

Detecting Flood-based Denial-of-Service Attacks with SNMP/RMON*

William W. Streilein, David J. Fried, Robert K. Cunningham
MIT Lincoln Laboratory

Abstract

We present our work in detecting DoS attacks through the polling of Remote Monitoring (RMON) capable devices. Rather than the introduction of special purpose hardware, our detection capability relies upon RMON capabilities present in existing infrastructure network devices, such as switches and routers. RMON is a special purpose Management Information Base (MIB) designed for the SNMP (Simple Network Management Protocol), which tracks low-level network usage indicators, such as byte and packet count, packet size, and packet transmission error events. Using RMON data polled from a live enterprise network, we have developed a detection algorithm for simulated flood-based DoS attacks that achieves a high detection rate and low false alarm rate. The detection algorithm relies not only on the raw RMON variables but also on relationships between the variables to achieve this detection rate. We also indicate how the introduction of RMON2 variables and an accurate network map can be used to improve DoS detection accuracy and reduce false alarms by identifying the sources of specific DoS-related traffic. Our approach is less expensive than many commercially available solutions, requiring no special purpose hardware. It is more accurate than commonly used univariate statistical approaches and it is fast, requiring only the computation of packet variables ratios and processing by a feed-forward neural network.

1. Introduction

Denial of service (DoS) attacks continue to threaten networks and computers connected to the Internet. As reported in the CSI/FBI Computer Crime and Security Survey of 2003[1], 42% of respondents indicated DoS attacks were a major problem for them. Financial setbacks due to DoS attacks alone were the 2nd biggest cause of revenue loss, just after proprietary information theft.

CERT [2] defines the DoS attack as “the prevention of authorized access to a system resource or the delaying of system operations and functions”. While other taxonomies for DoS attacks exist [3], DoS attacks are typically separated into two broad classes: “logic-based” attacks, which exploit vulnerabilities in a system’s software to render it unable to respond to valid user requests; and “flow-based” attacks, which exhaust available resources, such as CPU cycles, system storage or network bandwidth, again rendering the target machine unresponsive to requests. In the present study, it is appropriate to distinguish further within the class of flow-based DoS attacks between those that consume target resources and those that consume bandwidth. Resource

* This work is sponsored by the U.S. Air Force under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

consumption attacks, such as the TCP SYN attack, exhaust available memory on the target machine by overwhelming it with connection requests [4]. Bandwidth consumption attacks [5], such as the well-publicized attacks of early 2000, in which traffic at a rate of 3.5 Gigabits per hour flooded the target networks of major e-commerce sites and denied legitimate users' access, consume available network bandwidth. Our experiments focus on the modeling and simulation of flow-based, bandwidth-exhaustion DoS attacks.

While damaging single-source flow-based DoS attacks are still seen in the Internet, the trend today is toward the use of distributed DoS attacks or DDoS attacks. A DDoS attack is one in which multiple traffic generating sources take part in attacking a target machine. These attacks typically involve unwitting compromised machines or "zombies" that cooperate to inundate a target machine with traffic[6, 7].

The impact of DoS attacks has increased steadily in recent years [9] and, as more computers connect to the Internet, an ever-increasing supply of zombies and target machines [3] become available to the attacker wishing to launch a DoS attack. A recent survey estimated that nearly 4000 DoS attacks per week are active on the Internet at any given time [8]. Part of the reason for the increased threat due to the DoS and DDoS attack is that attack tools have advanced significantly in recent years, showing evidence of increased sophistication in automation, amplification, and stealth [10]. Attack tools are sufficiently easy to use, even would-be hackers with limited coding experience can launch worms and scripts that cause damage en par with the expert attackers [7, 11].

We use Remote Monitoring (RMON) data to model of network traffic behavior and to detect flow-based, bandwidth exhaustion DoS attacks. RMON [23] is a Management Information Base (MIB) specification for use with the Simple Network Management Protocol (SNMP) [24] that monitors attributes of network devices. Within the family of RMON MIBs, RMON1 identifies attributes of low-level ethernet traffic that can be used to characterize network utilization through byte, packet, and error counts. The RMON1 specification is supported by most enterprise routers manufactured by CISCO and NORTEL [25] and thus removes the need for special purpose DDoS detection hardware. Moreover, using the SNMP management model requires no reconfiguration of the network being observed. An SNMP network manager periodically queries an agent for attributes of interest and SNMP traffic and only introduces a small amount of load on the network.

Others have used the RMON MIBs to detect DoS attacks although most relied on RMON2 (layers 3 and 4 of the TCP/IP stack). One group of researchers utilized RMON in order to recognize when traffic to certain destination addresses exceeded expected limits [26]. However, this study relied upon RMON2 [27] which monitors network and application layers 3 and 4 of TCP/IP traffic. Although good detection capability was demonstrated, RMON2 is not commonly supported by today's network traffic devices. Another study [16] also relied upon RMON2-like variables for DoS detection, but took a different approach by monitoring indicators of network congestion to indicate anomalous DoS traffic.

Given the widespread problem DoS attacks pose and the damage caused by them, it is not surprising to see a number of research and commercial efforts attempting to detect and respond to them. A major problem for researchers seeking to detect DoS attacks, however, is the fact that DoS attack traffic generally doesn't contain specific signatures that can be detected by typical IDSes. Detection mechanisms must, therefore, rely on other techniques, such as recognizing anomalous bandwidth usage [12-14], modeling Internet-wide traffic [15, 16], and performing spectral analysis of packet arrival times [17] to be effective.

Once a DoS has been detected, address and application filters can be applied along the DoS network attack path to block the offending traffic at its source. However, actually tracing DoS traffic back to locate the source of attack traffic is difficult when source addresses are spoofed. Tracing DoS traffic back to its source is further complicated when the traffic appears to cross Internet Service Provider (ISP) boundaries. Cooperation between ISPs is crucial to the success of many traceback efforts. DoS traceback and response remain major areas interest in research and commercial realms [18-21].

Many commercial solutions provide DoS detection and response capabilities. However, these solutions typically require the use of expensive proprietary hardware and often require redesigning the client's network to accommodate the new hardware [13]. These systems typically rely upon setting thresholds on traffic indicators, which when exceeded are taken to indicate the presence of DoS traffic [12]. As models of network behavior, threshold-based detection systems may fail to adequately represent the dynamic nature of Internet traffic [22] and thus may produce too many false alarms to be useful.

The rest of the paper proceeds as follows. Section 2 describes our system development methodology and sets the stage for the data analysis, model development, and prototype sections that follow in Section 3. The results of our detection experiments on simulated DoS traffic are presented in Section 4, while Section 5 describes our prototype testbed and associated detection results. Section 6 discusses the problem of false alarm mitigation in the context of DoS detection and offers several solutions. It is followed by a brief summary of our work in Section 7.

2. Methodology

Although it would be useful to collect data from an operational network during a DoS attack, we had to settle for collecting operational data and superimposing DoS attacks. The use of operational data enables us to build realistic models of network traffic and to understand typical network usage parameters. It enables our system to remain relevant to real-world traffic, an important consideration when developing a system intended for Internet use [12, 28]. The collected RMON1 data was assumed to contain no DoS attack traffic. Other researchers have demonstrated success utilizing similar network modeling and attack traffic simulation approaches [29, 30]. Because RMON1 variables are basically simple counts of bytes and packets that have been seen by the RMON

monitoring device during a given time period, we introduce DoS-like traffic into the data by adding DoS-like byte and packet counts to the collected RMON1 data.

To ensure that artifacts introduced by superimposing the two traffic streams do not affect performance, we tested our prototype detection system on a testbed that emulates the collected network traffic, and developed and staged two representative DoS attacks, mstream [31] and bsz [32].

3. Data Collection, Analysis and Preparation

The network from which the data was retrieved is pictured in Figure 1. RMON data was collected from the central CISCO switch every 5 minutes during a 4-month period from October, 2001 through January, 2002. Five minute sampling of RMON1 variables amounted to the extraction of 288 data points every day from the switch. The CISCO switch sports 24, 100 Mbit/s ports and represents the central conduit for traffic to and from the Internet. Our study was based upon data retrieved from port 3, the port providing external connectivity to the Internet, via a firewall. Outside of the firewall, Internet traffic is bound by a T1 link, with a maximum traffic rate of 1.54 Mbits/sec.

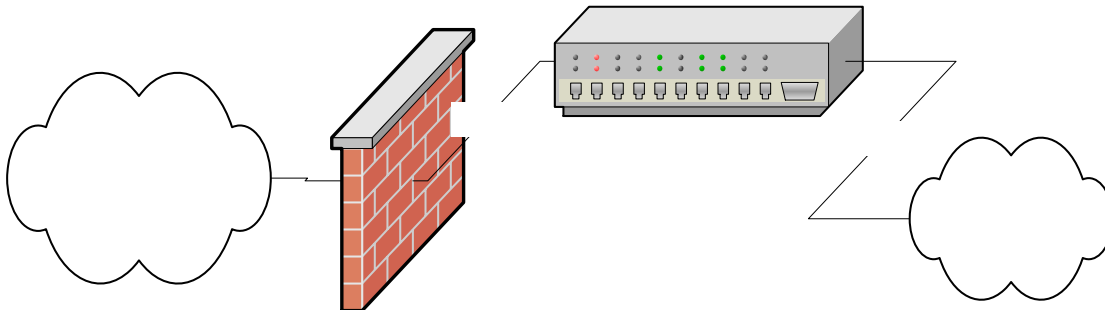


Figure 1. Network Diagram of data source. Port 3 handles all Internet traffic. Port 24 handles internal traffic to the network.

We focused our analysis only on workdays; data corresponding to weekends and holidays were removed from the data set. As shown in Figure 2, a plot of the mean bandwidth utilization as reflected in the RMON1 octet count across the day reveals the typical diurnal cycle for the workday. The workday begins around 7am when utilization increases and ends around 5 or 6 pm when utilization tapers off. Variability is small before and after hours of work, and large during the day. From this it is clear that any attempt at modeling normal traffic behavior on this network must consider time of day as a factor effecting utilization. It is also important to note that typical network utilization is less than $\frac{1}{2}$ of the external T1 line. This suggests that the introduction of ‘extra’ traffic at the $\frac{1}{2}$ T1 rate (750kbits/sec), equivalent to five mstream zombies sending 300, 60-byte packets a second, would exhaust the available bandwidth on the external link and be

DoS-like. Thus, byte and packet counts corresponding to a traffic rate of $\frac{1}{2}$ a T1 were added to the original RMON1 data for our simulations.

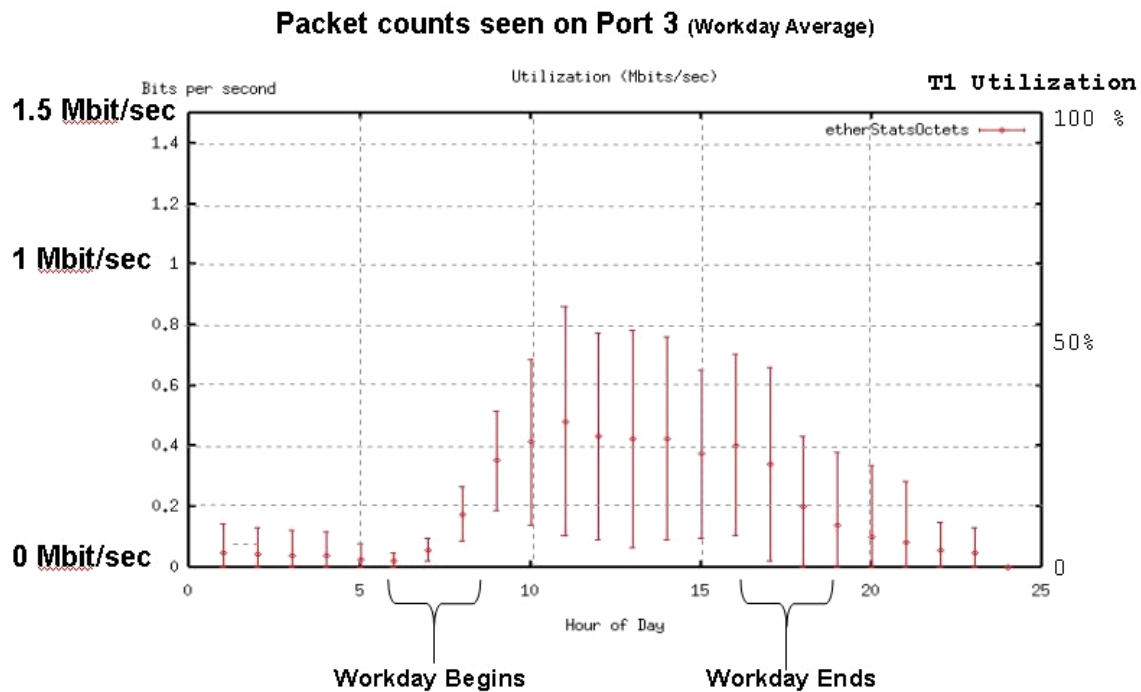


Figure 2. Network utilization as a function of time of day. Error bars indicate one standard deviation around average utilization.

RMON1 tracks the number of packets seen on the network. Packet size bins maintain counts of certain packet sizes including: less than 64 bytes, between 64 and 127 bytes, between 128 and 255 bytes, between 256 and 511 bytes, between 512 and 1023 bytes, and between 1024 and 1518 bytes. An overall packet count records all packets seen on the network. RMON1 also tracks transmission errors seen on the line due to collisions and congestion, however none of these errors was present in the collected data. By comparing each bin count to the overall packet count, it is possible to get a measure of the percentage of packets in the network that correspond to each of the six packet sizes recorded by RMON1. Figure 3 presents the ratio of packet size bins to the total packet count, as a function of the hour of the day. The ratios of packet sizes suggests that the majority of traffic on this network is web-destined traffic [29, 33]. Here, small packets representing client requests account for roughly 20% of the packets seen and large packets representing server responses with data payloads, account for 60% of the traffic. Because of their obvious relevance to the network traffic, these packet ratios form the major portion of the expanded input feature to our machine learning model for DoS detection.

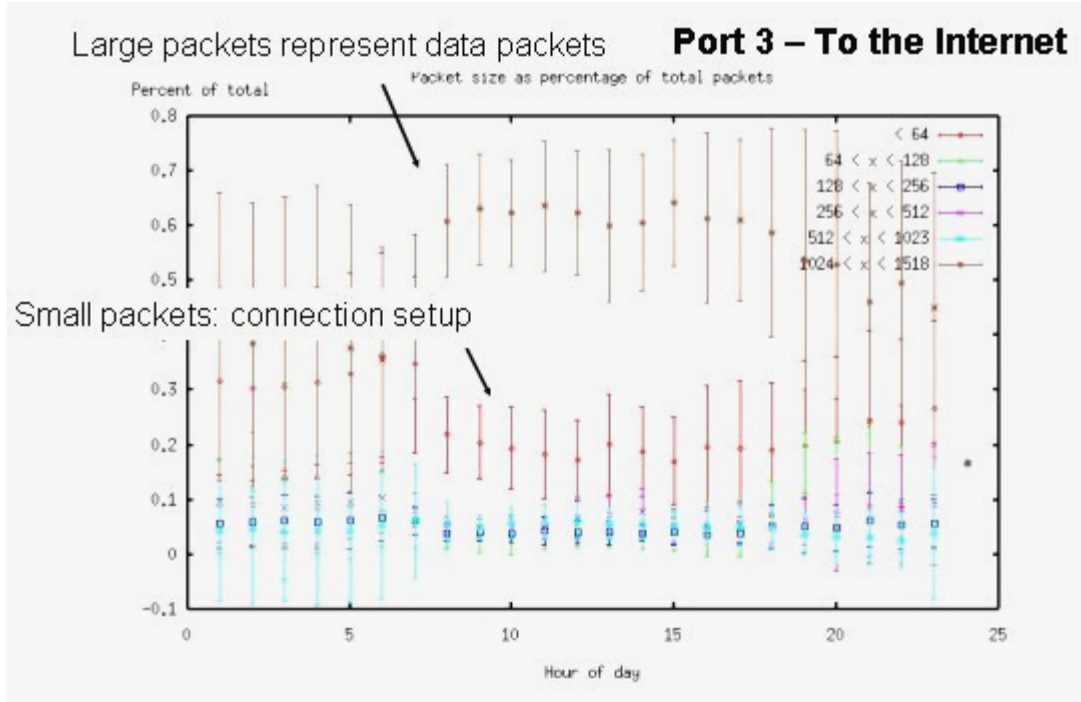


Figure 3. Packet Size Ratios. Connection setup accounts for 20% of traffic, Data packets from server account for 60% of traffic.

The results of our simulations and detection experiments are presented in the next section. For the experiments below, the data was divided equally into training and test sets; simulated DoS traffic was added to the data by adding in appropriate counts to the RMON1 octet and packet variables. Input feature vectors (described below) were generated separately for each set and presented randomly to the detection models during training.

4. Experimental Results

The following section presents our detection results for several experiments involving our DoS detection models and simulated traffic. The first experiment compares the detection capability of the statistical and the machine learning models. Experiment 2 investigates DoS traffic detection when the traffic is restricted to be in packets of specific sizes. In Experiment 3, we determine the amount of training data necessary to achieve adequate detection rates. And in experiment 4 we analyze detection results to understand when detection errors are most likely to occur during the typical day.

The detection results from our experiments are presented as Detect Error Tradeoff (DET) curves [34]. DET curves allow for the presentation of miss and false alarm probabilities on normal deviate scales (similar to log-log scale) and allow closer inspection of detection results that are very good..

4.1. Experiment 1 - Comparison of Univariate Gaussian and MLP Detection Models

In the first experiment we compared the detection capabilities of two models developed to detect simulated DoS traffic added to the collected RMON1 data. The first model assumed a simple uni-variate Gaussian distribution of network utilization. Separate Gaussian models were fitted to each of the 288¹ daily data points characterizing network utilization across the day. Estimates of network utilization for the model were obtained by consulting the `etherStatsOctets` variable contained within the RMON1 MIB. This model represents the modeling approach taken by many commercial offerings [13].

The second detection model employed a multi-layer perceptron trained to separate normal traffic from attack traffic. The MLP was constructed with 8 input nodes, 5 hidden units, and 2 output nodes. The model was presented with a feature vector that included: time of day (as a fraction of the total day), the ratios of each of the 6 packet bins counts to the total number of packets, and the overall T1 network utilization.

Although network traffic has been shown to be difficult to describe with non-power-law statistics [22], our analysis of the collected RMON data indicates that the standard statistical assumptions underlying the simple statistical model and the multilayer perceptron [35] can lead to quite good detection results.

Figure 4 presents detection results for the statistical model and the MLP detection models. As seen in the figure, the machine learning model out-performs the statistical model in both detection rate and false alarm rate. The statistical model intersects the equal-error line, where the cost of a miss is equal to the cost of a false alarm, at a detection rate of 93.1% and a false alarm rate of 6.9%. Along this same line, the machine learning model attains a detection rate of 98.6% and a false alarm rate of 1.4%. Given that the system would attempt to classify the sampled RMON1 data 288 times a day, these false alarm rates can be translated into 19.8 false alarms (FAs)/day, in the case of the statistical model, and 4.0 FAs/day in the case of the machine learning model. A low false alarm rate is important for intrusion detection systems because it reduces the amount of time a user must spend researching non-attack situations [36]. The false alarm rate of the statistical model seems too high to be useable in a real network. The false alarm rate of the machine learning model, by contrast, is better by a factor of 5.

The superior detection capability of the machine learning model is due in part to the enhanced feature vector. The presence of packet size ratios enables the MLP to better model the network traffic. Other researchers have shown that packet sizes can be a useful feature when building detectors of anomalous network traffic network traffic [29, 33]. In addition, the multilayer perceptron is a more sophisticated learning algorithm than the simple statistical model and as such is better suited to the task of approximating the posterior probabilities of normal and attack class input vectors [35].

¹ RMON1 data was sampled every 5 minutes: $60 \text{ min} / 5 \text{ min} * 24 \text{ hr} = 288$.

Port 3 – To the Internet

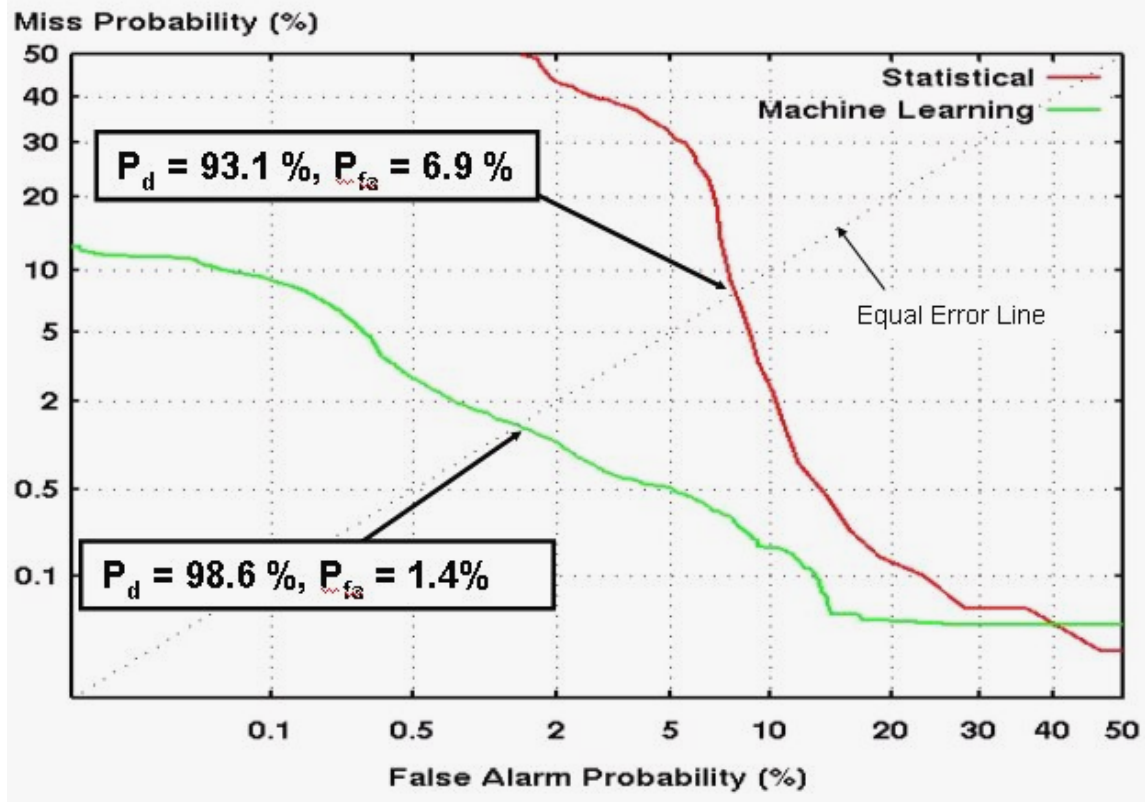


Figure 4. A comparison of detection results for the statistical and machine learning models of DoS detection. The machine learning model (MLP) outperforms the statistical model by achieving a higher detection rate and lower false alarm rate.

4.2. Experiment 2 – Detection Capability of Different Packet sizes

In Experiment 2, we investigate the notion that the improved detection capability of the machine learning model is due to the presence of packet size ratios in the input feature vector. By simulating attacks with packets of specific sizes, we can understand the extent to which packet size ratio and not network utilization alone is important for detection. For this experiment, we chose to implement attacks with two different packet sizes: 64-byte and 1270-byte packets. These packets sizes are typically used by common DoS and DDoS attacks [31, 32]. Because of the nature of RMON1 accounting, the 64-byte packet will be counted in the 64-byte packet bin while the 1270 byte packet will be counted in the 1024-1518 packet bin. A packet size of 1270 was chosen because it is mid-way between the two extremes of this largest packet bin. As shown in Figure 5, the machine learning model can detect the 64-byte attack 99.4% of the time at a false alarm rate of .6%, and can detect the 1270-byte attack 97% of the time at a 3.0% false alarm rate. It is worth noting that even though equal numbers of simulated DoS packets were added in both cases, roughly 20 times as much traffic is required in the 1270-byte case to achieve a detection rate near that of the 64-byte packet attack. The false alarm rate of 3.0% (8.6

FAs/day) during the 1270-byte attack is significantly higher than .6% (1.7 FAs/day) in the 64-byte attack.

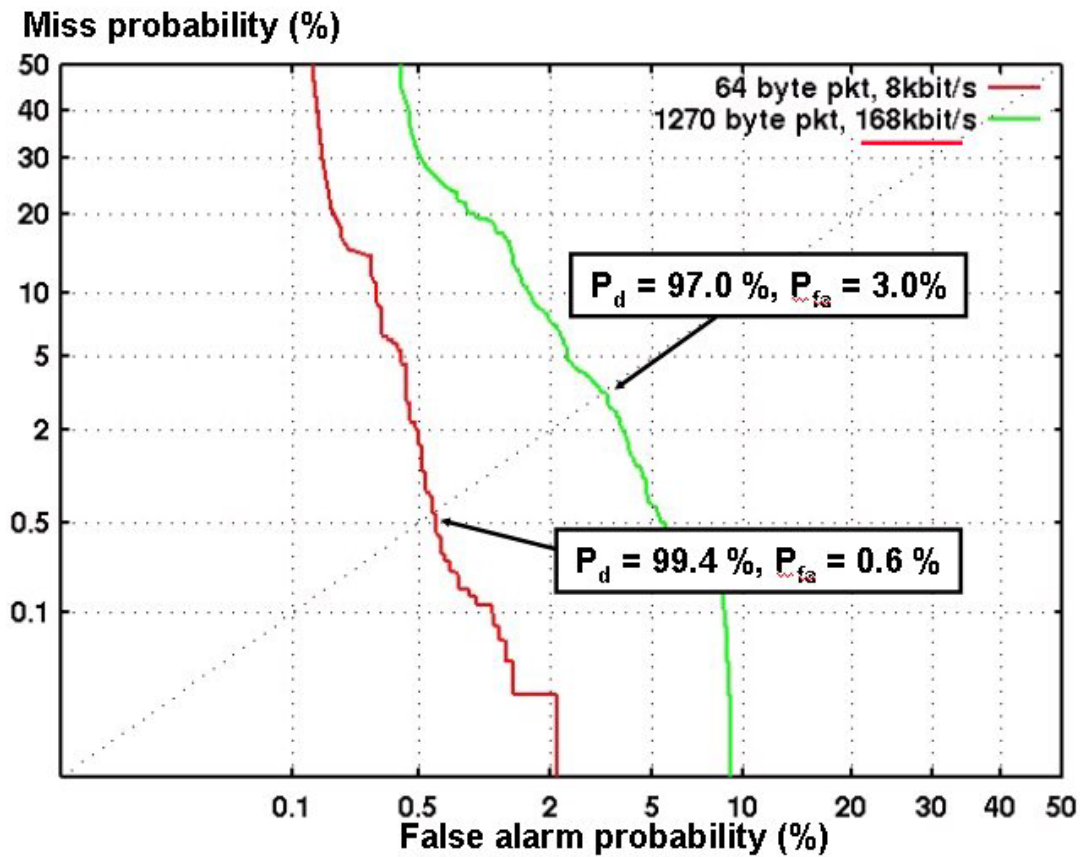


Figure 5. MLP Detection results for 64-byte DoS attack and 1270-byte DoS attack

The lower detection rate and higher false alarm rate for the 1270-byte attack can be explained by considering the data shown in Figure 3. As seen in this figure, large packets account for 60% of the traffic on the network, while small packets account for only 20% of the traffic. Given that a high ratio of large packets is typical for this network, more attack traffic is needed to achieve the same accuracy rates as with the smaller packets.

4.3. Experiment 3 - Impact of training duration on detection rate

In Experiment 3, we examine the detection capability of the machine learning model as we reduce the amount of training data supplied. Understanding the amount of training time necessary to achieve good results is important for any system that is intended to be deployed on a real network. Shown in Figure 6 are the detection results when 1 week, 4 weeks, and the full 16 weeks of training data are used to train the machine learning model. As can be seen in the figure, after 1 week of training data is used, the performance is similar to that of the statistical model's results. The detection rate is

89.5% while the false alarm rate is 10.5%. When 4 weeks are used for training, the false alarm rate is reduced by half (5% = 14 FAs/day) and the detection rate improves to 95%.

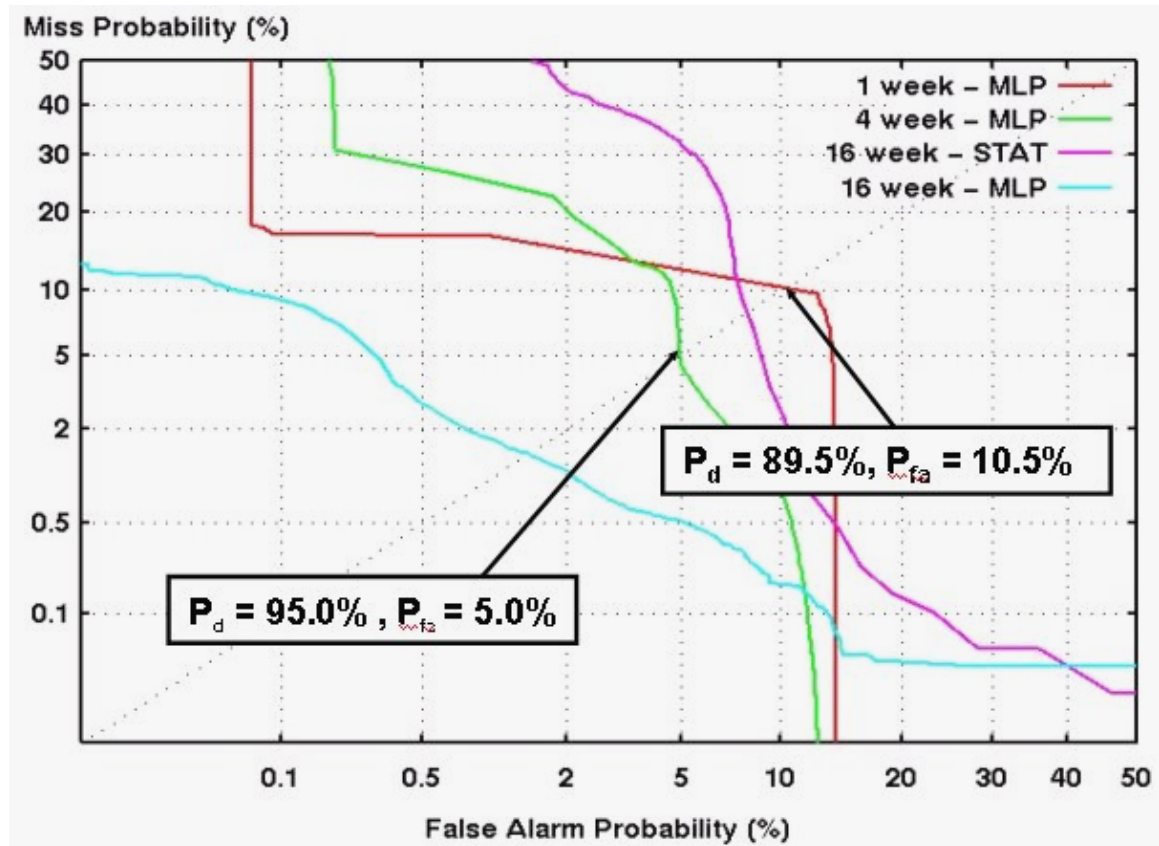


Figure 6. MLP detection results when different amounts of training data are supplied

4.4. Experiment 4 – Diurnal Accuracy Variability

When a human has to review alerts it is much better to have alerts occur during the day (when someone is working) than at night. As a final experiment, we returned to the original detection results to determine when false alarms and misses were most likely to occur during the typical day. As presented in Figure 2, traffic rate and network utilization is greater during the workday period from 7AM through 5PM, than during non-workday periods. The heaviest traffic and utilization occurs during the midday hours. During heavy traffic periods, it is more difficult to separate anomalous traffic from normal traffic and, thus more misses and false alarms occur during this period.

Taking the total number of misses and false alarms that occurred during any given hour and normalizing this count by the total number of days, we arrive at the data presented in Figure 7, which presents the expected number of occurrences on an hourly basis of false alarms and misses throughout the day. As seen in the figure, misses and false alarms occur with greater frequency during the working hours of the day. Thus, two analysts monitoring the network from 6AM to 10PM would encounter on average 96% (2.20) of the false alarms and 95% (4.67) of the missed detections occurring during the day.

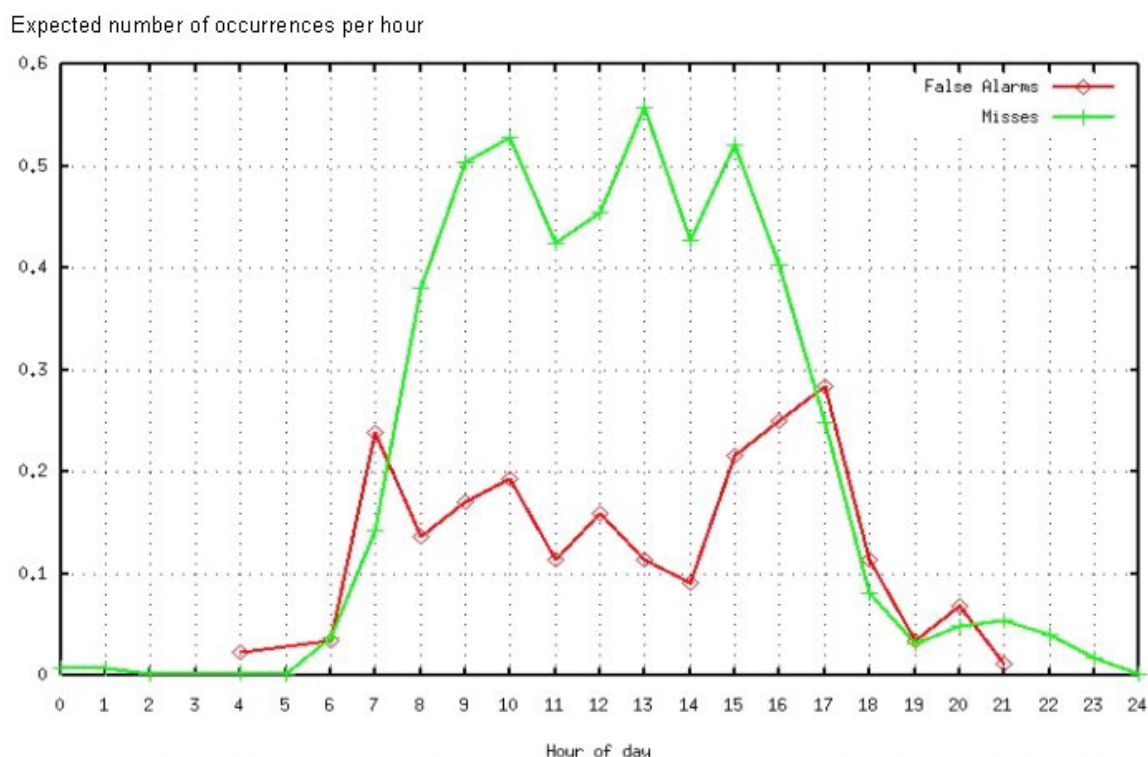


Figure 7. Expected number of occurrences of false alarms and misses per hour

5. Testbed Development and Prototype testing

To validate the machine learning model developed using superimposed DoS and network traffic, we deployed a prototype DoS detection system on a network traffic testbed. We employed the LARIAT [37] traffic generation system to replicate the normal traffic collected from the operational network. The replicated traffic consisted of a variety of TCP/IP services (FTP, HTTP, etc.) exchanged between several virtual machines and exhibited a similar diurnal cycle to that witnessed on the operational network. The testbed network enabled the staging of two DoS attacks (mstream and bsz) while RMON1 variables were monitored from both the CISCO switch and two dedicated RMON probes.

The results shown in Figure 8 demonstrate reasonable detection accuracy for the MLP-based prototype DoS detection system. At the point where the detection curve crosses the equal-error line, the system achieves a detection rate of 96.2% and a false alarm rate of 3.8% (10.9 FAs/day). Believing a portion of the false alarms were due to isolated spurious normal data points, we imposed a restriction upon the system that requires 2 and 3 consecutive data vectors be recognized as ‘attack’ before the presence of an attack is declared. This proximity-in-time of attack traffic has been used by other researchers to improve detection rates of attack traffic [38]. As shown in the figure, although the false alarm rate is reduced significantly to the rate of .2% (.5 FAs/day), the maximum detection rate goes down, as well. In fact, because of the consecutive alert restriction, the

system is unable to detect DoS traffic present at the beginning and ending edges of the test periods and, therefore, encounters a ceiling on its maximum detection rate well below 100%.

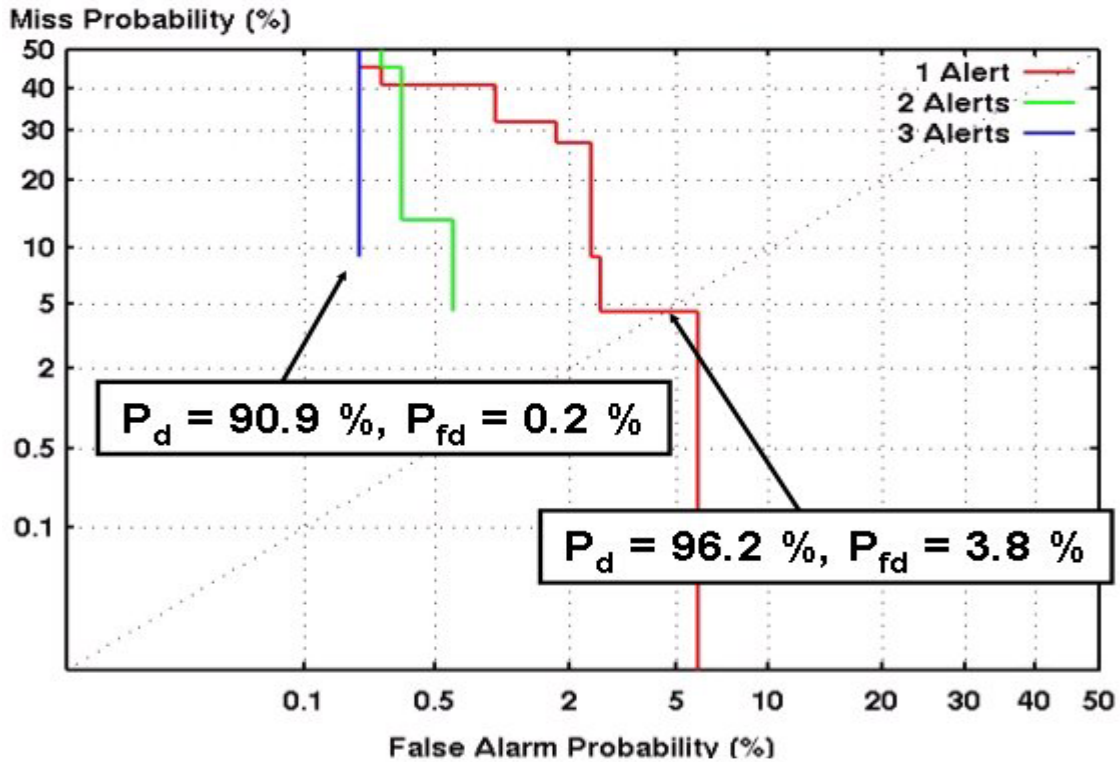


Figure 8. Detection results for prototype detection system on testbed during mstream and binary stream zero DoS attacks using the machine learning model.

6. Discussion – Other ways to reduce false alarms and RMON2

Although the detection system developed here works quite well, it is still worthwhile to consider ways to reduce the false alarm rate, as the number of false alarms per day that must be reviewed by network administrators and analysts is often an impediment to acceptance of intrusion detection systems [39]. Requiring the accumulation of evidence of an attack before declaring its existence is one method for reducing the occurrence of false alarms. However, this technique can reduce the timeliness of the detector, and reduce the effectiveness of the detector when faced with short-duration attacks. For example, if a detection system samples the network every T seconds and requires N consecutive indications before declaring the presence of an attack, such a system would not detect DoS attacks with a duration shorter than $N * T$. In our experiments, a sampling period of 5 minutes and a requirement of 3 alerts suggests that our detector would miss DoS attacks that last less than 15 minutes and would have to accumulate 15 minutes of evidence before declaring the presence of a longer duration attack. This detection delay could be mitigated by launching an automatic active test for DoS traffic immediately,

when the first indication is seen or by enhancing the detector's input feature vector to include recent DoS alert history.

Another way to reduce the number of false alarms produced by an anomaly-based intrusion detection system is to incorporate knowledge about the network being observed (i.e. a network map) into the detection process. The map could contain information about servers and clients and the services they are utilizing and could be consulted when an anomaly (a possible DoS attack) is detected to confirm that the source of the traffic is behaving unusually. Such a network map can be built by hand by the user or created automatically from network data, such as the address and application pairs witnessed in TCP/IP traffic. Although RMON1 is not able to peer into packet contents and recognize source and destination pairs, RMON2 is. RMON2 [27] tracks information at layers 3 and 4 of the TCP/IP network stack and could be utilized during training of the system to build a map of the network under observation.

7. Summary

We have demonstrated that an effective DoS detection system can be constructed that relies upon RMON1 data collected from existing network devices. Based upon 4 months of RMON1 data extracted daily from a live network switch in a production network, we developed two DoS detection models that were able to distinguish normal traffic from attack traffic. The machine learning model achieved a higher detection rate (98.6%) and lower false alarm rate (1.4%) than did the simpler statistical model (93.1%/6.9%), representative of the detection mechanisms commonly used by commercial DoS detection systems. Because of the high ratio of large-byte packets in our collected data, we found that DoS attacks composed of large-byte packets were not as easily detected as those composed of 64-byte packets. We also found that a reasonable detection rate can be achieved with only one month of training data. Finally, we validated our simulation approach using a live network testbed and staging two DoS attacks over emulated network traffic, to test a prototype detection system built from our machine learning model. The prototype system achieved a 96.2% detection rate with a 3.8% (9 FAs/day) false alarm rate.

8. References

1. *Computer Crime And Security Survey*. 2003, CSI - Computer Security Institute, www.security.fsu.edu/docs/FBI2003.pdf.
2. *Denial of Service*. 2003, http://www.cert.org/tech_tips/denial_of_service.html.
3. Menkus, B., *Understanding the Denial of Service Threat*. Information Systems Security, 1997: p. 3-8.
4. C. Schuba, I. Krusl, Kuhn, M., and Spafford, E. *Analysis of a Denial of Service Attack on TCP*. in *IEEE Symposium on Security and Privacy*. 1997. Oakland, CA.

5. F. Lau, S. H. Rubin, M. H. Smith, and Lj. Trajkovic. *Distributed Denial of Service Attacks*. in *2000 IEEE International Conference on Systems, Man, and Cybernetics*. 2000. Nashville, TN: IEEE.
6. *Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks*. 2000, CISCO, www.cisco.com/warp/public/707/newsflash.pdf.
7. Bellovin, S., *Distributed Denial of Service Attacks*. 2001, <http://www.research.att.com/~smb/talks/ddos/index.htm>.
8. Moore, D., Voelker, G. M., and Savage, S., *Inferring Internet Denial-of-Service Activity*. Proceedings of the 10th USENIX Security Symposium, 2001.
9. Csi/Fbi, *Computer Crime And Security Survey*. 2003, CSI - Computer Security Institute, www.security.fsu.edu/docs/FBI2003.pdf.
10. K. Houle, G Weaver, *Trends in Denial of Service Attack Technology*. 2001, CERT Coordination Center, <http://www.ietf.org/ids.by.wg/rmonmib.html>.
11. Cert, *NIMDA Worm/Virus Report -- Final*. 2001, CERT, <http://www.ietf.org/ids.by.wg/rmonmib.html>.
12. J. Mirkovic, G. Prier, P. Reiher. *Source-End DDoS Defense*. in *Second IEEE International Symposium on Network Computing and Applications (NCA'03)*. 2003: IEEE Computer Society.
13. Forristal, J., *Fireproofing Against DoS Attacks*. 2001, <http://www.ietf.org/ids.by.wg/rmonmib.html>.
14. T. M. Gil, M. Poletto, *Multops: a data-structure for bandwidth attack detection*. Proceedings of the 10th USENIX Security Symposium, 2001.
15. D. Moore, G. M. Voelker, S. Savage, *Inferring Internet Denial-of-Service Activity*. Proceedings of the 10th USENIX Security Symposium, 2001.
16. T. Saitoh, G. Mansfield, N. Shiratori. *Network Congestion Monitoring and Detection using the IMI infrastructure*. in *1999 International Conference on Parallel Processing*. 1999. Wakamatsu, Japan: IEEE.
17. A. Hussain, J. Heidemann, C. Papadopoulos. *A framework for classifying denial of service attacks*. 2003.
18. Aljifri, H., *IP Traceback: A new Denial-of-Service Deterrent*. IEEE Security & Privacy, 2003: p. 24 -31.
19. K. Park, H. Lee. *On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack*. in *IEEE Infocom*. 2001.
20. S. Shyne, A. Hovak, J. Riolo. *Using Active Networking to Thwart Distributed Denial of Service Attacks*. in *IEEE Aerospace Conference Papers*. 2001.
21. Moriarty, K., *Inter-provider coordination for Real-time Tracebacks*. 2003, www.nanog.org/mtg-0306/pdf/moriarty.pdf.
22. M. Crovella, A. Bestavros, *Self-similarity in World Wide Web traffic. Evidence and possible causes*. IEEE/ACM Transactions on Networking, 1997. **5**(6): p. 835-846.
23. Waldbusser, S., *RFC 2819*. 2000, <ftp://ftp.isi.edu/in-notes/rfc2819.txt>.
24. J. Case, M. Fedor, *RFC 1157*. 1990, <ftp://ftp.isi.edu/in-notes/rfc1157.txt>.
25. *Baystack 380 Features*, http://www.nortelnetworks.com/products/02/bstk/switches/baystack_380/features.html.

26. J.B.D. Cabrera, L Lewis Xinzhou Qin, Wenke Lee, R.K. Prasanth, B. Ravichandran, R.K. Mehra, *Proactive detection of distributed denial of service attacks using MIB traffic variables - a feasibility study*. Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium On, 2001: p. 609-622.
27. Waldbusser, S, *RFC 2021*. 1997, <ftp://ftp.isi.edu/in-notes/rfc2021.txt>.
28. T. Peng, C. Leckie, K. Ramamohanarao. *Protection from Distributed Denial of Service Attacks Using History-based IP Filtering*. in *IEEE International Conference on Communications (ICC 2003)*. 2003. Anchorage, Alaska, USA: IEEE.
29. D. Gregg, W. Blackert, D. Heinbuch, D Frunanage. *Analyzing Denial of Service Attacks using Theory and Modeling Simulation*. in *Workshop on Information Assurance and Security*. 2001. United States Military Academy, West Point, NY.
30. W. Blackert, D. Gregg, A. Castner, E. Kyle, R. Hom, R. Jokerst. *Analyzing Interaction Between Distributed Denial of Service Attacks and Mitigation Technologies*. in *DARPA Information Survivability Conference and Exposition (DISCEX'03)*. 2003: IEEE Computer Society.
31. Mstream, *Analysis of the mstream DoS attack*. 2003, <http://www.securitybeat.com/articles/mstreamanalysis.shtml>.
32. Bsz, *The Binary Streaming Zero attack*. 2003, <http://www.ietf.org/ids.by.wg/rmonmib.html>.
33. D. Pack, W. Streilein, S. Webster, R. Cunningham. *Detecting HTTP Tunneling Activities*. in *2002 IEEE, Workshop on Information Assurance*,. 2002. United States Military Academy, West Point, NY: IEEE.
34. A. Martin, G Doddington, T. Kamm, M. Ordowski, M. Przybocki, *The DET Curve In Assessment of Detection Task Performance*. 2001, <http://www.ietf.org/ids.by.wg/rmonmib.html>.
35. M. Richard, R. Lippmann, *Neural Network Classifiers Estimate Bayesian a posteriori Probabilities*. *Neural Computation*, 1991. **3**: p. 461-483.
36. R. Lippmann, R. Cunningham, D. Fried, S. Garfinkel, A. Gorton, I. Graf, K. Kendall, D. Mcclung, D. Weber. S. Webster, D. Wyschogrod, M. Zissman. *The 1998 DARPA/AFRL Off-Line Intrusion Detection Evaluation*. in *First International Workshop on Recent Advances in Intrusion Detection*. 1998. Louvain-la-Neuve, Belgium.
37. J. Haines, L. Rossey, R. Lippmann and Cunningham, R. *Extending the DARPA Off-Line Intrusion Detection Evaluations*. in *DARPA Information Survivability Conference and Exposition II*. 2001.
38. M. Mahoney, P. Chan, *Learning Rules for Anomaly Detection of Hostile Network Traffic*. 2003, Florida Institute of Technology. p. 11.
39. A Lazarevic, L Ertöz, A. Ozgur, V. Kumar, J Srivastava, *A Comparative Study of Anomaly Detection Schemes in Intrusion Detection*. 2003, <http://www.ietf.org/ids.by.wg/rmonmib.html>.