time, and the y-axis is the percentage of attacks that lasted a given amount of time. For example, when the curve crosses the y-axis, it indicates that approximately 0.5% of attacks had a duration of 1 minute. As we saw in the CDF, the bulk of the attacks are relatively short, lasting from 3–20 minutes. From this graph, though, we see that there are peaks at rounded time durations in this interval at durations of 5, 10, and 20 minutes. Immediately before this interval there is a peak at 3 minutes, and immediately after a peak at 30 minutes. For attacks with longer durations, we see a local peak at 2 hours in the long tail.

6.3 Victim classification

In this section we characterize victims according to DNS name, top-level domain, Autonomous System, and degree of repeated attacks.

6.3.1 Victim Name

Table 6 shows the distribution of attacks according to the DNS name associated with the victim's IP address. We classify these using a hand-tuned set of regular expression matches (i.e. DNS names with "dialup" represent modems, "dsl" or "home.com" represent broadband, etc). The majority of attacks are not classified by this scheme, either because they are not matched by our criteria (shown by "other"), or more likely, because there was no valid reverse DNS mapping (shown by "In-Addr Arpa").

Of the remaining attacks there are several interesting observations. First, there is a significant fraction of attacks directed against home machines – either dialup or broadband. Some of these attacks, particularly those directed towards cable modem users, constitute relatively large, severe attacks with rates in the thousands of packets per second. This suggests that minor denial-of-service attacks are frequently being used to settle personal vendettas. In the same vein we anecdotally observe a significant number of attacks against victims running "Internet Relay Chat" (IRC), victims supporting multi-player game use (e.g. battle.net), and victims with DNS names that are sexually suggestive or incorporate themes of drug use. We further note that many reverse DNS mappings have been clearly been compromised by attackers (e.g., DNS translations such as "is.on.the.net.illegal.ly" and "the.feds.cant.secure.their.shellz.ca").

Second, there is a small but significant fraction of attacks directed against network infrastructure. Between 2–3% of attacks target name servers (e.g., ns4.reliablehosting.com), while 1–3% target routers (e.g., core2-corel-oc48.paol.above.net). Again, some of these attacks, particularly a few destined towards routers, are comprised of a disproportionately large number of packets. This point is particularly disturbing, since overwhelming a router could deny service to *all* end hosts that rely upon that router for connectivity.

Finally, we are surprised at the diversity of different commercial attack targets. While we certainly find attacks on bellwether Internet sites including aol.com, akamai.com, amazon.com and hotmail.com, we also see attacks against a large range of smaller and medium sized businesses.

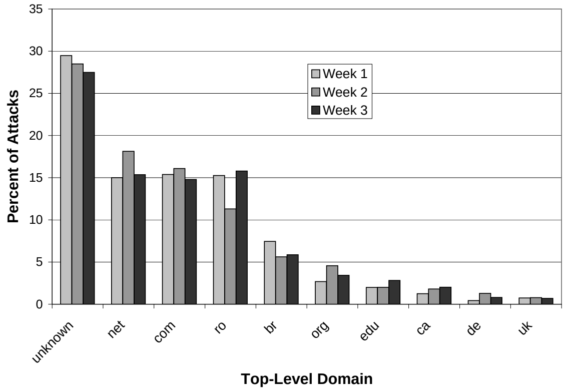

Figure 7: Distribution of attacks to the 10 top-level domains (TLDs) that received the most number of attacks.

6.3.2 Top-level domains

Figure 7 shows the distribution of attacks to the 10 most frequently targeted top-level domains (TLDs). For each TLD displayed on the x-axis, we show one value for each of the three weeks of our study in progressive shades of grey. Note that the TLDs are sorted by overall attacks across all three weeks.

Comparing the number of attacks to each TLD from week to week, we see that there is little variation. Each TLD is targeted by roughly the same percentage of attacks each week. The domain `unknown` represents those attacks in which a reverse DNS lookup failed on the victim IP address (just under 30% of all attacks). In terms of the "three-letter" domains, both `com` and `net` were each targeted by roughly 15% of the attacks, but `edu` and `org` were only targeted by 2–4% of the attacks. This is not surprising, as sites in the `com` and `net` present more attractive and newsworthy targets. Interestingly, although one might have expected attacks to sites in `mil`, `mil` did not show up in any of our reverse DNS lookups. We do not yet know what to conclude from this result; for example, it could be that `mil` targets fall into our `unknown` category.

In terms of the country-code TLDs, we see that there is a disproportionate concentration of attacks to a small group of countries. Surprisingly, Romania (`ro`), a country with a relatively poor networking infrastructure, was targeted nearly as frequently as `net` and `com`, and Brazil (`br`) was targeted almost more than `edu` and `org` combined. Canada, Germany, and the United Kingdom were all were targeted by 1–2% of attacks.

6.3.3 Autonomous Systems

As another aggregation of attack targets, we examined the distribution of attacks to Autonomous Systems (ASes). To determine the origin AS number associated

Kind	Trace-1		Trace-2		Trace-3	
	Attacks	Packets (k)	Attacks	Packets (k)	Attacks	Packets (k)
Other	1,917 (46)	19,118 (38)	1,985 (51)	25,305 (32)	2,308 (49)	17,192 (28)
In-Addr Arpa	1,230 (29)	16,716 (33)	1,105 (28)	24,645 (32)	1,307 (27)	26,880 (43)
Broadband	394 (9.4)	9,869 (19)	275 (7.1)	13,054 (17)	375 (7.9)	8,513 (14)
Dial-Up	239 (5.7)	956 (1.9)	163 (4.2)	343 (0.44)	276 (5.8)	1,018 (1.6)
IRC Server	110 (2.6)	461 (0.91)	88 (2.3)	2,289 (2.9)	111 (2.3)	6,476 (10)
Nameserver	124 (3.0)	453 (0.89)	84 (2.2)	2,796 (3.6)	90 (1.9)	451 (0.72)
Router	58 (1.4)	2,698 (5.3)	76 (2.0)	4,055 (5.2)	125 (2.6)	682 (1.1)
Web Server	54 (1.3)	393 (0.77)	64 (1.7)	5,674 (7.3)	134 (2.8)	730 (1.2)
Mail Server	38 (0.91)	156 (0.31)	35 (0.90)	71 (0.09)	26 (0.55)	292 (0.47)
Firewall	9 (0.22)	7 (0.01)	3 (0.08)	3 (0.00)	2 (0.04)	1 (0.00)

Table 6: Breakdown of victim hostnames.

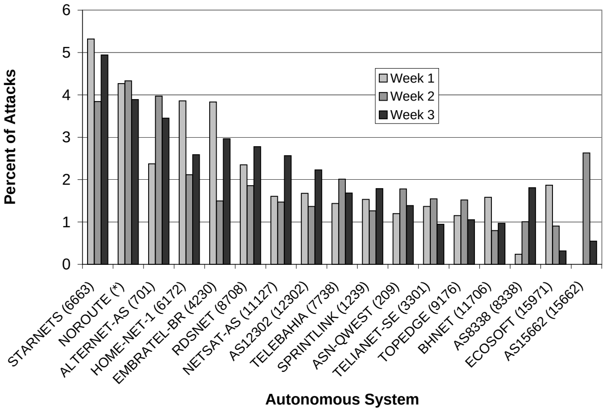

Figure 8: Distribution of attacks to Autonomous Systems (ASes) that were targeted by at least 1% of all attacks.

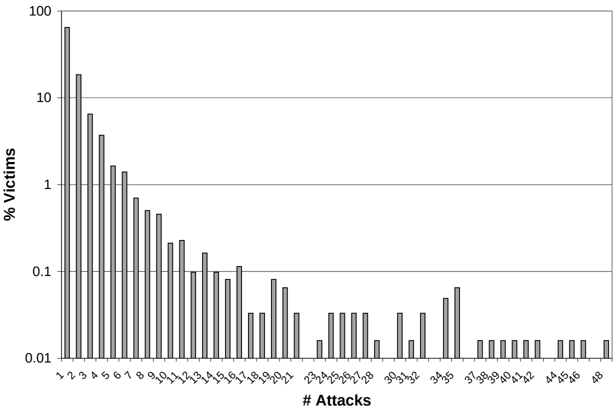

Figure 9: Histogram counting the number of victims of repeated attacks across all traces.

with the victim of an attack, we performed longest prefix matching against a BGP routing table using the victim's IP address. To construct this table, we took a snapshot from a border router with global routes on February 7, 2001. We then mapped AS numbers to identifying names using the NetGeo [17] service to do lookups in registry whois servers. We labeled addresses which had no matching prefix as ”NOROUTE”.

Figure 8 shows the distribution of attacks to the 17 ASes that were targeted by at least 1% of all attacks. As with top-level domains, each AS named on the x-axis is associated with three values, one for each of the three weeks of our study in progressive shades of grey. Note that the ASes are sorted by overall attacks across all three weeks.

From Figure 8, we see that no single AS or small set of ASes is the target of an overwhelming fraction of attacks: STARNETS was attacked the most, but only received 4-5% of attacks. However, the distribution of ASes attacked does have a long tail. The ASes shown in Figure 8 accounted for 35% of all attacks, yet these ASes correspond to only 3% of all ASes attacked. About 4% of attacks each week had no route according to our offline snapshot of global routes.

Compared with TLDs, ASes experienced more variation in the number of attacks targeted at them for each week. In other words, there is more stability in the type or country of victims than the ASes in which they reside. For example, EMBRATEL's percentage of attacks varies by more than a factor of 2, and AS 15662, an unnamed AS in Yugoslavia, did not show up in week 1 of the traces.

6.3.4 Victims of repeated attacks

Figure 9 shows a histogram of victims of repeated attacks for all traces combined. The values on the x-axis correspond to the number of attacks to the same victim in the trace period, and the values on the y-axis show what percentage of victims were attacked a given number of times in logarithmic scale. For example, the majority of victims (65%) were attacked only once, and many of the remaining victims (18%) were attacked twice. Overall, most victims (95%) were attacked five or fewer times. For the remaining victims, most were attacked less than a dozen times, although a handful of hosts were attacked quite often. In the trace period, one host was attacked 48 times for durations between 72 seconds and 5 hours (at times simultaneously). The graph is also truncated: there are 5 outlier victims attacked 60–70 times, and one unfortunate victim attacked 102 times in a one week span.

6.4 Validation

The backscatter hypothesis states that unsolicited packets represent responses to spoofed attack traffic. This theory, which is at the core of our approach, is difficult to validate beyond all doubt. However, we can increase our confidence significantly through careful examination of the data and via related experiments.

First, an important observation from Table 3 is that roughly 80% of attacks and 98% of packets are attributed to backscatter that does not itself provoke a response (e.g. TCP RST, ICMP Host Unreachable). Consequently, these packets could not have been used for probing our monitored network; therefore network probing is not a good alternative explanation for this traffic.

Next, we were able to duplicate a portion of our analysis using data provided by Vern Paxson taken from several University-related networks in Northern California. This new dataset covers the same period, but only detects TCP backscatter with the SYN and ACK flags set. The address space monitored was also much smaller, consisting of three /16 networks ($\frac{3}{65536}$'s of the total IP address space). For 98% of the victim IP addresses recorded in this smaller dataset, we find a corresponding record at the same time in our larger dataset. We can think of no other mechanism other than backscatter that can explain such a close level of correspondence.

Finally, Asta Networks provided us with data describing denial-of-service attacks directly detected at monitors covering a large backbone network. While their approach and ours capture different sets of attacks (in part due ingress filtering as discussed in Section 3 and in part due to limited peering in the monitored network), their data qualitatively confirms our own; in particular we were able to match several attacks they directly observed with contemporaneous records in our backscatter database.

7 Related work

While denial-of-service has long been recognized as a problem [14, 18], there has been limited research on the topic. Most of the existing work can be roughly categorized as being focused on tolerance, diagnosis and localization. The first category is composed of both approaches for mitigating the impact of specific attacks [4, 16] and general system mechanisms [25, 1] for controlling resource usage on the victim machine. Usually such solutions involve a quick triage on data packets so minimal work is spent on the attacker's requests and the victim can tolerate more potent attacks before failing. These solutions, as embodied in operating systems, firewalls, switches and routers, represent the dominant current industrial solution for addressing denial-of-service attacks.

The second area of research, akin to traditional intrusion detection, is about techniques and algorithms for automatically detecting attacks as they occur [22, 13]. These techniques generally involve monitoring links incident to the victim and analyzing patterns in the arriving and departing traffic to determine if an attack has occurred.

The final category of work, focuses on identifying the source(s) of DoS attacks in the presence of IP spoofing. The best known and most widely deployed of these proposals is so-called *ingress* and *egress* filtering [12, 5]. These techniques, which differ mainly in whether they are manually or automatically configured, cause routers to drop packets with source addresses that are not used by the customer connected to the receiving interface. Given the practical difficulty of ensuring that all networks are filtered, other work has focused on developing tools and mechanisms for tracing flows of packets through the network independent of their ostensibly claimed source address [3, 26, 23, 2, 24, 11].

There is a dearth of research concerned with quantifying attacks within the Internet – denial-of-service or otherwise. Probably the best known prior work is Howard's PhD thesis – a longitudinal study of incident reports received by the Computer Emergency Response Team (CERT) from 1989 to 1995 [15]. Since then, CERT has started a new project, called AIR-CERT, to automate the collection of intrusion detection data from a number of different organizations, but unfortunately their results are not yet available [7]. To our knowledge ours is the only quantitative and empirical study of wide-area denial-of-service attacks to date.

8 Conclusions

In this paper we have presented a new technique, "backscatter analysis," for estimating denial-of-service attack activity in the Internet. Using this technique, we have observed widespread DoS attacks in the Internet, distributed among many different domains and ISPs. The size and length of the attacks we observe are heavy-tailed, with a small number of long attacks constituting a significant fraction of the overall attack volume. Moreover, we see a surprising number of attacks directed at a few foreign countries, at home machines, and towards particular Internet services.

Acknowledgments

We would like to thank a number of people for their contributions to this project. We are particularly grateful to Brian Kantor and Jim Madden of UCSD who provided access to key network resources and helped us understand the local network topology. kc claffy and Colleen Shannon at CAIDA provided support and valuable feedback throughout the project. David Wetherall and Gretta Bartels at Asta Networks donated their time, data and insight. Vern Paxson of ACIRI also provided valuable data and feedback at several stages of our thinking. Finally, we thank the anonymous reviewers for their comments and suggestions. Support for this work was provided by DARPA NGI Contract N66001-98-2-8922, NSF grant NCR-9711092, and Asta Networks.

References

[1] Gaurav Banga, Peter Druschel, and Jeffrey Mogul. Resource Containers: A New Facility for Resource Management in Server Systems. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 45–58, February 1999.

[2] Steven M. Bellovin. ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, March 2000.

[3] Hal Burch and Bill Cheswick. Tracing Anonymous Packets to Their Approximate Source. In *Proceedings of the 2000 USENIX LISA Conference*, pages 319–327, New Orleans, LA, December 2000.

[4] Cisco Systems. Configuring TCP Intercept (Prevent Denial-of-Service Attacks). Cisco IOS Documentation, December 1997.

[5] Cisco Systems. Unicast Reverse Path Forwarding. Cisco IOS Documentation, May 1999.

[6] Computer Emergency Response Team. CERT Advisory CA-1996-21 TCP SYN Flooding Attacks. `http://www.cert.org/advisories/CA-1996-21.html`, September 1996.

[7] Computer Emergency Response Team. AirCERT. `http://www.cert.org/kb/aircert/`, 2000.

[8] Computer Security Institute and Federal Bureau of Investigation. 2000 CSI/FBI Computer Crime and Security Survey. Computer Security Institute publication, March 2000.

[9] R. D'Agostino and M. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, Inc., 1986.

[10] Tina Darmohray and Ross Oliver. Hot Spares For DoS Attacks. *;login:*, 25(7), July 2000.

[11] Drew Dean, Matt Franklin, and Adam Stubblefield. An Algebraic Approach to IP Traceback. In *Proceedings of the 2001 Network and Distributed System Security Symposium*, San Diego, CA, February 2001.

[12] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. RFC 2827, May 2000.

[13] Mark Fullmer and Steve Romig. The OSU Flow-tools Package and Cisco Netflow logs. In *Proceedings of the 2000 USENIX LISA Conference*, New Orleans, LA, December 2000.

[14] Virgil Gilgor. A Note on the Denial-of-Service Problem. In *Proceedings of the 1983 IEEE Symposium on Security and Privacy*, Oakland, CA, 1983.

[15] John D. Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, August 1998.

[16] Phil Karn and William Simpson. Photuris: Session-Key Management Protocol. RFC 2522, March 1999.

[17] David Moore, Ram Periakaruppan, Jim Donohoe, and kc claffy. Where in the World is netgeo.caida.org? In *INET 2000 Proceedings*, June 2000.

[18] Roger Needham. Denial of Service: An Example. *Communications of the ACM*, 37(11):42–47, November 1994.

[19] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. RFC 2330: Framework for IP performance metrics, May 1998.

[20] Vern Paxson. Personal Communication, January 2001.

[21] Jon Postel, Editor. Internet Control Message Protocol. RFC 792, September 1981.

[22] Steve Romig and Suresh Ramachandran. Cisco Flow Logs and Intrusion Detection at the Ohio State university. *login; magazine*, pages 23–26, September 1999.

[23] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical Network Support for IP Traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, pages 295–306, Stockholm, Sweden, August 2000.

[24] Dawn Song and Adrian Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. In *Proceedings of the 2001 IEEE INFOCOM Conference*, Anchorage, AK, April 2001.

[25] Oliver Spatscheck and Larry Peterson. Defending Against Denial of Service Attacks in Scout. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 59–72, February 1999.

[26] Robert Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In *Proceedings of the 2000 USENIX Security Symposium*, pages 199–212, Denver, CO, July 2000.