

Lightweight Detection of DoS Attacks

Sirikarn Pukkawanna*, Vasaka Visoottiviset†, Panita Pongpaibool†

*Department of Computer Science, Mahidol University, Rama 6 Rd., Bangkok 10400, THAILAND

E-mail: g4836585@student.mahidol.ac.th, ccvvs@mahidol.ac.th

†National Electronics and Computer Technology Center (NECTEC)

112 Phahol Yothin Rd., Klong Luang, Pathumthani 12120, THAILAND

Email: panita@nectec.or.th

Abstract-Denial of Service (DoS) attacks have continued to evolve and impact availability of the Internet infrastructure. Many researchers in the field of network security and system survivability have been developing mechanisms to detect DoS attacks. By doing so they hope to maximize accurate detections (true-positive) and minimize non-justified detections (false-positive). This research proposes a lightweight method to identify DoS attacks by analyzing host behaviors. Our method is based on the concept of BLINd Classification or BLINC: no access to packet payload, no knowledge of port numbers, and no additional information other than what current flow collectors provide. Rather than using pre-defined signatures or rules as in typical Intrusion Detection Systems, BLINC maps flows into graphlets of each attack pattern. In this work we create three types of graphlets for the following DoS attack patterns: SYN flood, ICMP flood, and host scan. Results show that our method can identify all occurrences and all hosts associated with attack activities, with a low percentage of false positive.

Keywords

Network security, intrusion detection, denial of service, traffic classification.

I. INTRODUCTION

Denial of Service (DoS) attacks pose a serious threat to the Internet. The main aim of DoS attacks is to disrupt service and network availability by attempting to reduce a legitimate user's bandwidth, or preventing access to service or system. This kind of attacks aims at rendering a network incapable of providing normal service by targeting either the network's bandwidth or its connectivity. These attacks achieve their goal by sending a stream of packets to overload a victim's network or its processing capabilities. Well-known examples of DoS attacks are flooding of TCP SYN packets and ICMP packets. Before launching attacks, attackers use tools such as port scan and host scan to discover services they can break into.

To secure networks against DoS attacks, tools such as Intrusion Detection System (IDS) must be deployed. IDS can detect DoS attacks either by matching traffic to signatures of well-known attacks (signature-based IDS), or by recognizing deviations from normal system behaviors (anomaly-based IDS). The drawback of the signature-based IDS is that it cannot detect new attacks. While the anomaly-based IDS can catch new attack patterns, its accuracy is a concern. It may flag a new non-attack activity as intrusion, resulting in a false positive. In general, IDS is notorious for its enormous resource consumption because it requires deep packet inspection and flow state maintenance.

In this paper, we propose a lightweight technique to identify DoS attacks without relying on payload inspection, or statistical behavior of overall traffic. Our technique is based on the concept of Blind classification or BLINC [1]. We evaluate our technique on four types of DoS attacks: SYN flood, ICMP flood, port scan, and host scan. Preliminary results show that our method can identify all occurrences and all hosts associated with attack activities, with a low percentage of false positives.

Following this introduction, the paper is organized as follows. Section 2 outlines previous studies in the area of DoS attack detection, namely Intrusion Detection System (IDS) and BLINd classification (BLINC) technique. Section 3 describes in detail our propose DoS detection technique and attack graphlets. Section 4 describes our experiments and results. Section 5 discusses limitations of our proposed method. Finally, conclusion and future work are discussed in Section 6.

II. RELATED WORK

A. Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) [16] is designed to analyze computer and network activities to detect intrusion attempts as they occur. IDSs can detect DoS attacks either by using traffic signatures or by recognizing anomalies in system behaviors. A signature-based IDS uses the signatures of the well-known attacks to determine if the packet represents a suspicious activity. Examples of type of signatures are port numbers and specific strings in packet payload. This concept is similar to anti-virus software on a PC that scans files and memory for known patterns of computer viruses. Anomaly-based IDS will detect abnormal behaviors by monitoring network traffic and comparing it with the baseline behaviors. The baseline will identify what is "normal" for that network. The baseline activity could be identified by a combination of average packet size, number of packets per second, flows per second, and bytes per second. Then the system can trigger an alert when it finds a significantly deviation from the baseline. A hybrid IDS that uses both technologies is also possible, where both signatures and baseline behaviors are used together either in series or in parallel.

A signature-based IDS and an anomaly-based IDS have following tradeoffs. A signature-based model is common in commercial IDSs. A signature-based IDS uses known signatures, so it may not be able to catch new attacks. However, the accuracy is high and the false positive rate is relatively low. On the other hand, an anomaly-based IDS can

detect unknown attacks, but it may result in a high false positive rate. That is, it may flag a normal activity as an intrusion. It remains a challenge for current intrusion detection technology to achieve high accuracy and low false alarms [8].

B. BLINC

The early detection of applications associated with TCP flows is an essential step for network security and network management. Port-based classification has been used extensively, but it is ineffective for applications whose ports change dynamically. A new trend of traffic classification is based on summarized flow information, such as flow duration, number of packets and packet inter-arrival time [4], [5], [6], [7]. BLINd Classification or BLINC [1] introduces a new approach for traffic classification without knowledge of port numbers, user payload, or summarized flow information. BLINC represents these patterns using graphlets. Graphlets are created by observing behaviors of hosts in three levels—the social level, the functional level, and the application level. At the social level, BLINC focuses on the popularity of a host, namely the number of distinct hosts a node communicates with. For example, p2p applications often interact with a large number of other hosts in a short time period. At the functional level, BLINC identifies the role of a host by observing the number of source ports a particular host uses for communication. For example, if a host uses a single source port in majority of its flows, BLINC assumes that this host provides a specific service (e.g., web server). At the application level, BLINC combines knowledge from two previous levels with transport layer interactions between hosts to identify the application of origin. For each application, BLINC creates behavior pattern in a form of graphlets. BLINC classification is the process of matching flow behaviors to a set of pre-defined graphlets. Moreover BLINC uses heuristics to refine final classification and to discriminate complex or similar graphlets. For instance, gaming, malware, and SpamAssassin flows, which have similar flow behaviors, are characterized by a series of packets of constant size.

The uniqueness of BLINC is that instead of studying TCP or UDP flows individually, BLINC looks at all flows generated by specific hosts. A key advantage of BLINC is that it can identify unknown applications, such as a new P2P protocol and malicious flows, which emerge as deviations from the expected behaviors. Note that these cases cannot be identified by payload or port-based analysis.

III. METHODOLOGY

This section describes our technique. Our proposed method for lightweight DoS detection is based on the idea of BLINd classification or BLINC [1]. In this work we focus on classifying four types of DoS attacks, namely SYN flood, ICMP flood, port scan, and host scan. We define three additional graphlets for SYN flood, ICMP flood, and host scan, and use the port scan graphlet defined previously in [1] (Figure 1). We detect each type of attacks by comparing flow behaviors against the graphlets as follows.

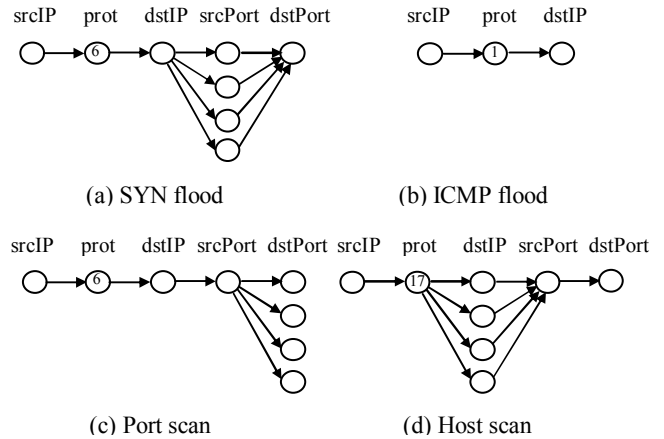


Figure 1. DoS attack graphlets

First, SYN flood exploits vulnerability of the TCP three-way handshake [2], [15]. During SYN flood, an attacker sends a lot of TCP SYN packets with a source IP address that does not exist or is not in use. The attacker also uses many random source ports to connect to a single destination port of a victim. Since the number of requests is large, the system will run out of resources and starts dropping normal connection requests. This results in a graphlet with multiple source ports shown in Figure 1(a). Secondly, ICMP flood attempts to crash operating system of a target host by sending many ICMP echo request packets. ICMP flood can be identified by the large volume of ICMP packets destined to the same destination IP address. Figure 1(b) shows the graphlet of ICMP flood.

Port scan and host scan are the tools attackers use to discover services they can break into. Although they may not necessarily cause service disruption, they provide good indicators of an onset of attacks. The objective of port scan is to find out vulnerable ports of a target host. In general, port scanning attackers use a single source port to connect to a range of ports at a single destination IP address. Note that the reverse behavior of port scan looks like SYN flood behavior and vice versa as shown in Figure 1(c). Finally, host scan checks a range of IP addresses for a certain service port. The resulting graphlet shows communication between a host and multiple destination IP addresses at a single source port and a single destination as shown in Figure 1(d).

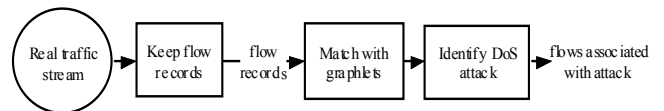


Figure 2. Flowchart for DoS detection

To identify all occurrences and hosts associated with DoS activities, we follow steps in the flowchart in Figure 2. Our detection has following three steps. The *Keep flow records* module captures real network traffic based on 5-tuple flow records (srcIP, protocol, dstIP, srcPort, and dstPort) and sends flow records to the *Match with graphlets* module, which maps each flow record to pre-defined graphlets shown in Figure 1. Finally, the *Identify DoS attack* module uses pre-defined threshold value to identify flows associated with

DoS activities. Flows that match with one of the graphlets are then classified as DoS traffic. Note that the graphlet matching is performed every fixed interval. The graphlets that have been classified in each interval will be removed from memory. The flows classified as DoS attack will be kept for future reference. Any unclassified graphlets will be carried over to the next analysis interval.

The key advantage of the proposed method is its lightweight. It can identify a group of hosts associated with DoS activities without analyzing packet content, packet size, or packet inter-arrival time. Furthermore, our technique can detect other network anomaly if they pose similar behaviors as these DoS attacks.

IV. EXPERIMENTAL EVALUATIONS

In this Section, we evaluate our lightweight DoS detection scheme through experiments with real attack traffic. Effectiveness of our scheme is measured in terms of detection accuracy and percentage of false positives. Accuracy is defined as a percentage of attack flows correctly classified as DoS attack over the total number of attack flows. Similarly, false positive is a percentage of non-attack flows misclassified as DoS attack over the total number of non-attack flows.

The effectiveness of our method depends on two parameters. One is the analysis interval—time for each round of graphlet matching. The other is a threshold value for graphlet matching. For example, to detect SYN flood, threshold is the number of source ports used by an attacking host. For ICMP flood, threshold is the number of ICMP packets. For port scan and host scan, thresholds are the number of destination ports per source IP and the number of destination IP addresses respectively.

In our experiments, we use software tools *Neptune* [9], *nmap* [10], *jping* [11], and *nbtsan*, [12] to creating DoS attacks. *Neptune* is a SYN flood attack tool from “Project Neptune”. It will attack a victim host by continuously sending TCP SYN packets at a rate of 248 SYN packets per second on average with a spoofed source IP address. *Jping* is a DoS tool that will crash a remote host by flooding a large number of ICMP packets at a rate of 472,297 ICMP packets per second on average. *Nmap* is one of the most powerful information-gathering tools available. There are a variety of scanning modes available, such as port scanning and TCP fingerprinting. In our experiments, we use *nmap* to run port scan attacks. *Nmap* will send TCP SYN packets to a range of destination port of a target host and wait for each response. If a SYN-ACK packet is received, it indicates that the port is listening. On the other hand, if a RST is received, it is indicative of a non-listener, i.e., closed port. This technique is often referred to as “half-open” scanning, because it doesn’t open a full TCP connection. The average number of TCP SYN packets generated by *nmap* is 7,930 packets per second on average. Finally, we use *nbtsan* to generate host scan attacks. *Nbtsan* is a program for scanning IP networks for NetBIOS name information. It sends NetBIOS status query to each address in a supplied range. For each responded host, it keeps record of IP address, computer name,

logged-in user name, and MAC address. The average scanning rate is 91 UDP packets per second.

Next we discuss five sets of experiments to test the effectiveness of our method.

A. Experiment I: Pure Single Attack

In this experiment, we generate DoS attack traffic between two computers with the set up shown in Figure 3. This experiment contains four sub-experiments. Traffic data of each sub-experiment consists of one pure type of DoS attack, namely TCP SYN flood, ICMP flood, port scan, and host scan. The attack traffic is generated for the duration of 10 minutes. We repeat each sub-experiment five times. We evaluate accuracy of our method at the 2-minute analysis interval. The thresholds for SYN flood, ICMP flood, port scan, and host scan are 10,000 source ports, 250,000 ICMP packets, 8,000 destination ports, and 35 destination hosts respectively. The accuracy and false positive rate of the four sub-experiments are shown in Table I.

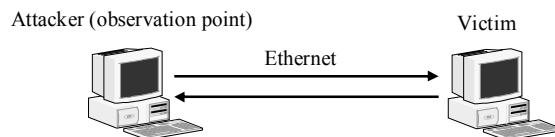


Figure 3. Pure DoS attack testbed

TABLE I
ACCURACY OF PURE, SINGLE DO S ATTACK DETECTION

Exp	Attack type	Accuracy	False positive
I-1	SYN flood	100%	0%
I-2	ICMP flood	100%	0%
I-3	Port scan	100%	8.17%
I-4	Host scan	100%	0%

From Table I, we found that our method can detect DoS attacks with 100% accuracy. There is no false positive in all except the port scan experiments. A closer look reveals that the false positive in port scan is a result of misclassifying a set of reverse port scan flows as SYN flood. The reverse traffic of port scan is the SYN-ACK or RST packets from a victim sending in response to the SYN packets of the attacker, which inevitably match the SYN flood graphlet.

B. Experiment II: Single Attack + Background Traffic

In this experiment, we add real background traffic on top of single DoS attack traffic in experiment I. The setup is shown in Figure 4. The background traffic is collected from a research office at National Electronics and Computer Technology Center (NECTEC). Users in this office are 35 undergraduate students. All hosts are on the same broadcast LAN 100 Mbps. The captured interval time is two minutes. We apply the same thresholds from previous experiment I. Table II shows characteristics of captured traffic and accuracy of each sub-experiment, averaged after five repetitions.

From Table II, we found zero false positive in all but the port scan experiments, similar to the result of experiment I. It turns out that if traffic data contains real background traffic,

the false positive in case of port scan is less than that of pure attack traffic. This is because the background traffic interrupts the reverse flows of port scan, causing slower arrival of response packets.

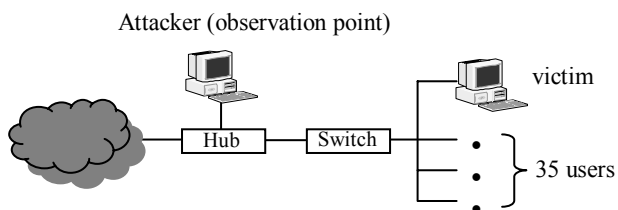


Figure 4. DoS attack testbed with background traffic

TABLE II
ACCURACY OF DoS ATTACK DETECTION WITH BACKGROUND TRAFFIC

Exp	Attack type	Total Flows	Byte (MB)	Accuracy	False positive
II-1	SYN flood	24,285	23.3	100%	0%
II-2	ICMP flood	203,989	130	100%	0%
II-3	Port scan	33,561	27.3	100%	5.97%
II-4	Host scan	3,314	2.1	100%	0%

C. Experiment III: Threshold Evaluation

In order to cope with false alarms, we experiment with different threshold values for each attack graphlet. We find appropriate thresholds for detecting each attack, assuming a two-minute analysis interval. For each graph, we generate five sets of single attack traffic on real background traffic collected at the same location as experiment II, and compute average accuracy at different thresholds. Figure 5 shows effects of different threshold values to the false positive of detection.

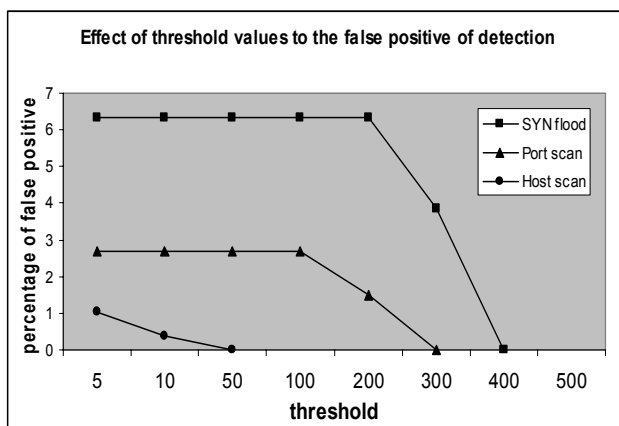


Figure 5. Effect of threshold to the false positive of detection

We found that thresholds of 400 source ports, 300 destination ports, and 50 destination hosts are sufficient to detect SYN flood, port scan, and host scan generated by *Neptune*, *nmap*, and *nbtscan* respectively. ICMP flood shows zero false positive for all thresholds because there is no

ICMP packet in background traffic. To differentiate ICMP flood from real ICMP traffic, we use threshold of 10,000 ICMP packets.

D. Experiment IV: Multiple Attacks + Background Traffic

Next we evaluate accuracy of our method, assuming two-minute analysis interval and graphlet thresholds from experiment III. We carry out two sub-experiments. The first sub-experiment consists of one instance each of SYN flood, ICMP flood, port scan, and host scan. The second one contains two instances per each of the four attacks. Both experiments were performed five times on real background traffic from the same location as experiment II. Table III lists characteristics of captured traffic and accuracy of the two sub-experiments.

TABLE III
CHARACTERISTICS OF COLLECTED TRAFFIC DATA

Exp	Captured time on 2007-05-02	Total Flows	Bytes (MB)	Accuracy	False Positive
IV-1	10.40-10.50	255,133	193.7	100%	13.40%
IV-2	11.25-11.35	301,671	203.6	100%	17.47%

The results in Table III have proven that our method can detect all occurrences of DoS attacks (i.e., 100% accuracy). However, it yields higher false positive rate than the case of single attack in experiments I and II. This is mainly due to misclassifying a set of reverse SYN flood flows as port scan and vice versa, as discussed previously.

E. Experiment V: Effect of Analysis Interval

Another parameter of interest is the analysis interval. While we recognize that two-minute interval provides sufficiently accurate results, we want to know how near real-time our method can detect DoS attacks. We measure detection accuracy as a function of analysis interval, shown in Table IV. The traffic used in the experiments is the same set as that in experiment IV-1.

TABLE IV
EFFECT OF ANALYSIS INTERVAL

Analysis Interval	30 sec	1 min	1.5 min	2 min
Accuracy	98.44%	100%	100%	100%
False positive	15.07%	13.40%	13.40%	13.40%

From our experiments, we found that if we gradually reduce analysis interval from 2 minutes to 1.5 and 1 minute, the accuracy and false positive of detection do no change. However, when the analysis interval is reduced to 30 seconds, the accuracy of detection decreases from 100% to 98.44% and the false positive of detection increases from 13.40% to 15.07%. Therefore, we can conclude that decreasing the analysis interval from two minutes to one minute has no impact on the overall performance of our method. Moreover, one minute is the earliest we can detect attack traffic without compromising detection accuracy.

V. DISCUSSION

In this Section, we would like to highlight some issues and limitations of our proposed method.

False alarms: In our method, there are several events that may cause false alarms of detection. Example scenarios are as follows:

- In some cases, one attack could fit more than one DoS attack graphlets. For example, a port scan activity could be marked as port scan and SYN flood. Since the reverse behavior of port scan looks like SYN flood behavior and vice versa, our system classifies a set of reverse port scan flows as SYN flood. To avoid such mismatch, we could consider the TCP flag in addition to the 5-tuple flow record. However, this would make our detection method heavy-weight. Instead, we choose to ignore the TCP states and flags, and only match flow records with attack graphlets. Therefore, the false positives between port scan and SYN flood are inevitable.
- Some applications, such as download manager, often open a large number of connections in short time. An example of this application is Flashget [17]. Flashget will create a large number of connections from multiple source ports to a single destination port of a server for downloading a file. This behavior is similar to that of SYN flood. However, the difference between SYN flood and this kind of application is that the SYN flood will not complete a TCP 3-way handshake with the target victim. Therefore, we may need to apply a technique described in [18] in order to distinguish them.
- Many P2P file sharing applications are more inclined to use a single source port to connect to a lot of destination IP addresses for sharing files [19]. This behavior looks like a host scan activity. A carefully selected threshold of number of destination IP addresses will help differentiate the two activities.

Configuring threshold: As shown in experiment III, accuracy of detection depends highly on setting appropriate thresholds for each graphlet. However, a threshold such as number of flows may depend on many environmental factors, for example, available bandwidth of current network, characteristic of DoS attack tools, number of generated attacks, duration of attack, operating system, and computer architecture of the attack host. Therefore, the threshold values presented in this paper may be specific for our experimental setup only. Network administrators may need to adjust their thresholds according to their network environment to achieve high accuracy and low false alarms.

Spoofed IP address: our proposed method cannot identify real attacker if source IP addresses are spoofed. Our system will recognize spoofed source IP address as attacker.

Encrypted packet header: our entire approach is based on relationships among the fields of the packet header. Consequently, our technique has the ability to characterize encrypted traffic as long as the encryption is limited to the

packet payload. Should layer-3 and layer-4 packet headers be encrypted, our methodology cannot analyze.

-Pinpointing original attackers: A DoS attack may be a part of distributed DoS (DDoS) attack activities. Under DDoS attacks, an attacker may not directly attack the victim, but exploit multiple agents to generate attack on its behalf. In this case, our method cannot identify the original attacker of such DDoS attack activity.

VI. CONCLUSION AND FUTURE WORK

We propose a lightweight method to identify DoS attacks and their onsets. Our method can identify SYN flood, ICMP flood, port scan, and host scan, based on the idea of BLINC's host behavior analysis. The procedure has two steps. First we create attack graphlets by examining unique flow behaviors. Secondly, we identify an attack flow by matching flow records to the pre-defined graphlets. The advantage of our method is that it can identify all occurrences and all hosts associated with attack activities without relying on packet payload, packet inter-arrival time, or size of individual packets. Moreover, it can effectively detect anomalous behaviors in the network if the flow behaviors are similar to DoS attacks. In addition, our method can perform near real-time detection, within one minute interval, with low false alarms.

We are in the process of developing graphlets for other types of DoS and DDoS attacks, such as Smurf, Trinoo, TFN/TFN2K, and Stacheldraht [2], [13], [14]. We also plan to compare our performance with a de-facto IDS, such as Snort [3]. In addition, we plan to improve our method to be more real-time and to distinguish between DoS and applications with similar traffic behaviors, such as P2P file sharing.

ACKNOWLEDGMENT

We would like to thank the National Electronics and Computer Technology Center (NECTEC) in Thailand for allowing us to collect traffic data.

REFERENCES

- [1] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *ACM Sigcomm*, 2005.
- [2] J. Mirkovic, J. Martin, and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," *ACM Sigcomm Computer Comm. Rev.*, vol. 34, no.2, 2004, 39–53.
- [3] Snort, <http://www.snort.org>.
- [4] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," *Internet Measurement Conference*, 2004.
- [5] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," In *Proceeding of Passive and Active Measurement Workshop*, 2004.
- [6] D. Zuev and A. Moore, "Traffic classification using a statistical approach," In *Proceeding of Passive and Active Measurement Workshop*, 2005.

- [7] A. Moore and D. Zuew, "Internet traffic classification using Bayesian analysis," In *Proceeding of ACM SIGMETRICS*, 2005.
- [8] Yan Qial and Xie Weixin, "A Network IDS with Low False Positive Rate," In *Proceeding of the 2002 Congress on*, Vol.2, pp. 1121-1126, 2002.
- [9] Neptune <http://www.phrack.org/archives/48/P48-13>.
- [10] nmap <http://insecure.org/nmap/>.
- [11] jping <http://www.tenebril.com/src/info.php?id=11777455>.
- [12] nbtscan <http://www.inetcat.net/software/nbtscan.html>.
- [13] CERT Coordination Center, "Denial of Service Attacks," http://www.cert.org/tech_tips/denial_of_service.html.
- [14] CERT Coordination Center, "Trends in Denial of Service Attack Technology," October 2001, http://www.cert.org/archive/pdf/DoS_trends.pdf.
- [15] CERT Coordination Center, "TCP SYN flooding and IP spoofing attacks," <http://www.cert.org/advisories/CA-1996-21.html>.
- [16] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Technical Report 99-15, Department of Computer Engineering, Chalmers University, March 2000.
- [17] Flashget <http://www.flashget.com/>.
- [18] Haining Wang, Danlu Zhang, and Kang G. Shin, "Detecting SYN Flooding Attacks," In *Proceeding of IEEE INFOCOM'2002*, New York City, June 2002.
- [19] Security Focus, "Identifying P2P users using traffic analysis", <http://www.securityfocus.com/infocus/1843>.