Proceedings of the 2002 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 17–19 June 2002

What do we mean by Network Denial of Service?

Clay Shields
clay@cs.georgetown.edu
Dept. of Computer Science
Georgetown University
Washington, DC, 20057

Abstract— **Recent network denial-of-service attacks have brought about awareness of the vulnerability of increasingly important network services. While denial of service is not a new problem, and some of the network aspects of denial of service have been addressed, there is currently no unifying definition of what constitutes network denial of service. The goal of this paper is to propose a definition of network denial of service, and to demonstrate a simple network model that can be used to construct a taxonomy of network denial-of-service attacks. This taxonomy provides a means of categorizing existing attacks and demonstrating how future attacks might be constructed, as well as providing a simple and precise way of describing attacks.**

I. Introduction

While members of the research community have long been aware of the existence of network denial-of-service (NDoS) attacks, the problem is under-represented in academic literature. Indeed, no explicit definition of network denial-of-service seems to exist. Different authors, either in presenting possible attacks or examining real ones, have identified a number of different methods of denying service across the network. Some authors view denial-of-service attacks solely as an attacker's consumption of resources that prevents legitimate users from using those resources [1], [2]. Others present attacks that deny service by causing network devices required for packet delivery to function incorrectly [3]. Still others present attacks that can result in denial of service when information required for proper operation is corrupted or not available [4], [5]. While each of these attacks clearly result in services being denied, they seem only related in results, rather than in structure.

The purpose of this paper is to propose a definition of what constitutes a network denial-of-service attack, and to put forth for consideration a simple taxonomy of denial-of-service attacks that both includes known attacks and shows where new attacks may be discovered.

The next section presents a more detailed overview of work in the area. Section III presents a model of the network including its constituent devices and the resources they provide, and is followed in Section IV with related work on denial of service as it has been characterized within a single system. Section V contains the proposed definition of network denial of service and the taxonomy of possible attacks.

II. Background

Each published view of network denial-of-service is somewhat different. Some authors view the problem in terms of resource consumption, and this primarily at end systems rather than in network devices. Other authors examine the effects of the propagation of incorrect information, such as bad addresses or routing updates, or the effects of network devices that do not follow the proper protocol for their operation.

The widely publicized distributed denial-of-service attacks against Yahoo and other major on-line companies in February of 2000 were not the first such attacks against commercial sites to take place across the Internet. As early as 1996, some Internet service providers were being affected by a network denial-of-service attack known as *SYN flooding.* NDoS continues to be a problem today [6]

In a SYN flooding attack, an attacker sends connection request packets, known as SYN packets, to a particular host and service. These packets consume memory at the victim as it must store information for each pending request. The amount of memory dedicated to storing information about pending connections was often fairly small, so it was easy for an attacker to disrupt normal operation of the system.A full description and solution for this attack was published by Schuba, *et. al.* [1]. This attack worked by consuming a specific limited resource in the end host: the amount of memory available to store connection requests.

Meadows was the first to attempt to formalize network denial-of-service attacks based on resource consumption [2]. This work examined the ability of an attacking system to send messages that would result in resource consumption by the recipient, and proposed a framework for protocol designers to follow to determine the tolerance of their protocol to denial-of-service attacks. Meadow's model focuses primarily on the costs of protocols that are incurred between end systems, and should be useful for examining protocols

ISBN 0-7803-9814-9/$10.00 ©2002 IEEE

that operate at the highest layer of the network. While it does not focus on attacks that can consume resources in devices that provide network-level services, such as simple packet forwarding, it also seems that the model could be easily adapted to demonstrate the efficacy of such attacks. It is not clear, though, that the type of solution proposed for designing protocols could be easily applied to lower network levels, given that network devices are not really participants in a two-way, higher-level communication. Additionally, while the notion of cost is used to good effect, the model does not attempt to define what the costs actually might be in an actual network, in terms of what resources are available for consumption.

Other work has examined different types of network denial-of-service that are not based on consuming resources. Instead the denial of service aspects arise from the corruption or unavailability of information needed for proper provision of service, or from improper function of the network devices that provide the basic functionality of the network. The vulnerability to the first type of denial of service is alluded to in papers that cover the Domain Name System (DNS) and the Address Resolution Protocol (ARP) [4], [5]. While neither paper addresses NDoS directly, the authors point out that disrupting the mapping may result in packets being misdelivered. This might result in data being delivered to the attacker or some accomplice. While the disclosure of data this way might be a serious problem, there is also a denial of service aspect involved, because if packets are delivered to the incorrect destination, they are not being delivered to the correct destination.

An example of incorrect operation of routing devices was presented by Cheung and Levitt [3]. They examine the problem of how to identify a misbehaving network router and locate routes around it. These authors cite examples of inadvertent denial of service that occurred when erroneous routing updates were transmitted by a faulty router. The authors show that a malicious router could have caused the same effect. The authors also detail other attacks a single router could use to cause denial of service, like purposefully mis-routing or dropping packets.

Needham was the first to examine the effects of denial-of-service attacks at the application layer, focusing primarily on end-to-end solutions for a particular application [7], [8]. Though network denial of service is considered, the types and effects of attack are not clearly delineated, and the recommendations for defending against them are specific to the application chosen. Needham does recognize that it is possible to achieve denial of service by corrupting information as well as consuming resources, providing an early start towards a comprehensive understanding of the problem.

Another inclusive view of network denial-of-service attacks was taken by Ptacek and Newsham [9] in their discussion of methods of foiling intrusion detection systems. They cite a number of different resource-consumption attacks, including attacks that exhaust memory, bandwidth and CPU resources. They also point out that systems that react against intrusion attempts by blocking out the attacker might be able to be tricked into reacting against an innocent victim — an example of an information corruption attack. The intent of this work is not to define NDoS, though, and while examination of possible denial-of-service attacks against a particular network device is useful for those designing or operating that device, it does not provide a complete view of NDoS for an entire network.

III. Network Model

A computer network is made up of a number of different devices that cooperate to provide services to members of the network. The model we will use reflects the devices and operation of an IP network, though a similar model could be constructed for other types of networks as well. There are several types of devices in this model. At the network edges there are *hosts*, which run applications that use the network to send and receive packets and provide services to other end hosts. *Routers* are individual devices that exist to forward packets between end hosts, make up the forwarding infrastructure of the network, and do not run applications that provide services but which can generate packets for the purposes of controlling the network. *Switches* are similar to routers but switch packets based on link-layer rather than network addresses. For improved security, some subnets are protected by *firewalls*, which are generally hosts that perform packet forwarding only after applying some filtering rules, and which might perform a proxy function, forwarding application-level data for end hosts.

Figure 1 illustrates a small sample network illustrating the placement of these devices. In this diagram, there are hosts I, R, A and H. They communicate across the network links, a, b, c, d, e, f, g, and h, between routers 1, 2, 3, and 4. Additionally, there is a firewall, FW, and a switch s in the network. Notice that each host is connected to the network over a shared link, though other hosts on the same subnetwork are not shown. In future examples, I will represent the initiator of a communication, R the responder to that communication, A an attacker, and H some innocent host.

The services these devices cooperate to provide can be modeled in a layered manner, with higher-level services being dependent on the correct operation of lower-services. The OSI model of layering [10] contains seven network layers that operate together for complete network functionality. In practice, not all layers in the OSI model are commonly implemented or used. As show in Figure 2, which is adapted from Stevens [11] and shows a diagram of the OSI model and an approximate mapping of the IP protocol suite, fewer layers are actually needed to produce a good

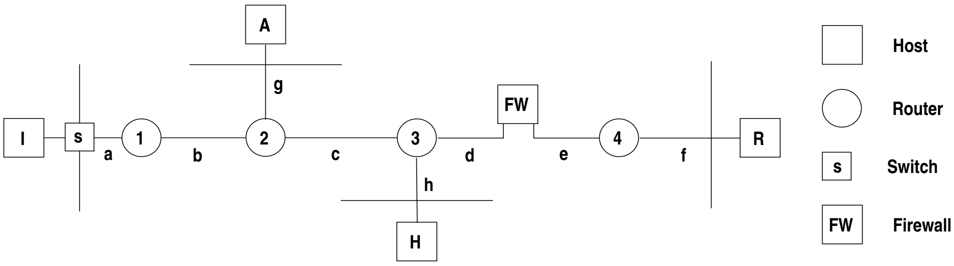

Fig. 1. Example network path

model of an IP network.

The model used in this paper is a slight simplification of the layering adapted from Stevens, and is the rightmost diagram in Figures 2. This model will have only three layers. For simplicity's sake, we ignore the device driver and hardware level. While this does limit the the ability of the model to reflect some existing attacks, these attacks are few in number and are generally limited to attackers that have direct access to the shared physical media. The bottom layer, referred to as the network layer, provides only simple packet forwarding services, and maps to the IP level of the Stevens Model. The middle layer, generically referred to as the transport layer, provides end-to-end network communication between hosts and also covers network control messages, generated by routers and internal network devices. This layer maps to the TCP/UDP/ICMP layer in Stevens. Finally, the highest layer will be referred to as the application layer, and will represent applications running on end hosts that obtain or provide service by replying on the lower network layers.

Notice that not every device provides service at each network level. For example, a switch only performs packet delivery and generally does not maintain state about or participate in any transport-level protocols. A router will generally only provide network and transport level services, though it uses application level protocols for control purposes, while end hosts will participate in all levels of service.

Additionally, there are services that do not strictly follow the layering defined above. In the Figure 2, these are DNS and ARP, which provide translation between the different types of addresses used at different levels. While ARP runs on individual hosts, the DNS system [12] employs a hierarchy of distributed hosts that perform address lookup and resolution. While the address lookup occurs at the application layer, the results returned become information that is critical to correct network operation at the network level. While we do not explicitly consider attacks against DNS server information below, the model we will construct can easily be expanded to show such attacks [13].

Each of the devices in the network has some resources available to perform their function. These resources might be specific to a particular service layer, or they might be global across all layers within the same device. In this

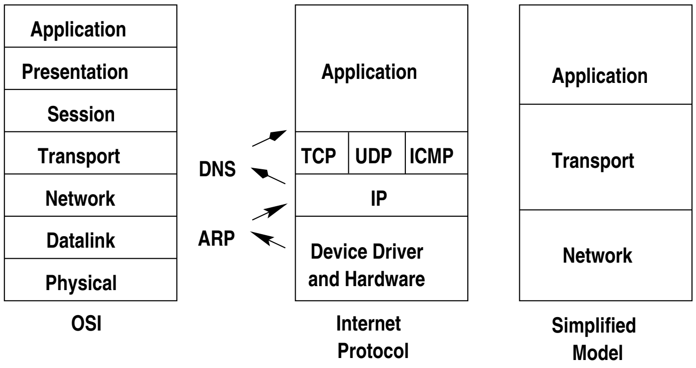

Fig. 2. Network Layering (adapted from Stevens)

model, the resources possibly available at each device are *memory*, which is volatile memory in a device used for temporary storage of information; *processing*, which represents the availability of CPU cycles to perform operations; *storage*, which represents long-term, non-volatile storage such as disk space; *state*, which represents the proper protocol code and the operating state of that protocol code; and *variables*, which is information not tied to the protocol state and which might be used for processing network requests.

Notice that *bandwidth*, which is available transmission space over some physical link, is not explicitly considered in this model. The reason for this is that while bandwidth can be consumed, IP packets are typically not dropped while they are on the transmission line. Instead, bandwidth is a limiting factor in how quickly routers can empty their transmission buffers over a particular network link. When more packets arrive at a router than can be buffered, some packets are dropped, either as they arrive or from the transmission buffers depending on the router policy. As several routers contend for transmission time over a link, each will start dropping excess packets. The resource that is the limiting factor in causing packet loss is therefore router buffer memory, rather than bandwidth. In other networks, where packet loss could occur in transmission, the model might have to include bandwidth as a specific resource.

While it is possible to more finely detail the available resources, this breakdown still allows for a reasonable taxonomy of network denial-of-service attacks that affect these resources. A more detailed model might be required to fully model all possibilities for denial of service in a particular network. For example, a wireless network may be vulnerable to attacks that drain device batteries.

IV. Denial of Service

Denial of service is not a new problem in computer science. It has been well characterized as it applies to processes in individual systems [14], [15], [16]. These models generally assume that there is some finite *maximum waiting time* (MWT) for a process to access a shared service. Denial of service for an individual system can then be defined as [14]:

"*A group of authorized users of a specified service is said to deny service to another group of authorized users if the former group makes the specified service unavailable to the latter group for a period of time which exceeds the intended (and advertised) service MWT*".

This definition has three main components: a maximum waiting time, services, and authorized users. Our objective is to see how well this definition applies to network denial of service, so we examine each component in turn.

Most network services do have a specified MWT. Often network protocols use a timer to ensure replies to messages sent are received within a reasonable period of time, otherwise a retransmission is attempted or the protocol fails. This is part of the TCP protocol, for example [17]. Other network services, most notably those trying to ensure some level of quality of service [18], do attempt to provide some guarantees about network latency or throughput, and thus could be considered to have a specific MWT. Additionally, even when protocols do not have a specified timers, there is often an implicit MWT that is dictated by either the patience of the human operator of a process or by the decreasing time value of the data.

The network provides a number of services that are best viewed as being provided by different layers in the network. It is important to notice that the services of each higher layer depend on the services provided by some lower layers. It is therefore possible to target a specific service for an attack, and consequently possible to target a lower-level service for an attack that will effect all higher-level services. Some network services are available to any system that is attached to the network and that is capable of sending and receiving packets. This is certainly true of the lowest-level service of packet delivery, in which the only barrier to sending packets is obtaining a network connection. It is relatively easy to get network access, as many companies are willing to provide it given that you present the requisite funds. There is no widely deployed method of performing authentication at the lower network layers. Though IPSec implementations [19] add this ability, the authentication checks are still done at the receiving end, rather than at the network ingress point. Therefore, even if the authentication headers are incorrect, the packets are still delivered through the network. In other cases, the legitimate provision of some network services might require authorization dependent on authentication using an Internet Protocol address, password, or cryptographic credential. This type of authentication is normally performed in applications at the highest layer of the network, and the packets are still delivered by the network even if the authorization fails. Therefore, unlike the processes in the definition quoted above, a network attacker does not necessarily have to be authorized to use the service it is attempting to disrupt. Instead, he can attack a lower-level network service that is required to operate correctly support the higher-level service which requires authentication.

Overall, the definition denial of service given above is congruent with network denial of service, but only if we consider "authorized users" to be all users who can send and receive packets using the network, regardless of any authorization that might occur for a particular higher-level service being attacked.

V. Network Denial of Service

While there are many ways to attack a network, remote attacks that use only the resources of the network itself are more interesting than attacks against the physical components of the network. Clearly, an attacker armed with a back-hoe and the location of the links between routers could easily destroy the physical communication lines, resulting in denial of service. Other types of physical or electro-magnetic attacks against routers and hosts are possible as well. These types of attacks are less interesting than remote attacks, however, as the attacker must have some physical presence to conduct these attacks, limiting the number of targets he has access to and leaving open the possibility of detection or capture using real-world investigative techniques.

The definition of network denial of service that we develop should therefore reflect the nature of NDoS attacks, which use network services to disrupt the network. It should also reflect the possibility that multiple attackers could be involved, as in a distributed coordinated attack [20], and that the attack can have multiple victims, either intentionally or as a collateral effect. It should reflect the fact that the nature of the attack will involve disrupting the legitimate use of resources, and be able to differentiate between accidental and deliberate attacks. These goals are reflected in the definition below.

Definition: **A network denial-of-service attack occurs when some set of network entities intentionally uses network services with the goal and effect of causing consumption or corruption of network resources in such a way that some other set of network entities have their ability to access otherwise usable net-**

work services degraded or so delayed as to render them unusable.

Notice that this definition is similar in structure to the one used for individual systems recounted in Section IV, though it is made specific for networks. Any host capable of sending packets can be considered an authorized user of the network. While there is no specific MWT defined across all network protocols, each individual protocol might have a timer, or the patience of a human user might be exceeded.

Additionally, this definition reflects another common aspect of network denial-of-services attacks, in that attacks can originate from and effect multiple entities in the network. That multiple entities can suffer simultaneously from a single NDoS attack should be apparent, as if the attack has its effects it some point in the network (such as at a router) it could prevent traffic from reaching all downstream entities. For example, in Figure 1, if the attacker were to send enough traffic to saturate router 2, host I would be unable to reach any of the other hosts in the network.

The common use of multiple network entities to launch an attack is to provide *amplification* of attack traffic, though it can be done for such reasons as to hide the true initiator of an attack as well [21]. An attacker at a single host who wishes to conduct an attack that will consume resources may be unable to generate sufficient traffic to deny service by itself. However, by organizing multiple hosts to coordinate attacks, the sum of all the traffic send might be sufficient for the attackers needs. This organization might be done via compromise of hosts, as seen in distributed denial of service attacks (DDoS), or it might take place by organizing the operators of multiple machines to simultaneously use their hosts to request some network service, as in a web sit-in [22]. An attacker might also obtain amplification by exploiting properties of protocols in the network [23], [24], [25].

VI. A NDoS Taxonomy

The definition and models that have been developed are sufficient for creating a taxonomy of network denial-of-service attacks. In considering each attack, we need to determine how the attack is conducted; the resource or resources consumed or corrupted by the attack; the devices in the network in which these resources reside and thus are affected; and the service layer targeted by the attack. This will allow us to see how attacks, which appear similar in mechanism, actually differ in effect. For example, both SYN flooding and DDoS attacks result from the same mechanism of a large stream of packets. However, SYN flooding consumes memory at a host, whereas DDoS attacks consume router memory resources. It is therefore important to differentiate between which resources are affected, as this relates to the notion of cost put forth by Meadows [2].

The result of each network denial-of-service attack can therefore be regarded as a 4-tuple of $\{means, effect, resource, location\}$. $Means$ is the general method that the attacker uses to cause the attack. In an IP network, as an example, this would consist of which protocol or level in the network which was being used for the attack, and any method of amplification being used. $Effect$ is the effect that the attack has on the limiting resource. In general, this will either be to corrupt or consume the resource in question. *Resource* is the resource necessary to the proper operation of the network that is being consumed or corrupted. *Location* is the particular network device where the resource being effected exists.

This taxonomy enables a succinct method of describing attacks, simply by following the 4-tuple directly. For example, the SYN attack mentioned above could be easily described as a TCP flooding attack that consumes memory at the end host. The DDoS attack is a IP flooding attack that gets amplification through multiple compromised hosts and consumes forwarding buffers at routers. This language is simple and precise. Most attacks be divided into *consumption* or *corruption* attacks. Additionally, there are some terms in common use that can be applied to specific instances of corruption. *Crashing* occurs when an attacker sends packets constructed to take advantage of errors in software that can kill a process or operating system. *Conditioning* is a corruption attack that targets machines that learn behaviors or detect anomalies by feeding them incorrect learning examples.

A. Known NDoS Attacks

The taxonomy developed is illustrated through a small number of examples, which are chosen only to demonstrate the method and not to provide a definitive list of existing attacks nor to show the relative frequency of each class of attacks. The results are shown in Table I. Note that many of the attacks listed are no longer possible due to software improvements and upgrades, but space limitations preclude a thorough discussion of the attacks. Blank areas in the table may result from one of three things: a class of attacks that is difficult to implement; an area that had not yet been explored for possible NDoS attacks; or just the failure of the author to locate of any example of an appropriate attack.

B. Determining Possible Future NDoS Attacks

Given the model and taxonomy we have developed, there are two ways to determine what new attacks might be seen. The first method is to perform a complete analysis of the hundreds of denial-of-service attacks that are in existence, and to use this analysis to fully complete the taxonomy above. Beyond demonstrating problem areas that need to be addressed in helping to prevent NDoS attacks, this method can point to area in the taxonomy that are under-

Resource	Layer	Initiating Host	Responding Host	Router	Firewall
	Network			DDoS Attack	
Memory	Transport		SYN Flooding [1]		
	Application			HTTP Proxy DoS [26]	Bordermanager Slow DoS [27], Decoy Blues [28]
	Network				
Processing	Transport		Stream DoS [29]		
	Application		Nuke Nabber [30]		
	Network				
Storage	Transport				
	Application		Mail Bombing	Routedsex [31]	
	Network		Teardrop [32], [33] Oasis [34]		
State	Transport	FIN/RST Spoofing [35], [36], [37]	FIN/RST Spoofing Land [32], [33]	Ascend Crash [38]	
	Application			Cisco Remote Crash [39]	Gauntlet Lockup [40]
	Network	ARP Cache Poisoning [5]			
Variables	Transport			RIP Spoofing [41]	
	Application				

TABLE I

A Partial Taxonomy of Network Denial-of-Service Attacks

populated relative to other areas. These sections could be underpopulated for two reasons. It might simply be difficult to cause a sufficient effect on the resources targeted to result in denial of service. For example, though Table I is simply populated with example attacks, rather than a frequency count of existing attacks, there seemed to be very few attacks that were able to effect the processing power of a router. This is likely due to the fact that routers are designed to operate correctly at line speed, which is in fact one of the defenses against flooding attacks suggested by Needham [7], [8].

On the other hand, there could be viable attacks that have not been considered. To continue with the router processing example, it might be possible to target router processing resources. For example, though routers are able to forward packets at line speed, the processing of control messages and "unusual" packets, such as those containing IP options, takes more processing power. Though most routers have separate processors for forwarding and control messages, if an attacker could send many of these types of packets, it might be possible to consume enough of the router's processing to prevent proper operation of routing updates and proper processing of unusual packets. This would not be likely to effect the router's ability to forward packets immediately, though it could cause eventual corruption of routing information if routing updates were missed.

The second method of determining possible denial-of-service attacks is to consider the path across the network from some initiator to some responder. Along that path there are a number of devices that must possess both ample and uncorrupted resources to support the communication. In the case that an attacker can consume or corrupt those resources, the communication will fail. In short, a way to look at what can be attacked is to look at what has to work; if an attacker can make something that must work fail, then the attack will succeed. We can use the same taxonomy developed above to examine what attacks are possible and some properties of those attacks.

For example, an attack can be easily designed that will attempt to cause a certain effect at a particular location. Assume that in Figure 1 an attacker at A wishes to prevent I from communicating with R, but would also like for R to be unaware that a NDoS attack was occurring. Assume that A is aware of the network topography. Examination of the figure will show that there are a number of places that A can attack. It can attempt to disrupt any of the routers $1, 2, 3$, or 4 or it can attempt to disable the firewall, as all of these devices must have adequate resources for communication to occur. In the following attack, A attempts to cause enough congestion in link c to effect the memory of routers 2 and 3.

While A is able to send packets, it is obviously a benefit if he can also exploit functionality of the network protocols to amplify the traffic sent. The attacker therefore chooses to spoof what appear to be valid IP packets from I to R, but which have the TTL set to 2. The packets will travel from A to router 3, at which point the TTL will expire, causing a ICMP Time Exceeded packet to be sent to I. Therefore, twice as many packets will be flowing from router 2 to router 3 as A sends. If this is not enough traffic, then A might attempt to coerce H into joining the attack, perhaps by breaking in and starting traffic flowing from H to I. While not as effective as a smurf attack [24], which gains a large amplification of traffic, this small amplification might be enough to cause congestion at routers 2 and 3 and to prevent I and R from communicating.

Finally, new protocols can provide other opportunities for new denial-of-service attacks. Meadows, for example, discussed the use of cryptographic signatures to verify SYN packets, and points out that to do so still might result in a network denial-of-service attack, because even though memory consumption is reduced (by not storing state about pending connections) resources are still taken in verifying the signatures [2]. Indeed, though the possibility is there, the cost of obtaining more memory seems likely to be higher than that of obtaining more processing. Meadows' point is well taken, however. With the introduction of commonplace cryptography, it may be possible to force enough cryptographic operations to occur to consume enough processing power to degrade other operations.

VII. Conclusions

This paper has proposed a definition of network denial-of-service attacks that unifies the current differing views of how denial of service can occur. In creating a model of the network that is appropriate for the definition, this paper has also provided a simple taxonomy that can be used to classify network denial-of-services. This model, while focused on exisiting IP networks, can easily be adapted to other networks as well. The taxonomy presented can be used to identify differences between existing attacks, or it can be used to identify potential future attacks.

VIII. Acknowledgments

This paper arose from discussions with Mahesh Tripunitara, and I am thankful for his questions and insight on the subject.

References

[1] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," in *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 208–223, May 1997.

[2] C. Meadows, "A Formal Framework and Evaluation Method for Network Denial of Service," in *PCSFW: Proceedings of The 12th Computer Security Foundations Workshop*, IEEE Computer Society Press, 1999.

[3] S. Cheung and K. Levitt, "Protecting Routing infrastructures from Denial of Service Using Cooperative: Intrusion Detection," in *Proceedings of the New Security Paradigms Workshop (NSPW-97)*, (New York), pp. 94–106, ACM, Sept. 23–26 1997.

[4] S. M. Bellovin, "Using the Domain Name System for System Break-ins," in *Proceedings of the fifth USENIX UNIX Security Symposium: June 5–7, 1995, Salt Lake City, Utah, USA* (USENIX Association, ed.), (Berkeley, CA, USA), pp. 199–208, USENIX, June 1995.

[5] M. Tripunitara and P. Dutta, "A Middleware Approach to Asynchronous and Backward-Compatible Detection and Prevention of ARP Cache Poisoning," in *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pp. 303–309, December 1999.

[6] D. Moore, G. Voelker, and S. Savage, "Inferring Internet Denial of Service Activity," in *Proceedings of the 2001 USENIX Security Symposium*, (Washington D.C.), August 2001.

[7] R. M. Needham, "Denial of service: an example," *Communications of the ACM*, vol. 37, pp. 42–46, Nov. 1994.

[8] R. M. Needham, "Denial of Service," in *Proceedings of the 1st Conference on Computer and Communication Security*, pp. 151–153, November 1993.

[9] T. H. Ptacek and T. N. Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," tech. rep., Secure Networks, Inc., Suite 330, 1201 5th Street S.W, Calgary, Alberta, Canada, T2R-0Y6, Jan. 1998.

[10] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987.

[11] W. R. Stevens, *UNIX Network Programming, Interprocess Communications*, vol. 2. Upper Saddle River, NJ 07458, USA: Prentice-Hall, second ed., 1998.

[12] P. V. Mockapetris, "RFC 1034: Domain names — concepts and facilities," Nov. 1987.

[13] CERT Advisory CA-98.05, "Multiple Vulnerabilities in BIND." `http://www.cert.org/advisories/CA-98.05.bind\_problems.html`, April 1998.

[14] V. Gilgor, "A Note on the Denial-of-Service Problem," in *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 139–149, May 1983.

[15] C.-F. Yu and V. Gilgor, "A Formal Specification and Verification Method for the Prevention of Denial of Service," in *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 187–202, May 1988.

[16] J. K. Millen, "A Resource Allocation Model for Denial of Service," in *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 137–147, May 1992.

[17] W. R. Stevens, *TCP/IP Illustrated — The Protocols*. Reading, MA, USA: Addison-Wesley, 1994.

[18] D. Towsley, "Providing Quality of Service Packet Switched Networks," *Lecture Notes in Computer Science*, vol. 729, 1993.

[19] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," Internet Request for Comment RFC 2401, nov 1998.

[20] F. Cohen, "A Note on Distributed Coordinated Attacks," *Computers and Security*, vol. 15, pp. 103–121, 1996.

[21] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM Computer Communications Review (CCR)*, vol. 31, July 2001.

[22] D. Denning, "Hacktivism: An Emerging Threat to Diplomacy," December 1999.

[23] CERT Advisory CA-99-17, "Denial-of-service tools." `http://www.cert.org/advisories/CA-1999-17.html`, December 1999.

[24] CERT Advisory CA-98.01, "Smurf IP Denial-of-Service Attacks." `http://www.cert.org/advisories/CA-98.01.smurf.html`, January 1998.

[25] AUSCERT, "Denial of service (dos) attacks using the domain name system (dns)." `http://www.ciac.org/ciac/bulletins/j-063.shtml`, August 1999.

[26] SectorX, "HTTP Proxies Denial of Service." `http://packetstorm.securify.com/0006-exploits/proxy.dos`.

[27] Chicken Man, "Bordermanger slow denial of service." `http://packetstorm.securify.com/0002-exploits/bordermanager-dos.txt`.

[28] R. W. Temmingh, "Decoy Blues." `http://packetstorm.securify.com/DoS/decoyblues.pl`, October 2000.

[29] T. Yardley, "Explanation and code for stream.c issues."

http://packetstorm.securify.com/DoS/stream-dos.txt, January 2000.

[30] Anonymous, on BUGTRAQ mailing list, "Various *lame* DoS attacks." http://packetstorm.securify.com/new-exploits/nukenabber-DoS.txt, November 1998.

[31] xt, "Routedsex." http://packetstorm.securify.com/DoS/routedsex.c.

[32] CERT Advisory CA-97.28, "IP Denial-of-Service Attacks." http://www.cert.org/advisories/CA-97.28.Teardrop_Land.html, December 1997.

[33] CERT Advisory CA-98.13, "Vulnerability in Certain TCP/IP Implementations." http://www.cert.org/advisories/CA-98-13-tcp-denial-of-service.html, December 1998.

[34] oasis, "oasis2.c." http://packetstorm.securify.com/0006-exploits/oasis2.c.

[35] FreeBSD, Inc, "Freebsd tcp rst denial of service vulnerability." http://www.ciac.org/ciac/bulletins/j-008.shtml, October 1998.

[36] L. Joncheray, "Simple Active Attack Against TCP," in *Proceedings of the Fifth USENIX UNIX Security Symposium*, (Salt Lake City, Utah), June 1995.

[37] S. M. Bellovin, "Security problems in the TCP/IP protocol suite," *CCR*, vol. 19, pp. 32–48, Apr. 1989.

[38] Secure Networks Inc., "Ascend routing hardware vulnerabilities." http://www.ciac.org/ciac/bulletins/i-038.shtml, August 1998.

[39] Cisco, "Cisco ios remote router crash." http://packetstorm.securify.com/advisories/cert/bulletins/VB-98.08.Cisc%o_router_crash, 1998.

[40] M. Frantzen, "Gauntlet firewall exploit code." http://packetstorm.securify.com/9907-exploits/Gauntlet_Firewall_Locku%p.txt.

[41] K. Knox, "Ip (routing information protocol) version 1 spoofer." http://rootshell.com/archive-j457nxiqi3gq59dv/199711/rip.c.html, 1996.