A study of denial of service attacks on the Internet

David J. Marchette

Abstract

The public Internet is a critical component of the information infrastructure supporting finance, commerce, and civil and national defense. Denial of service attacks on major Internet sites, both the direct effect on the attacked sites and the indirect collateral effects on the Internet as a whole do considerable financial damage on a regular basis. Denial of service attacks could be a part of a concerted attack on the flow of information. This coupled with a physical attack of some kind poses a substantial threat.

Monitoring these attacks in a timely manner is problematic because the institutions under attack often have good financial and security incentives not to share that information, and getting the information in a timely (cyber-scale) manner is difficult without a (computerized) automatic notification process. Remote detection using backscatter allows the detection of attacks in a completely passive manner without any cooperation from the primary target computer(s). This paper discusses some of the mathematical and statistical aspects of backscatter analysis, and illustrates some interesting practical issues in the analysis.

1 Introduction

Suppose there is a coordinated denial of service attack (using one of a selection of freely available tools) on the public banking access sites of the 10 largest US banks (or the largest stock trading sites). Financial institutions are reluctant to share information, so it might take a while (hours or days) to sort out the size and the scope of the attack, or to even find out that the attack took place. A method of determining the scope of the attacks without relying on self-reporting is clearly needed.

The basic idea of most denial of service attacks is to flood a computer with bogus requests, or otherwise cause it to devote resources to the attack at the expense of the legitimate users of the system. A classic in this genre is the SYN flood. The attacker sends SYN packets requesting a connection, but never completes the handshake. One way to do this is to set the source IP address to a nonexistent address (this process of changing the source address is called "spoofing" the address). For each SYN packet, the victim computer allocates a session and waits a certain amount of time before "timing out" and releasing the session. If enough of these "bogus" SYN packets are sent, all the available sessions are devoted to processing the attack, and no legitimate users can connect to the machine.

A related attack is to send packets that are out of sequence, or errors, forcing the victim computer to spend time handling the errors. For example, if a SYN/ACK packet is sent without having received an initiating SYN packet, the destination computer generates and sends an RST (reset) packet. If the attacker can arrange to have millions of SYN/ACK packets sent, the victim computer will spend all its resources handling these errors, thus denying service to legitimate users. One way to arrange this, is through a distributed denial of service tool, such as trinoo or TFN2k. These tools compromise a set of computers, dispersed across the IP address space, then use these "intermediate victims" to send thousands of packets to the intended victim. Each packet is crafted to have a random (spoofed) source IP address, so the attacking machines cannot be identified. See [Mar01], [Che01] and [NNM01] for descriptions of some distributed denial of service attacks.

The result of such an attack is a number of reset (or other) packets appearing at random sites around the Internet, with no obvious session or initiating packets to explain them. See Figure 1. This is used by [MVS01] to estimate the number of denial of service attacks during three one week periods, by counting how many unsolicited packets are seen addressed to one of the 2^{24} possible IP addresses they monitored.

2 Analysis

Following [MVS01], we can compute some of the probabilities of detection needed to analyze backscatter packets. Assume the spoofed IP addresses are generated randomly, uniformly on all 2^{32} addresses, and independently. Assume there are m packets sent in an attack on a given victim. If we monitor all packets to n IP addresses, then it is easy to see that the probability of detecting an attack is:

$$P[\text{detect attack}] = 1 - \left(1 - \frac{n}{2^{32}}\right)^m . \tag{1}$$

From this, one obtains the result that the expected number of backscatter packets we detect is

$$\frac{nm}{2^{32}}. \tag{2}$$

We would like to determine how many packets were originally sent. This will give an estimate for the severity of the attack, and might allow us to infer whether the attack was likely to have been mounted by multiple attackers, for example through a distributed denial of service tool. To do this, note that the probability of seeing exactly j packets, under our independence assumption, is

$$P[j \text{ packets}] = \binom{m}{j} \left(\frac{n}{2^{32}}\right)^j \left(1 - \frac{n}{2^{32}}\right)^{m-j} . \tag{3}$$

The maximum likelihood estimate for m, using Equation 3, is

$$\hat{m} = \left\lfloor \frac{j2^{32}}{n} \right\rfloor . \tag{4}$$

Thus, if we see j packets, we can use Equation 4 to estimate the size of the attack.

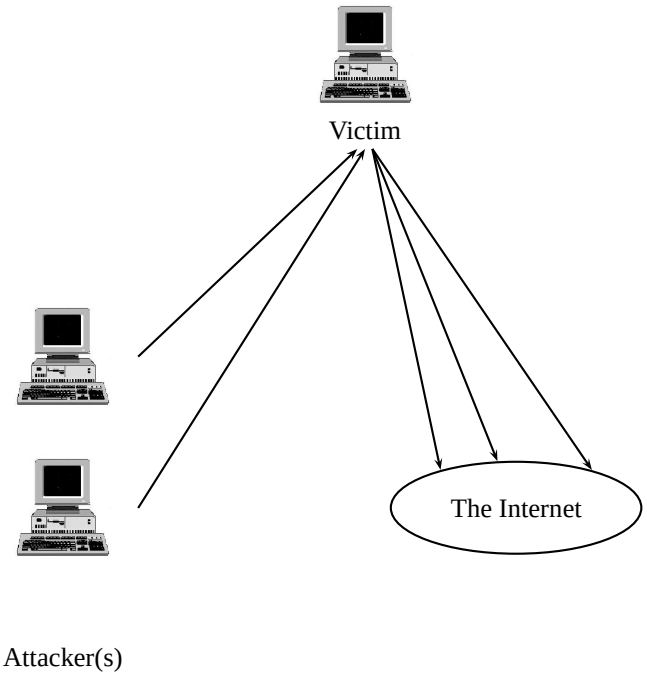

Figure 1: Backscatter from a denial of service attack. Packets are sent to the victim from one or more attackers. These packets have spoofed IP addresses, which cause the victim's response to be sent to random addresses within the Internet.

Note that if the attacker chooses from a subset of all possible IP addresses, say of size N, then we must replace 2^{32} in Equations 1–4 with N. These equations, then, under the assumptions of uniformity and independence, allow us to estimate the original size of the attack from the packets we see at our network, assuming we know the original size of the pool of IP addresses from which they are selected.

There is also the question of determining the number of attacks. [MVS01] do this by defining an attack as a series of packets with a maximum inter-packet gap less than a fixed value. The idea being that if there is a long enough gap between packets, then it is reasonable to assume that these correspond to different attacks. They then count the number of attacks they detect, and report over 12,000 attacks in the three week period they investigated.

This assumes that all attacks generate packets to the network monitored. We will assume that all attack packets generate backscatter (until the machine ceases to function), ignoring issues such as filtering firewalls or other kinds of mechanisms that may block either the attack or the backscatter packets. If our monitored network (n IP addresses) is sufficiently small, and a sufficiently small number of packets are sent in the

attack, there is a reasonable probability that we will receive no packets.

Several calculations are possible to determine whether the assumptions are valid. For example, [MVS01] suggest using the Anderson-Darling test of uniformity to test that the IP addresses are in fact uniformly distributed. We will discuss this further below. This of course assumes we know how many IP addresses are in the complete pool. A perusal of the attack code available on the Internet shows that the tools often allow the user to choose which octets of the IP address to randomize, thus reducing the pool. Assume the pool contains N addresses, and we monitor n addresses, and that the attack packets are sent every t seconds. Then if we knew how long a gap one should expect to see between detected packets, one could use this to estimate N, and thus be able to use the equations above for the estimate of the size of the attack. This would also be important for the determination of a definition of attack, as one would want the gap to be many standard deviations larger than this expected delay. The calculation is straightforward. The expected number of attack packets between two detected packets (assuming independence) is:

$$\begin{aligned}\sum_{s=1}^{N}\left(1-\frac{n}{N}\right)^{s-1}\frac{n}{N}s &= \frac{(1-(n+1)(1-\frac{n}{N})^{N})N}{n}\\ &\approx \frac{N(1-e^{-N})}{n}\\ &\approx \frac{N}{n}\end{aligned}$$

The variance of the number of packets between two detected packets is

$$\begin{aligned}&\sum_{s=1}^{N}\left(1-\frac{n}{N}\right)^{s-1}\frac{n}{N}s^2-\left(\sum_{s=1}^{N}\left(1-\frac{n}{N}\right)^{s-1}\frac{n}{N}s\right)^2\\ =\;&\frac{N(N-n-N(1+n)^2(1-\frac{n}{N})^{2N}-n(1-\frac{n}{N})^{N}(nN-1))}{n^2}\\ \approx\;&\frac{N(N-n)}{n^2}.\end{aligned}$$

So, from this we see that we expect a gap of around $\frac{tN}{n}$ seconds between packets from a given victim. For example, if an attacker sends 100 packets per second, and one monitors 2^{24} IP addresses, one expects to see a new packet about every 2.5 seconds, and a spread of three standard deviations gives a 10 second gap. This rate of attack is quite low ([MVS01] claim intensities of as large as 600,000 packets per second), but this is only for illustration's sake (even at this rate, a SYN flood can be quite effective). Similar calculations can easily be done for other values of n, N, and t.

All these calculations have been predicated on the attacker choosing randomly from 2^{32} possible IP addresses. Many attack tools choose from a subset of these, such as only selecting octets from the range 1–254. This can be easily incorporated in the above analysis, by replacing the 2^{32} by the appropriate number.

Table 1: Data sets used in the SYN/ACK study.

Data Set Name	Duration	# days	# packets
April	April 4 – April 17	14	10,449
May	May 9 – May 17	9	23,264
June	June 1 – June 15	15	27,845
July	July 1 – July 15	15	59,666
Sept	Sept 1 – Sept 17	17	210,774
Oct	Sept 19 – Oct 15	26	1,253,714
Dec	Oct 28 – Dec 12	66	5,421,893
Jan	Jan 1 – Jan 31	31	665,392
	Total	193	6,672,997

3 Experimental Results

To determine the extent that the assumptions of the theory are met, we consider a data set taken from a network of 2^{16} IP addresses. The data consists of unsolicited SYN/ACK packets received during two periods: April 4, 2001 – Jul 16, 2001 and September 1, 2001 – Jan 31, 2002. During these periods there were times when the sensor was down, for a total of 210 hours. The full data set consisted of 5,842 hours. We refer to the network on which the data was collected as the "protected network" throughout this discussion.

Missing data brings up one of the practical issues in a study of this kind. The protected network is a working network with a moderate load, and so there is the problem of determining which packets were solicited and which were not. This is exacerbated if there are packets that were not captured by the sensor, either because it was unable to handle the load or because the sensor was down. With SYN/ACK packets, we need to know if the SYN packet was sent. If it was, and the sensor failed to capture it, we will notice further packets (ACKs, PUSHs, etc), and can therefore determine that the SYN/ACK is a part of a legitimate session, and therefor not backscatter. For this reason, we focus on SYN/ACK packets in this section.

3.1 The Data

In order to avoid the gaps in our data collection, we broke the data into eight subsets, as depicted in Table 1. These are named according to the last month in which data was collected for that subset. As will be seen, this split was not perfect, as there were still a few gaps within the larger subsets.

We further restrict our investigation to web server (port 80) traffic. Thus we are considering only unsolicited SYN/ACK packets to our network from port 80. Figures 2 and 3 depict the data for the eight data sets. In these, the x-axis corresponds to time (in hours) from the start of the data set, and the y-axis corresponds to the victim (source) IP address. The IP address is always a 32-bit number with the highest octet in the highest bits. One dot is plotted for every packet (there is considerable overplotting in these pictures, but they serve to illustrate the data).

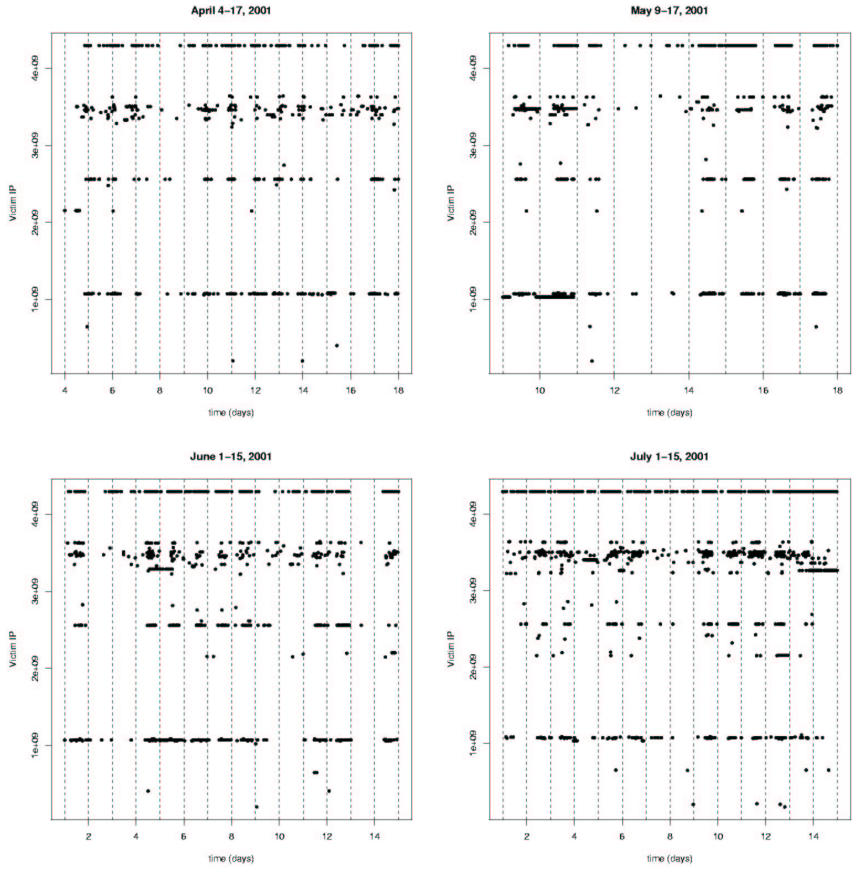

Figure 2: The attacks for the first four data sets. The x-axis is time, the y-axis is a 32-bit number corresponding to victim IP address. A dot is placed for each packet. Days are indicated by dotted lines.

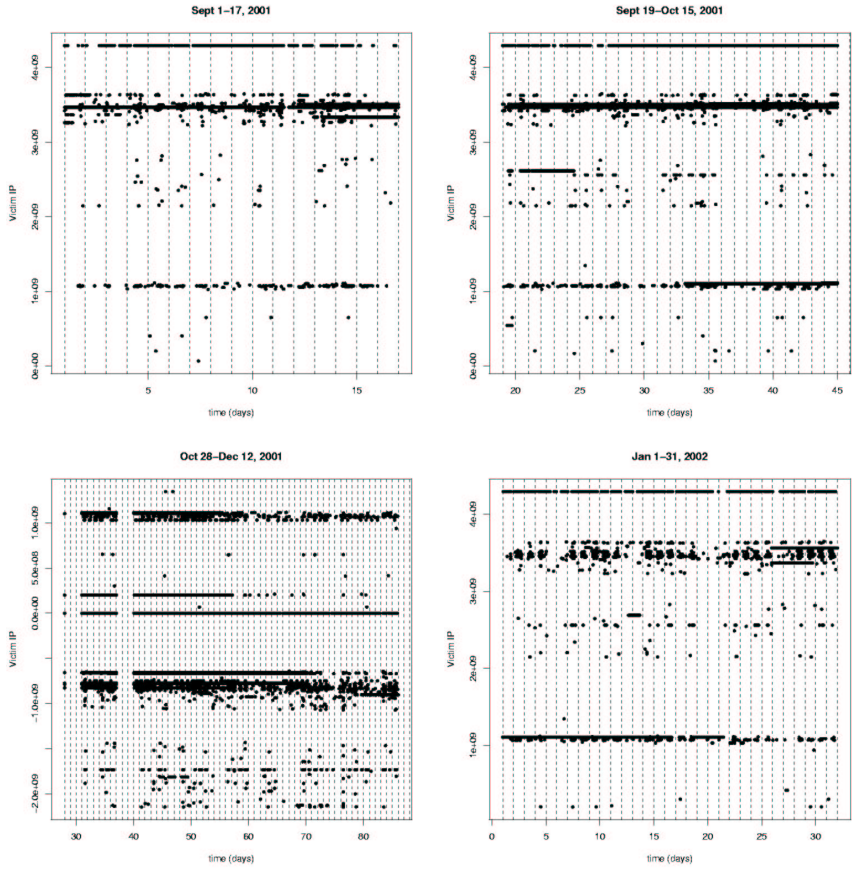

Figure 3: The attacks for the second four data sets. The x-axis is time, the y-axis is a 32-bit number corresponding to victim IP address. A dot is placed for each packet. Days are indicated by dotted lines.

Table 2: Number of attacks in each data set.

Data Set	T = 5 minutes	T = 1 hour
April	1,510	1,231
May	3,072	1,585
June	2,901	2,248
July	1,727	1,220
Sept	3,493	1,520
Sept/Oct	5,216	1,847
Oct/Dec	48,050	3,990
Jan	3,804	3,070

As can be seen in these Figures, there are a number of obvious attacks, as well as some very long-lived attacks. At this resolution it is impossible to count the attacks, and so we need to define exactly what we mean by an attack. For our purposes, we define an attack to be a sequence of packets from a single victim such that no gap between packets exceeds a fixed value (T). The results for two values for this threshold are presented in Table 2. If we restrict our definition to those attacks for which we received more than ten packets, we have the results reported in Table 3.

Table 3: Number of attacks in each data set for which there were more than 10 packets.

Data Set	T = 5 minutes	T = 1 hour
April	54	42
May	62	60
June	97	80
July	149	107
Sept	375	192
Sept/Oct	1,324	177
Oct/Dec	6,551	414
Jan	263	206

Some care is needed in counting the packets in an attack. Figure 4 depicts the packets from one victim. The destination (spoofed) IP addresses are on the y-axis, and time is on the x-axis. Note the characteristic "streaking" in this Figure. This is a result of resent packets. When the victim does not receive an answer to it's SYN/ACK, it waits a small amount of time and then assumes the packet was lost in transit and resends the packet. It repeats this several times, each time increasing the wait period. This results in the "streaks" in the Figure, and in an over-estimate of the number of attack packets, if this is not taken into account. We define a resent packet to be one which agrees with a previous packet in the source and destination IPs and ports, and the acknowledgment number, and which is received within 1 minute of the first such packet. The numbers in Table 3 are computed using this definition, and so resends are not counted in the definition of an attack.

Note that the resends can also be used to help determine whether the packets are backscatter from a denial of service attack, or are a scan of the protected network. One

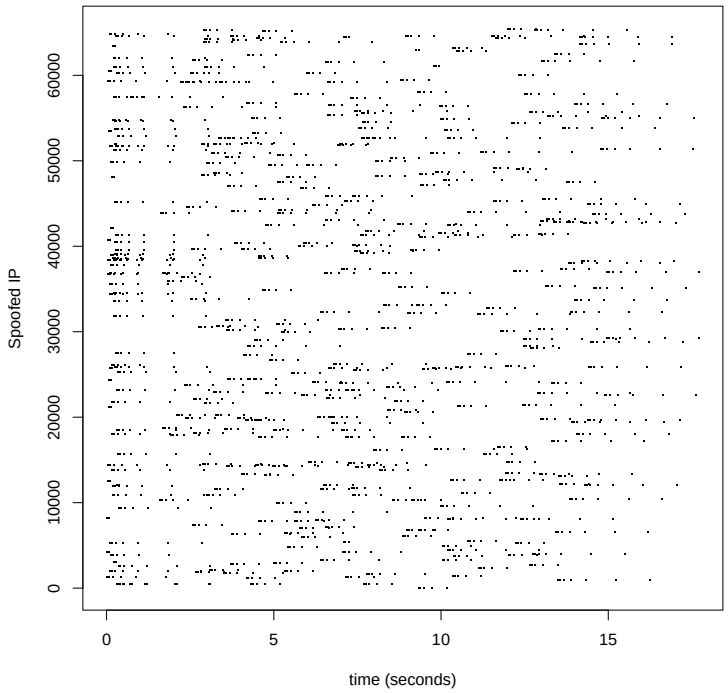

Figure 4: 2,160 packets from a single victim computer.

expects to see resends in backscatter. Scan tools that send a single packet per host/port will not show this pattern, while those that send multiple packets will typically not increase the time between packets, nor will they tend to have as large a time between packets as one sees with resent packets.

3.2 Attack Statistics

Figures 5 and 6 show histograms for the (log base 2 of the) number of packets detected in the attack, after removing resends, for the different data sets, for the two values of T. Our estimate of the number of packets in the original attack (assuming we believe that the attacker is selecting spoofed IP addresses uniformly, independently, from all 2^{32} possible IP addresses) can be obtained by multiplying the x-axis values by 16.

One observation is that the densities are surprisingly similar across all the data sets. The histograms appear to support a hypothesis of roughly three modes to the density, indicating (perhaps) the existence of three different types of attacks.

It is likely that many of the packets in the bin at 0 (corresponding to a single packet detected in the attack) represent errors in the process of selecting “unsolicited” packets

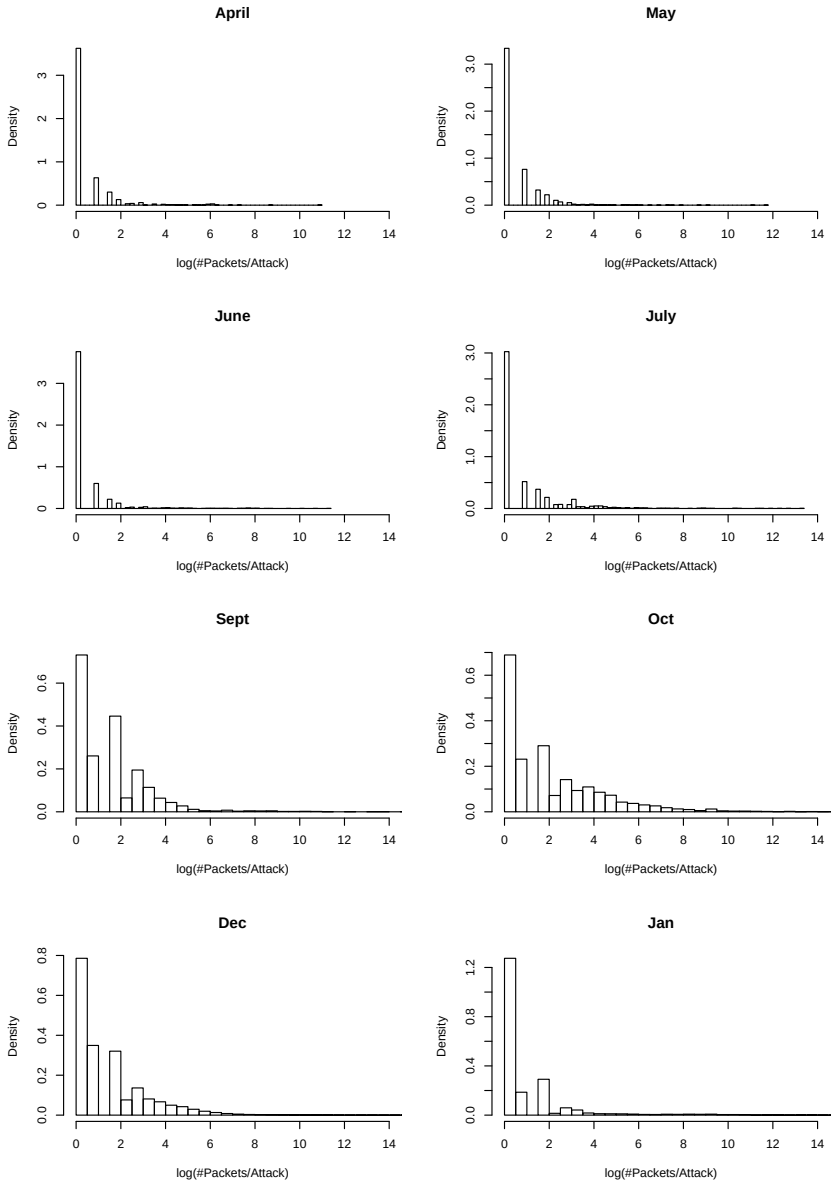

Figure 5: Histogram of the log (base 2) of the number of packets per attack. These counts are computed after the resends have been removed, as described in the text. T=5 minutes.

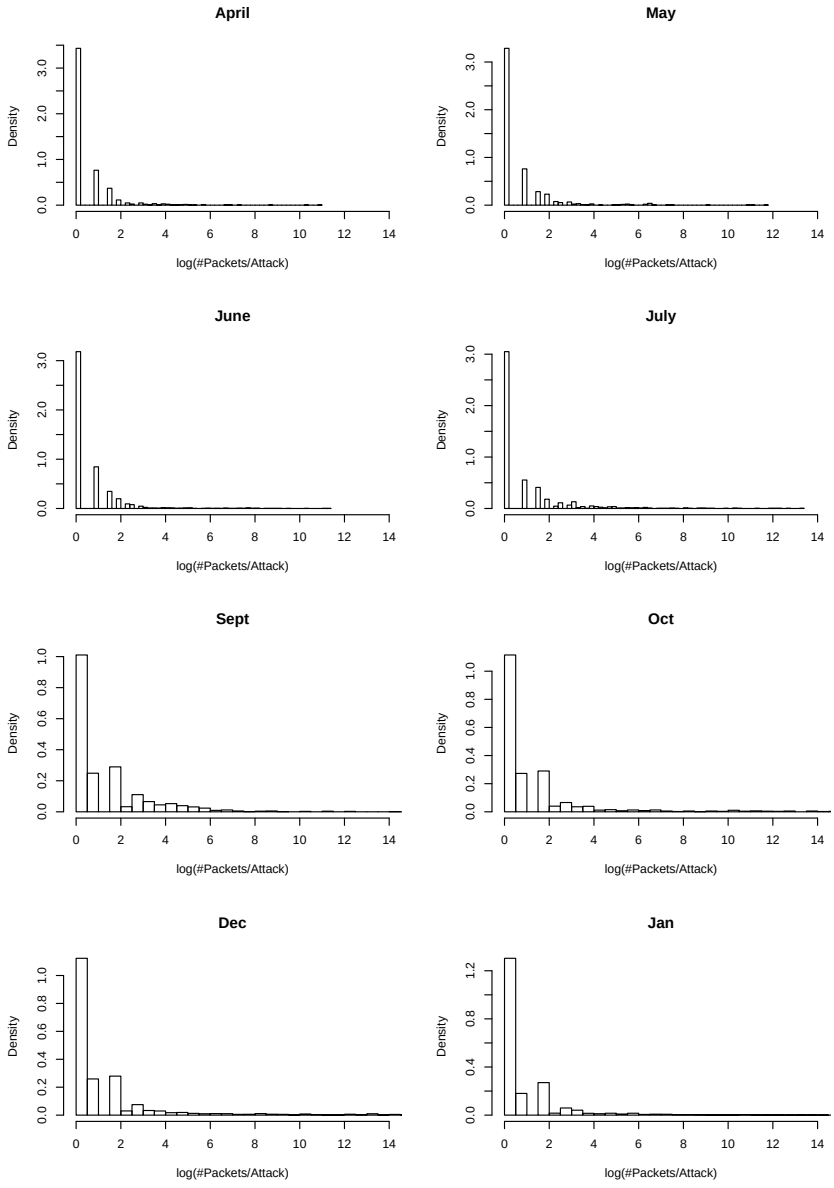

Figure 6: Histogram of the log (base 2) of the number of packets per attack. These counts are computed after the resends have been removed, as described in the text. T = 1 hour.

(for example, dropped SYN packets at the sensor), or are the results of other attacks (such as scans) or errors unrelated to cyber attacks.

Another explanation is that these are low packet rate attacks. Since a SYN flood need only fill the connection table of the victim, and keep it filled, an attack lasting only a few hours need not send more than 2^{16} packets (our estimate of the number of packets in an attack in which we observe 1 packet). Thus, it seems reasonable to suggest that for attacks against single servers (that cannot load-balance using a server farm, for example), attacks of this magnitude might be effective, and popular, accounting for the large number of such "attacks" detected.

We now turn to the question of whether the attack is random, that is, whether the spoofed IP addresses have been (uniformly) randomly selected from all 2^{32} possible IP addresses. Some pictures will be informative. While looking at pictures is subjective, and cannot detect subtle deviations from randomness, it can be very effective in detecting unexpected structure. (Note: in all the analysis which follows we use T = 5 minutes.) Figures 7 and 8 depict the packets from two victims. In these plots each packet is plotted as a dot, with x value corresponding to time and y value corresponding to the spoofed destination IP address. This is computed from the IP $x.x.y.z$ as $256y + z$. Figure 7 seems to pass a "looks random" test, while Figure 8 shows definite non-random structure. This manifests itself in two ways. First, it is obvious that the intensity of the the attack is not constant throughout the attack. Second, there is a diagonal structure detectable in the packets, showing a high degree of correlation. This attack does not satisfy our assumptions of independent random selection of spoofed IP addresses.

Figure 9 depicts attacks against two victims consisting of 9,674 and 22,716 packets. These show quite different structure, indicating several different attack tools were used. The top figure shows an attack with linear structure, overlapping an attack that looks to the eye to be fairly random. The bottom figure shows an attack with quite complicated dependence structure, with both a linear component, and some measure of clearly deterministic structure. This latter kind of attack was not observed in the data prior to the October data set.

Because of the systematic nature of the IP address selection in the bottom plot of Figure 9, the data passes a goodness-of-fit test (the Kolmogorov-Smirnov test) with flying colors. This test assumes (and does not test for) independence, and so is invalid for these data.

The above observations indicates that the blind use of goodness-of-fit tests will be of little use for these data. The changing intensity, and structure in many of the attacks make any assessment by a goodness-of-fit test problematic at best. Thus, each attack must be assessed individually, testing the different intensity regions separately. Further, it is vital that tests for dependence be used, in addition to distributional tests.

The number of large attacks (attacks with more than 1000 packets) seems to be increasing in these data. In April the average was approximately one such attack every two weeks, while by December the rate was approximately two per day. This may be a short time phenomenon (the rate does appear to have dropped to about 1 per day by January), or it may be a result of the increasing availability of attack tools or new attack paradigms. Further data is needed to assess this trend.

It might seem natural to assume that the attacks with linear structure are actually

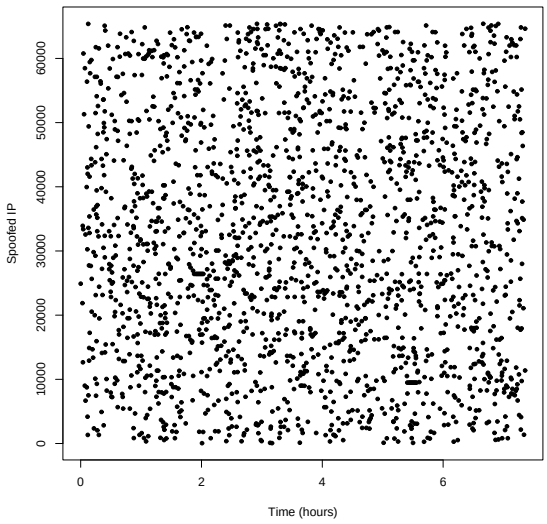

Figure 7: 1,997 packets from a single victim computer in April. The x-axis corresponds to the time of arrival of the packet, the y-axis corresponds to the last two octets of the spoofed destination IP address.

scans of the protected network, rather than backscatter from denial of service attacks. A perusal of the data shows that some of the attacks exhibiting linear structure do not have resent packets associated with them, lending credence to this hypothesis. Of the 69,773 attacks in the $T = 5$ minute data, 60,488 contained resent packets. Also, of the 248 attacks consisting of more than 1000 packets (after eliminating resends), 209 of them had resends associated with them. An alternative explanation would be that these victims have been configured to not send retries, but to rather drop the connection if an ACK packet is not received promptly. There is a technique, referred to as "SYN cookies", in which the victim encodes state information in the SYN/ACK packet, and thus does not resend packets. See

http://cr.yp.to/syncookies.html.

The case against the hypothesis that these attacks represent scans of the protected network rests on three observations: first, it is unusual to scan a network from port 80, although one could certainly do this, provided one had the permission necessary to use this port; second, the linear structure does not manifest itself as a sequential pass through the IPs in the domain, but rather, on a small scale, has an apparent random component to it; third the existence of apparent "resend" packets argues against any of the known scan tools. Thus, regardless of the actual nature of the attack, the linear structure still remains to be explained.

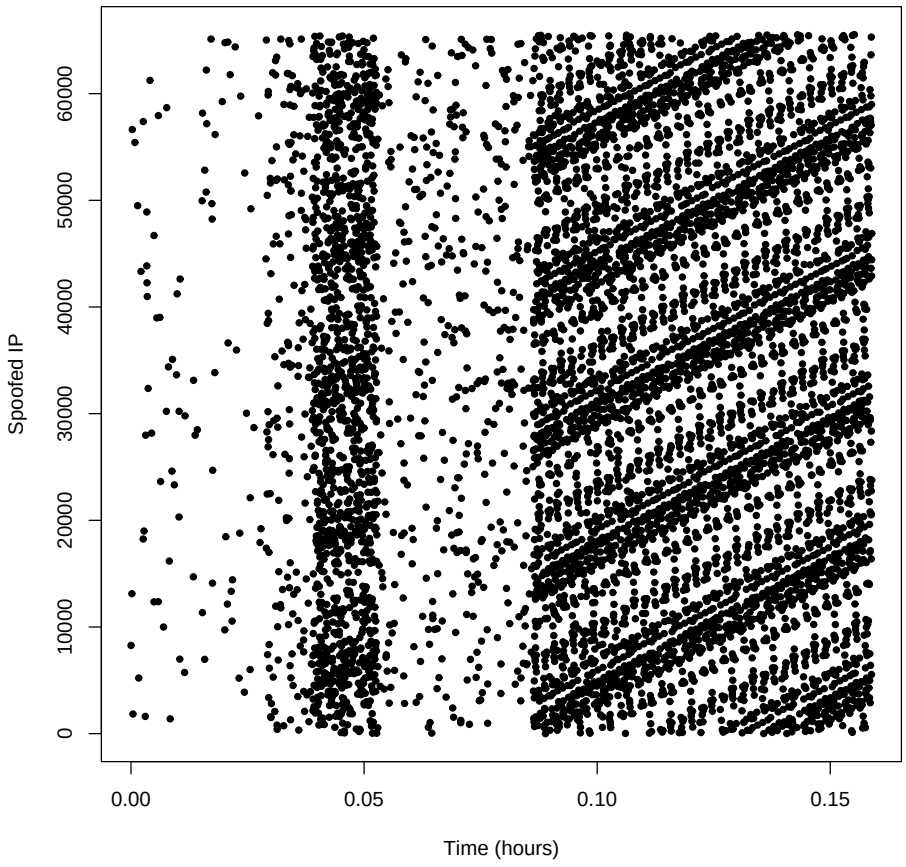

Figure 8: 7,137 packets from a single victim computer in October. The x-axis corresponds to the time of arrival of the packet, the y-axis corresponds to the last two octets of the spoofed destination IP address.

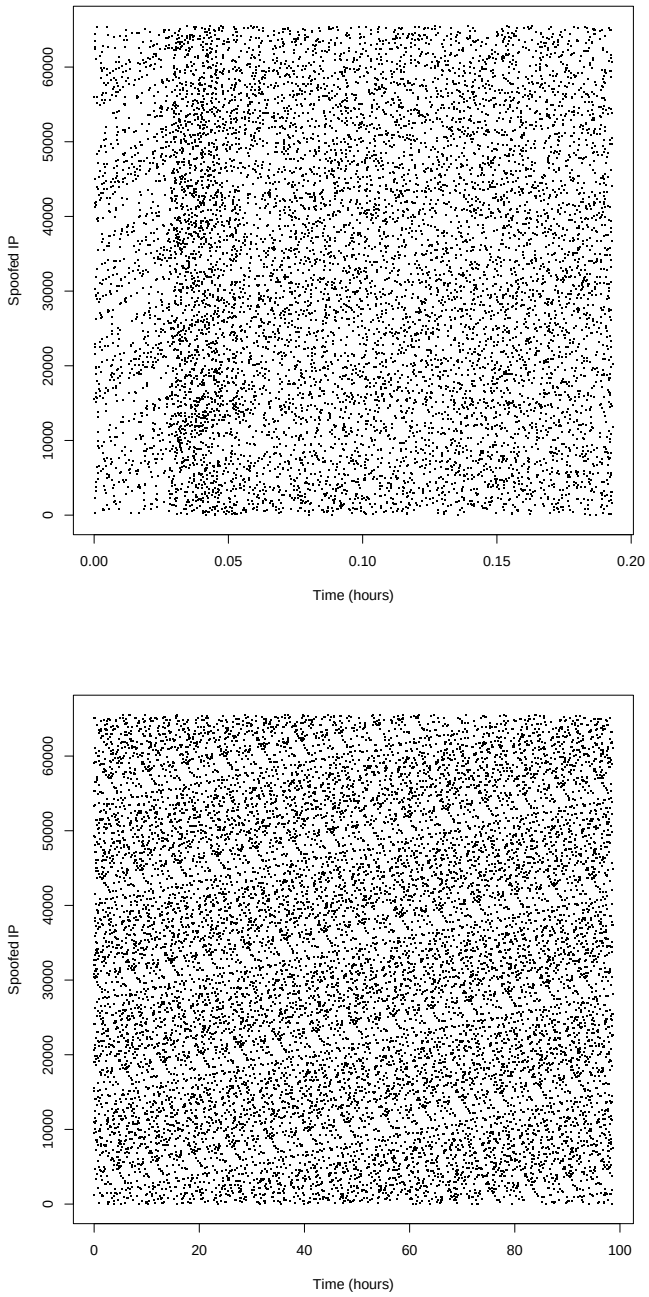

Figure 9: Attacks on two victims, showing nonrandom structure. The top figure represents 9,674 packets collected in July, while the bottom represents 22,716 collected in November.

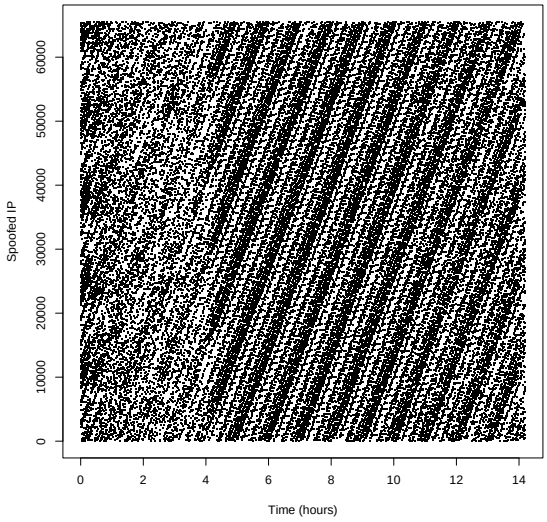

Figure 10: An attack on a United Kingdom Internet Service Provider.

Figure 10 is interesting, in that the same pattern is replicated over six different victims, corresponding to addresses xxx.xxx.xxx.3-8. This is an Internet service provider in the UK, which is obviously using a server farm to load balance. Our hypothesis is that this is a distributed denial of service attack, where different attackers received different IP addresses when the victim IP was resolved through a DNS lookup.

This raises the question of whether the linear pattern that we are seeing is an artifact of the attack or the response of the victim. Perhaps it is the load balancing that is inserting the linear structure into the attack. Perhaps the non-random IPs are a result of the timing of responses from the victim, rather than an error in the attack tool's random number generator.

As can be seen in the Figure, the character of the attack changes approximately four hours into the attack. Is this a change in the packets sent in the attack, or a change in the strategy of the victim(s)? This change occurs approximately simultaneously for all six victims, indicating that in either case the change is coordinated.

Victim action seems unlikely to be the cause, partly from the standpoint that there seems to be little value in it from the point of view of the victim, and partly from further observation of other attacks. A closer look at Figure 9 (top) reveals that there is an overlap between structured and non-structured attack patterns within the same victim. This is hard to reconcile with the hypothesis that victim response is responsible for the pattern. Thus, we believe that the pattern is a result of the activity of the attacker.

As can be seen in Figure 11, these data are highly correlated, which is hardly surprising given the pictures. One can use this information to build a model of the generating process.

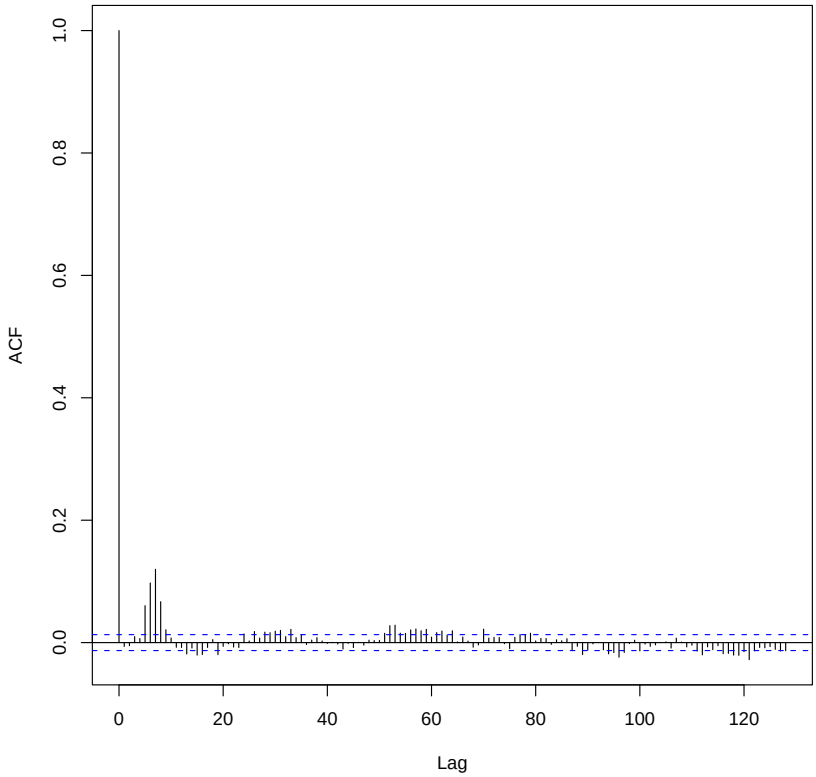

Figure 11: Autocorrelation function for the data in the lower plot of Figure 9, showing statistically significant autocorrelation.

Figures 12 and 13 provide a view of the number of attacks ongoing as a function of time. Attacks are defined as packets from individual victims, with no gap between packets of more than five minutes.

For the first four data sets, we see that while attacks occur throughout the time periods considered, there are rarely more than a few attacks at any time, and attacks typically last less than a day. There is some activity between May 10 and May 11, when there were 9 simultaneous attacks. Otherwise, the attack level is quite low.

The last four data sets show considerable activity. The ramp up in attack levels starts in mid September, and continues through to late November. At the height of the attacks there were over 30 victims under attack, and the attacks lasted for a month.

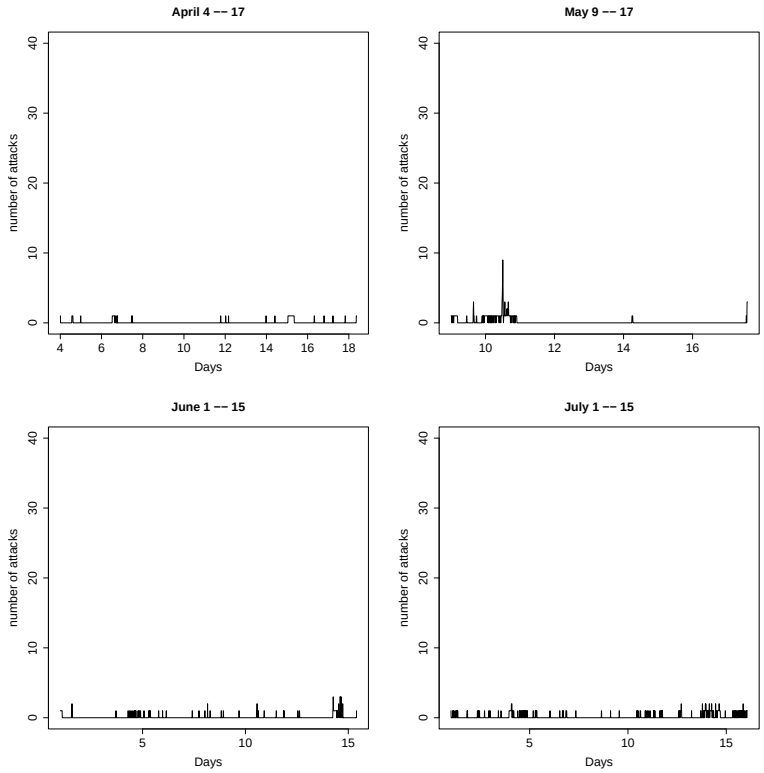

Figure 12: Number of attacks detected as a function of time.

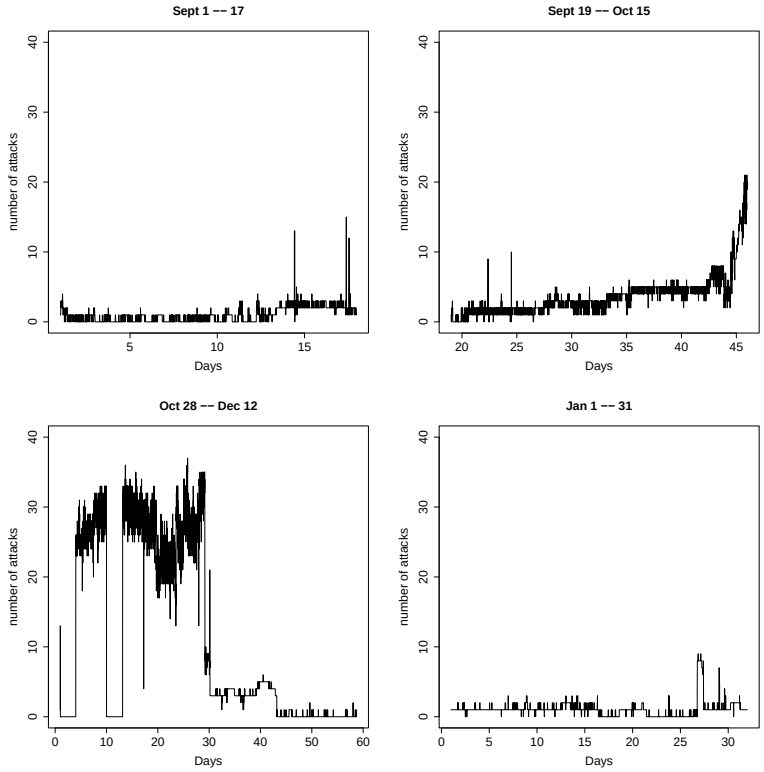

Figure 13: Number of attacks detected as a function of time. The gaps in the October plot are due to sensor drop-out.

4 Conclusions

The problem of measuring the number of denial of service attacks on the Internet is a difficult one, since many organizations are hesitant to report these attacks. Even when they do report them, they are often after the fact, and of little value for warning other potential victims of the threat. By utilizing the backscatter packets from certain classes of attacks, we have demonstrated that one can track these attacks in real-time, and we have shown that the attack level on the Internet can be quite high for extended periods of time.

Further work is needed in modeling these attacks, determining the algorithms used for generating packets, and thus providing some ability to classify the attacks. Knowing something about the attack can provide useful information for potential victims to use in defending against the attacks. Also, by monitoring trends in the attacks we can, potentially, identify when new classes of attacks are created, or when a new massive attack is underway.

The problem of determining the impact of the attack on the victim is a difficult one, which we have not addressed here. The victim machine could go down, in which case the backscatter packets would cease, but this may be indistinguishable from a cessation of the attack. It would be of value to determine whether subtle changes in the backscatter packets can be used as indications of the effect of the attack on the victim.

There are some methods available to defend against denial of service attacks, but these are not perfect and have difficulty with large distributed attacks. It would be valuable to incorporate that defense strategy into our analysis so that we could determine whether the victim is defending against the attack, and measure the effectiveness of the defense. With that said, the mere fact that we can track these attacks in real time without the cooperation of the victims and without adding to the load on the network is a powerful and useful tool. Clearly there are plenty of opportunities for statisticians to aid in the analysis of these data.

References

[Che01] Eric Y. Chen. AEGIS: An active-network-powered defense mechanism against DDoS attacks. In Ian W. Marshall, Scott Nettles, and Naoki Wakamiya, editors, *Active Networks: IFIP-TC6 Third International Working Conference*, pages 1–15. Springer, 2001. Lecture Notes in Computer Science 2207.

[Mar01] David J. Marchette. *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint*. Springer, New York, 2001.

[MVS01] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. Available on the web at www.usenix.org/publications/library/proceedings/sec01/moore.html, 2001. USENIX Security '01.

[NNM01] Stephen Northcutt, Judy Novak, and Donald McLaclan. *Network Intrusion Detection. An Analyst's Handbook*. New Riders, Indianapolis, IN, 2001.