

Analysis of DoS attack traffic data

Minho Sung, Markus Haas and Jun Xu
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{mhsung,mhaas,jx}@cc.gatech.edu

Abstract

A server at the College of Computing, Georgia Institute of Technology, was subjected to a denial of service (DoS) attack on March 27, 2001. A Cisco router's Netflow functionality was used to gather flow-level information on the attack's traffic. This paper analyzes the captured data to explore the dynamics of the attack and the characteristics of the attacking packets. We also seek to determine what tools were used to launch the attack and make some postulations as to the motivation behind the attack.

1 Introduction

Denial of service attacks have become a serious threat to the integrity of many networks and the availability of the services that they provide. Their growing frequency has made them a fact of life on the Internet. According to the 2000 Information Security Industry Survey [3], more than 37 percent of companies have experienced a DoS attack. In recent incidents, DoS attacks shut down well-known web sites such as Yahoo and E-Trade for several hours at a time, costing millions of dollars in lost business [10].

However, there are very few published studies on actual DoS attacks. One reason is that it is not always possible to capture the data on the attack since logging tools such as Cisco's Netflow can be computationally expensive to invoke on a router. Another reason is that companies that were attacked usually do not want to report the incident, let alone publish their studies on the captured data. They fear embarrassment, or the

appearance of vulnerability that could lead to copycat attacks or loss of business.

In this paper, we analyze the DoS attack traffic data captured at the gateway router in the College of Computing, Georgia Institute of Technology. We look at the data from two different angles. First, we categorize the network traffic by protocol, and study the traffic volume of different protocols during the attack. Second, we analyze the distribution of source addresses and source and destination ports of the attacking packets in an effort to identify the type of DoS program that was used in the attack.

The rest of the paper is organized as follows. In the next section, we summarize related works about DoS attacks. In section 3, we explain the network traffic we analyzed and the environment in which it was captured. Section 4 analyzes captured network traffic, and discusses what happened during the attack. Section 5 shows the distribution of the source addresses and source and destination port numbers for the attacking packets. Section 6 describes the work done to determine what tool was used to launch the attack.

2 Related Work

Denial of service attacks were first reported in significant numbers in 1996, and have been growing in number since then [12]. The most popular type of DoS attack is the TCP SYN flood attack [5]. No foolproof way has been found to stop them entirely, but both cryptographic [17, 15] and non-cryptographic [21, 13] solutions have been proposed to address them. Recently,

large-scale distributed DoS attacks have drawn considerable attention [10]. Most of the proposed defenses against them focus on IP traceback [2, 4, 7, 21, 22, 20] methods.

Research has also been done in other aspects of distributed DoS attacks. Gil proposes an attack-resistant data structure to enable routers to detect ongoing DoS attacks [11]. Zhou and Schneider propose an on-line certificate authority [24] which is robust against DoS attacks using a combination of service replication, proactive recovery and secret sharing techniques. Techniques to mitigate the effect of distributed DoS attacks have been proposed in [18] and [16].

Other proposals attack the DoS problem by calling for the tightening of global network security. Internet sites are urged to install intrusion detection software that is able to detect the attempt by an adversary to install DoS attack programs on their computers, thereby draining the supply of potential attackers [14]. Also, egress filtering [9] at every ISP is recommended to detect and drop packets sent using spoofed IP addresses. It is unclear whether these ideas will be to be practical in the near future since they require global cooperation.

To quantitatively approximate DoS attack activity on the Internet, Moore proposes a new technique called backscatter analysis [19]. He uses the response packets generated by the victim of an attack to infer the number of DoS attacks occurring on the Internet. The most important assumption in his paper is that attackers spoof source addresses randomly. In other words, the spoofed source addresses are uniformly distributed across the entire IP address space. One of our contributions in this paper is to show that Moore's assumption is supported by actual DoS attack data.

3 DoS attack data

In this section, we describe the data we have captured and used for this study.

3.1 Contents of flow data

The DoS attack data was captured by a Cisco router's Netflow [23] functionality on March 27,

2001 from 13:23:56 to 17:16:49 (about 4 hours). The data contains 14,108,760 flow entries, containing summary information about 41,185,656 network packets, or more than 2 GB worth of network traffic. An apparent TCP SYN flood attack raged on from 14:53:36 to 16:19:06, resulting in 12,964,891 TCP SYN packets sent to one of our servers.

Netflow only captures essential flow level information, not the full packet, so some information is lost, but there is enough information to make some valuable conclusions from the data. Each flow record contains the following fields.

- **StartTime** : Start time of this flow
- **Sif** : Input interface index
- **SrcAddr** : Source IP address
- **Sof** : Output interface index
- **DstAddr** : Destination IP address
- **Protocol** : IP protocol
- **SrcPort** : TCP/UDP src port number
- **DstPort** : TCP/UDP dest port number
- **Packets** :Packets sent in duration of flow
- **Octets** : Octets sent in duration of flow
- **ToS** : IP type of service
- **Flag** : TCP flags (union of all flags)

3.2 Environment for data capture

The router used to collect the Netflow data does the filtering for the College of Computing's incoming traffic. The data generated records all of the incoming packets, not just those which would pass through the router's ACL and reach the intended victim. This allowed us to see several interesting artifacts in the data which may be features of egress filtering at the originating site. We will discuss the implications of this more fully in section 5.2. The attacked machine was a Sun Sparc 10 with a 10 megabit ethernet interface to the network. The characterization of the

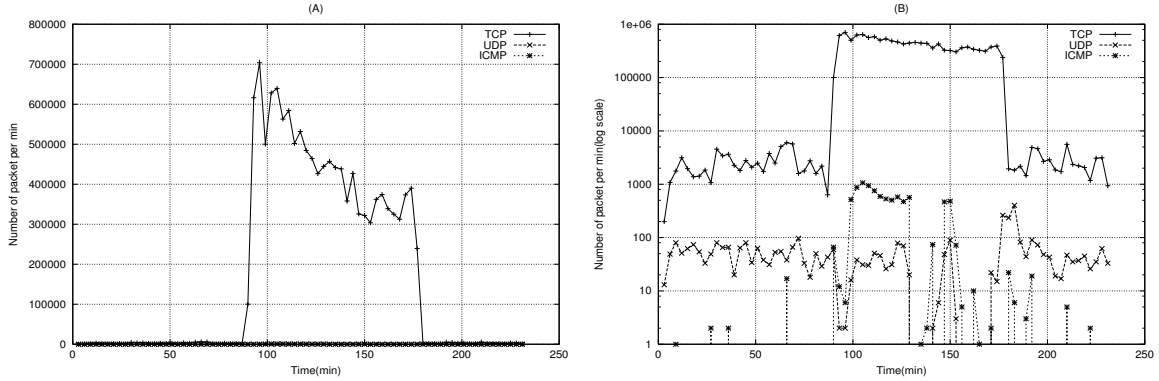


Figure 1: Incoming traffic components

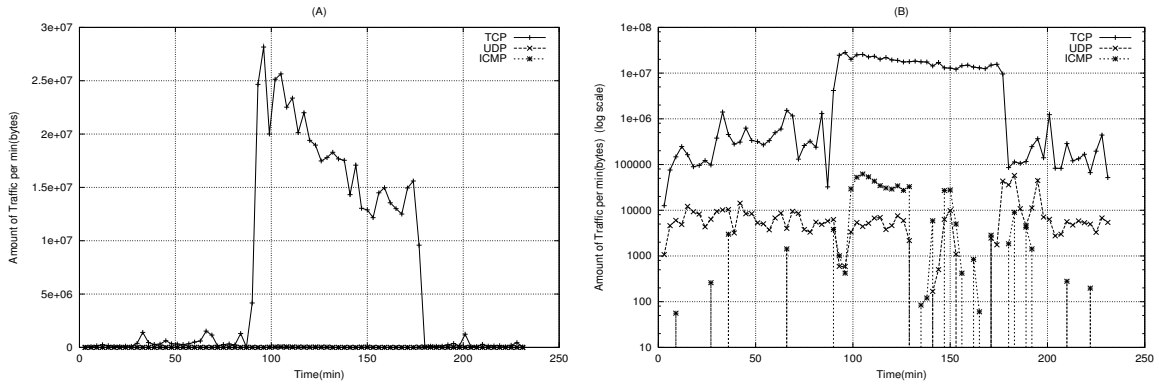


Figure 2: Incoming traffic directed to the Victim

machine's duties is interesting, as it may provide some insight as to the motivation behind the attack. It is one of a pair of machines which are intended for any non-schoolwork related use that students want to put them to. Any undergraduate student can have a shell account if he requests it. The services seen running on this server are all of the typical services such as telnet, ssh, and ftp, but not a web server. Students are free to run their own services not requiring root privilege, so various IRC and chat servers are also sometimes seen running on this machine.

4 Network Flow

In this section, we describe the symptoms of the attack inferred from the network traffic. We divide the network traffic into three components. The first component is incoming traffic destined

for the victim. The second is outgoing traffic originating from the victim. The third component is neither sent nor received by the victim, and is not considered as involved in the attack. However, it is still important to study it since it is indirectly affected by the attack.

For convenience, we will use the elapsed time from the start time of the flow capture (March 27, 2001 13:23:56) instead of the absolute time to draw time graphs. Also, graphs in this paper will be drawn by two methods: normal graphs and Y-axis log scale graphs to show the network traffic dynamics more clearly. In the figures, graph (A) shows the normal graph and graph (B) shows the log-scale graph.

Figure 1 (A) shows the three components of the incoming traffic, the number of TCP, UDP and ICMP packets directed to the victim. Figure 1 (B) shows the same graph, but drawn with a

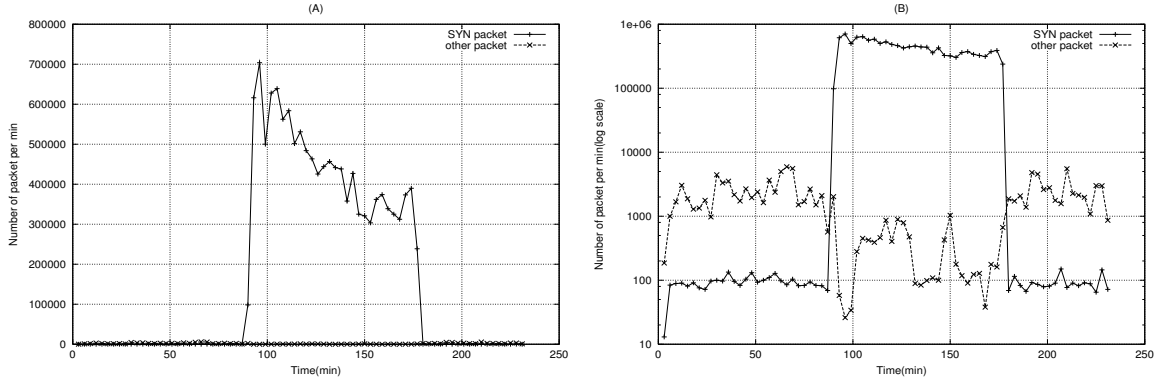


Figure 3: SYN packets vs other packets in incoming TCP traffic

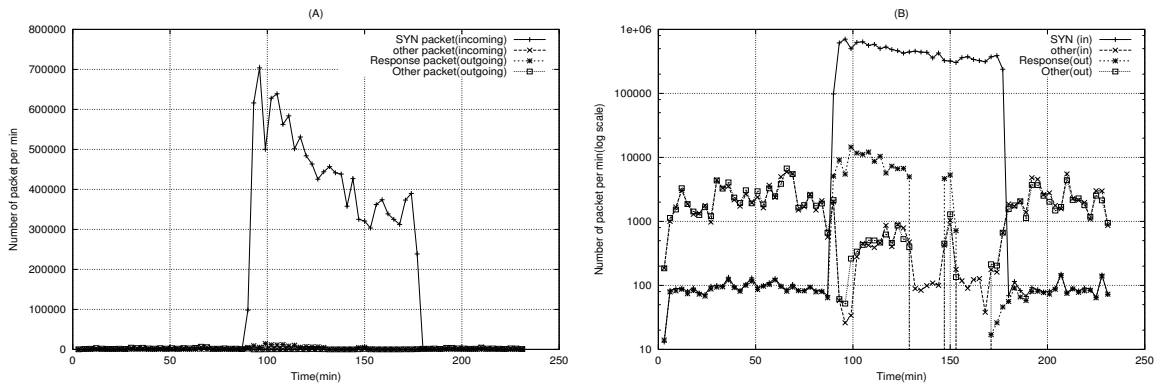


Figure 4: Response(SYN+ACK or SYN+RST) packets in outgoing TCP traffic

log scale Y-axis to show the UDP and ICMP traffic more clearly. The X-axis in both graphs represents the elapsed time in minutes.

We can see from the figure that there is a large traffic spike between 90 min and 176 min (March 27, 2001 14:53:36 to 16:19:06), which indicates the occurrence of a DoS attack during that period. The number of TCP packets increases abruptly, and the vast majority of them are SYN packets. ICMP traffic also increased, presumably caused by ‘Destination Host Unreachable’ messages created in response to packets with non-routable source addresses. As we will see in Section 5, the IP addresses the attacker (or attackers) spoofed are randomly generated without regard for non-routable, multicast, or unassigned netblocks. We cannot identify the type of ICMP traffic in the captured data because that information is not recorded in the Netflow [23] data.

Figure 2 shows the amount of traffic in bytes whereas figure 1 shows the number of packets. These two graphs have almost the same shape which indicates the proportional relationship between the number of the packets and the amount of the traffic. So, in future graphs, we will only show the number of the packets.

We can see the dynamics of the incoming TCP traffic in figure 3. This shows the number of SYN versus other packets. With the increase of SYN packets, other packets are drastically decreased. This is one of the symptoms of the attack. The server is so overwhelmed by attempting to handle the bogus connections that legitimate connections cannot be maintained.

Figure 4 shows the response packets induced by SYN packets. The victim’s response packet has the (SYN+ACK) flags set if the destination port of the incoming SYN packet is open. Other-

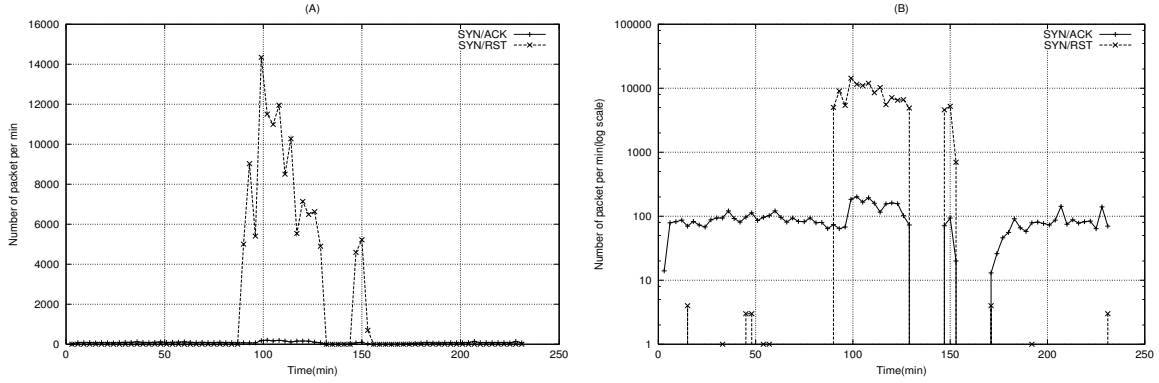


Figure 5: Response(SYN+ACK or SYN+RST) packets

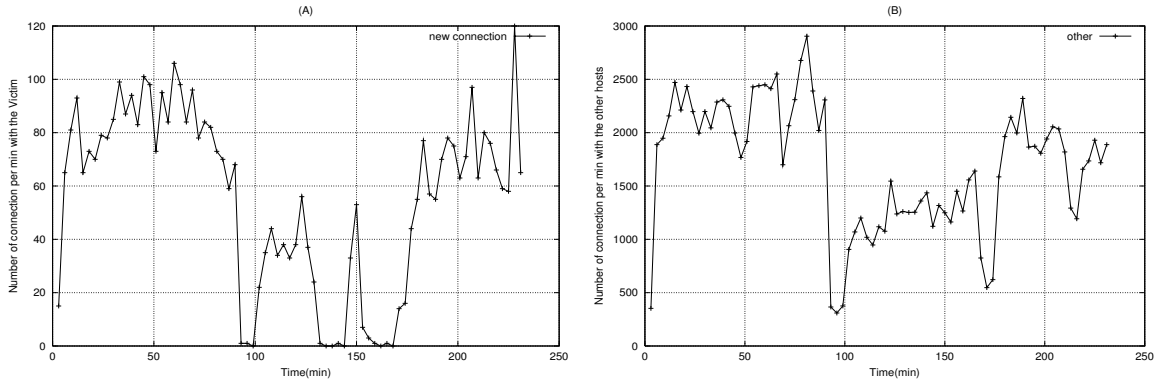


Figure 6: Number of new connection per min

wise, the victim responds with a packet that has the (SYN+RST) flags set. We notice that the victim responded to almost every SYN packet before the attack, but after the start of the attack the victim can only respond to a small percentage of them. After the attack, the number of response packets returns to normal.

Figure 5 represents the rate of (SYN+ACK) response versus (SYN+RST) response. Because the number of open ports is small relative to the whole possible range of port numbers, the number of (SYN+RST) responses is much greater than the number of (SYN+ACK) responses during the attack. During normal traffic flow, this situation should never occur. The College of Computing's network staff did attempt to mitigate the attack's effect by dropping all traffic to the host during parts of the attack, which is shown in this view by the times where there are

no responses from the server.

Finally, figure 6 (A) shows the number of new TCP connections per minute between the victim and the other hosts, and (B) shows the number of new TCP connections per minute between a non-victim host and another machine (the third component). A new connection is defined as a flow which has four or more TCP packets exchanged. We can clearly see that the number of new connections between the victim and the other hosts is significantly decreased during the attack, even while the host is still actively connected to the network. Also, figure 6 (B) shows that the DoS attack also affects connectivity to other hosts on the victim's network.

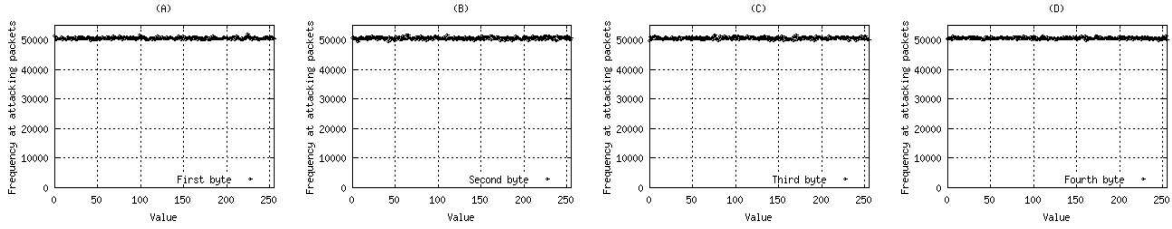


Figure 7: Source address distribution at each byte

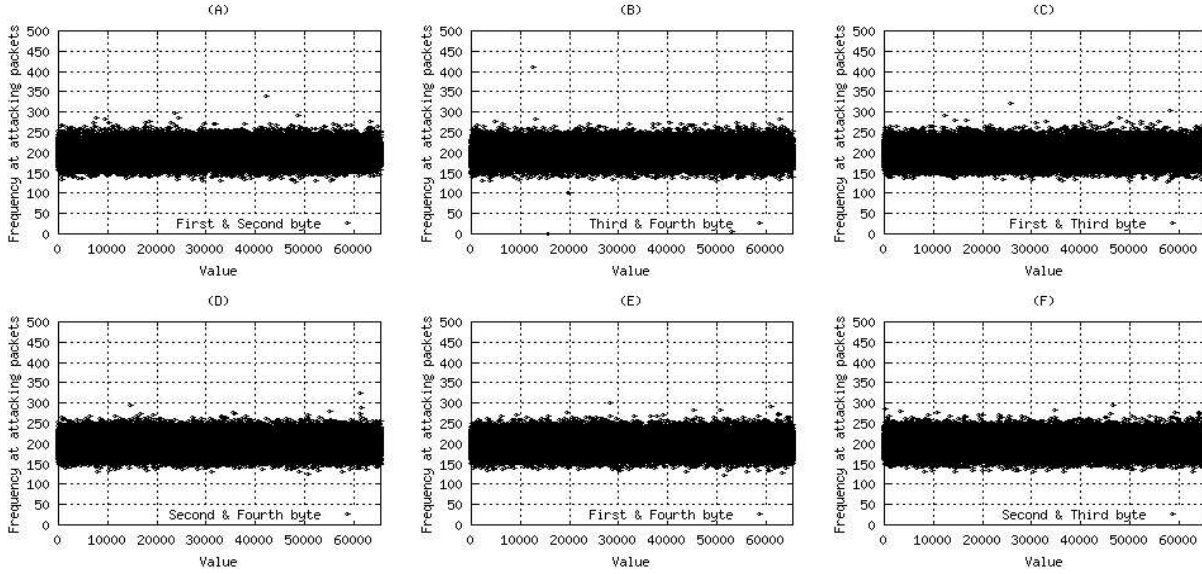


Figure 8: Source address distribution at combination of two bytes

5 Address uniformity

During the period of the DoS attack, the victim received a tremendous number of SYN packets. In this section, we analyze the distribution of the source addresses, and the source and destination port numbers of the attack’s packets. The result of our analysis is that these three values are uniformly distributed throughout their possible ranges, with the minor exception of some destination port numbers that are missing, possibly from egress filtering at the source network. This shows that the values for three parts are randomly generated by the attacking program.

5.1 Source address distribution

It is hard to show the distribution of the source addresses used for spoofing because the entire IP

address space is huge relative to the number of used addresses. There are 12,964,891 SYN packets received, which represents only 0.3 percent of the IP address space. So, we divide the 32 bits IP address into four bytes, and show each byte’s distribution. We also show the address distribution on six two-byte combinations: (first, second), (third, fourth), (first, third), (second, fourth), (first, fourth), and (second, third) to further demonstrate the uniformity of the distribution.

Figure 7 (A),(B),(C) and (D) show the distributions of the first, second, third and fourth bytes of the IP addresses used for attacking, respectively. All four graphs show the same uniform distribution. This indicates very strongly that the values for each byte are randomly selected. The fact that the distribution of the first byte is uniform means there is no consideration

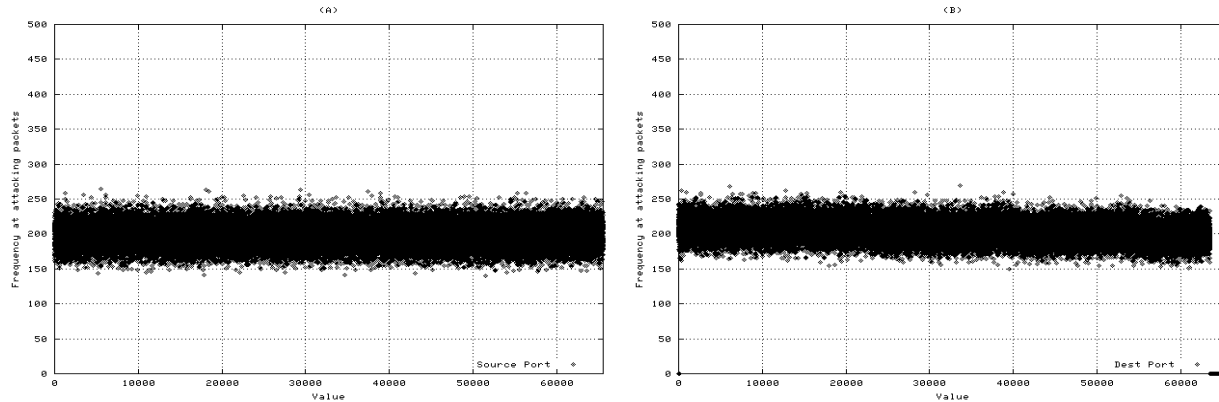


Figure 9: Source and destination port number distribution

made as to the IP address class when the attacking program generates the spoofed address.

Figure 8 shows the six graphs for the distribution of the six combinations of the two bytes each. Again, all these graphs show a uniform distribution, which is convincing evidence that the packet addresses are generated randomly.

5.2 Port number distribution

Figure 9 (A) and (B) represent the distribution of source and destination port numbers, respectively. The spoofed source ports are uniformly distributed across nearly the entire range of possible values, 0 to 65535. There are no SYN packets with destination port numbers 0, 111, and 63501 to 65535. This is not caused by any filtering done at the receiving end - the router recorded the Netflow trace data before applying any local filtering rules. This suggests some sort of egress filtering at the source of the attack. Filtering port 0 and 111 (sunrpc) are reasonably well known procedures, but the high port range being missing does not correspond to any well-known security mechanism, nor are the ports in that range assigned to any particular application.

6 Identification of software and motivation of the attack

6.1 Source program determination

There are many different denial of service tools available to those wishing to launch an attack. From the data in the Netflow traces, it is possible to reconstruct most of the characteristics of the packets that can be used to match them up with the DoS tool that sent them. The most useful characteristic in determining the source of the packets was the source IP distribution. At the time of the attack, there was no released Microsoft operating system available that allowed the user access to raw sockets needed to spoof the originating IP address of an outgoing packet. That fact alone dismisses a large number of available attack programs. The fact that Unix-based hosts were responsible for the attack suggests that the attacker is more sophisticated and knowledgeable than the average cracker launching denial of service attacks, and possibly controls more than one host that can be used for denial of service attacks. That, and the fact that the rate of incoming SYN packets at the peak of the attack suggests that more than one host was used to launch the attack, makes it likely that a distributed denial of service tool was used.

Another characteristic of the attack that can be used to narrow the search is the lack of an increase in incoming UDP and ICMP packets. Many tools, both with and without distributed

capability, use UDP or ICMP, in some cases concurrently with TCP. These tools can also all be removed from consideration. Other characteristics of packets generated by various tools that were used for disqualifying them from consideration were packet length, flags (having URG, PSH, RST, or NUL set), limited source and destination port range, or non-random source addresses. The only package found which matched all of the characteristics seen is Stacheldraht [6, 8], one of the newer distributed denial of service tools that is quite feature rich, difficult to detect, and easy to control from a central location.

It must be noted that the distribution medium for Stacheldraht and other attack tools is usually C source code, which the user must compile himself to get a working tool. In the case of distributed denial of service tools such as Stacheldraht, it also allows the user to compile in a password to exchange between client and server when commands are given to the compromised attack hosts. It would seem that the ability for anyone to make modifications to any tool calls into question the ability to correctly determine which tool was used to launch that attack. However, one author's contact with a person familiar with the community that launches such attacks confirmed that few, even among those knowledgeable enough to compromise Unix hosts, fully understood the workings of the code that they were running [1]. Therefore, it is not likely that one tool was modified in such a way that made its packets appear to be from another. A greater possibility for error lies in the possibility that the tool used to launch the attack was not found in the author's searches of attack tool repositories. Just as not all viruses written become major menaces, not all attack tools that are created achieve widespread distribution.

6.2 Motivations for the attack

It is unknown exactly what profile of attacked targets fit because of the reluctance of many organizations to reveal that they have been attacked. From what is known, it is likely that there are several motivations for attacks. The first is attacks on major corporations on whom

an attack would be very noticeable and generate much publicity. Among victims of such attacks are Yahoo E-Trade, and Microsoft. A second class is attacks that are politically motivated. Web sites belonging to such controversial entities as the PLO and offices of the Israeli government have frequently been the targets of denial of service attacks. A third motivation for attacks is more personal. Individuals may launch attacks based on perceived slights or simply as jokes. Evidence to support this comes from IRC transcripts provided by one author's contact, which show one person making a joke at the expense of another, and then the second person stating that he is launching a DDoS attack against the first person's IP address [1]. Those attacks are generally not as intense compared with the first two categories of attacks, and are usually not maintained for very long. Another bit of evidence for disputes being settled with denial of service attacks comes from one author's previous employment at a firm specializing in server hosting. Frequently, when a customer's web server came under attack, conversations with the customer revealed that he had some idea of who was behind the attack and why. The machine against which the attack considered in this paper was conducted is a relatively old, low-profile server on a college network. It hosts no important services, and had no possibility of gaining the attacker any notoriety from his actions, so the first two motivations for denial of service attacks do not apply here. The third is much more likely to be a factor. Discussions with several people who host or have knowledge of IRC servers indicate that they are frequently the targets of denial of service attacks. Since students are free to, and do, run IRC servers on this machine, it is a possibility that a perceived slight on an IRC channel is what led to this attack.

7 Acknowledgements

We would like to thank Peter N. Wan, a Research Scientist in the College of Computing's Computing and Network Services group, who provided the captured data and analysis tools. We would also like to thank Dan Forsyth, Associate Di-

rector of the Computing and Network Services group for his comments and suggestions.

8 Conclusion

In this paper, we used Netflow data from a Cisco router to analyze the traffic generated as part of a SYN flood denial of service attack on a server in the College of Computing at the Georgia Institute of Technology. An analysis of the spoofed source addresses, source and destination port numbers, as well as other packet characteristics such as flags and packet size were used to determine the program used in the attack. This data was also used to support some assumptions made in previous research in the field regarding distributions of source addresses. Also, the characteristics of the traffic flow and the choice of attacked host were used to suggest possible motivations for carrying out the attack. This attack was likely carried out using the Stacheldraht distributed denial of service tool, and there is a good possibility that the cause of the attack was an interpersonal issue, rather than an organized attempt to silence the function of the server that was attacked. Hopefully the information presented in this paper can contribute to the effort to find ways to counter denial of service attacks. The complete lack packets with destination ports 0, 111, and 63501 through 65535 suggests that all the compromised hosts involved in this attack are on the same network, as finding that egress filtering in multiple separate networks is quite unlikely.

References

- [1] T. Rodery. Interview conducted October 2001.
- [2] S. Bellovin. Internet Draft: ICMP Traceback Messages. Technical Report, Network Working Group, March 2000.
- [3] A. Briney. Information Security Industry Survey *Information Security*, 5(1):40–68, September 2000.
- [4] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. In *Proc. Usenix LISA 2000*, December 2000.
- [5] CERT. TCP SYN flooding and IP spoofing attacks Advisory CA-96.21, September 1996.
- [6] CERT. Denial-of-Service Developments Advisory CA-2000-01, January 2000.
- [7] D. Dean, M. Franklin, and A. Stubblefield. An algebraic approach to IP traceback In *Proc. NDSS 2001*, pages 3–12, February 2001.
- [8] D. Dittrich. The “stacheldraht” distributed denial of service attack tool. Available at <http://www.washington.edu/People/dad/>, December 1999.
- [9] P. Ferguson. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. Technical report, January 1998. RFC 2267.
- [10] L. Garber. Denial-of-Service attacks rip the Internet. *IEEE Computer*, 33(4):12–17, April 2000.
- [11] T. Gil and M. Poletto. MULTOPS: a data-structure for bandwidth attack detection. In *Proc. 10th Usenix Security Symposium*, August 2001.
- [12] J. Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, August 1998.
- [13] Checkpoint Inc. TCP SYN Flooding Attack and the FireWall-1 SYNDefender. Available at http://www.checkpoint.com/products/security/firewall-1_access.html, 1997
- [14] Inc ISS. *Distributed Denial of Service Attack Tools*. Internet Security System, 2001. <http://www.iss.com>.
- [15] A. Juels and J. Brainard. Client Puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proc. of NDSS'99*. Internet Society, March 1999.

- [16] F. Kargl, J. Maier, S. Schlott, and M. Weber. Protecting Web Servers from Distributed Denial of Service Attacks. In *WWW-10*, May 2001.
- [17] P. Karn and W. Simpson. Photuris: Session-Key Management Protocol. IETF, March 1999. RFC 2522.
- [18] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. Technical report, ACIRI and AT&T Labs Research, February 2001.
- [19] D. Moore, G.M.Voelker, and S.Savage. Inferring Internet Denial-of-Service Activity. In *Proc. 10th Usenix Security Symposium*, August 2001.
- [20] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proc. ACM SIGCOMM 2000*, pages 295–306, August 2000.
- [21] C. Schuba et al. Analysis of a Denial of Service Attack on TCP. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1997.
- [22] D. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. In *Proc. Infocom 2001*, April 2001.
- [23] Cisco Systems. White paper: NetFlow services and applications. Available at <http://www.cisco.com>, June 2000.
- [24] L. Zhou, F. Schneider, and R. Renesse. COCA: A Secure Distributed On-line Certification Authority. Technical Report 2000-1828, Dept. of Computer Science, Cornell University, December 2000.