EFFECTS OF ORDERED ACCESS LISTS IN FIREWALLS

Faheem Bukhatwa and Ahmed Patel
Computer Networks and Distributed Systems Research Group,
Department of Computer Science,
University College Dublin,
Belfield, Dublin 4, Ireland

ABSTRACT

Firewalls are hardware and software systems that protect a network from attacks coming from the Internet. Packet filtering firewalls are efficient, fast and provide a good level of security and have withstood the test of time. Firewalls based on packet filtering provide protection through granting or denying access to passing packets. Each individual incoming or outgoing packet is inspected against a number of rules in an access list. The result of this inspection determines the decision to be made. The great expansion in communication and the increased number of users on the Internet place more pressure on firewalls to provide greater security at higher performance levels without being the bottleneck of communication. This enforces the search for better or more efficient methods of implementing firewalls. In this paper we examine the affects that ordering of the rules in access lists has on the performance of packet filtering. This is done through simulation of a network device performing packet filtering. Simulation enables more experiments to be carried out under exact repeated conditions allowing comparisons to be made which otherwise may not be possible.

KEYWORDS

Firewalls, filtering, access lists, packet classification, simulation.

1. INTODUCTION

In recent years advances in computing and telecommunication technologies have expanded the computer systems capabilities and greatly expanded user requirements and available tools. As a consequence, users and their organisations are becoming more dependent on the services provided by their systems and computer networks (Muftic et al, 1994). Data, programs and information critical to functioning or even survival of an organisation are kept on computer systems and exchanged over telecommunication facilities. This trend raises the need for secure systems for processing and exchanging the information.

Computer networks are becoming very convenient targets for attacks and illegal operations. Security of a system must be addressed to prevent, detect and correct different forms of attacks. Firewall technology is used to protect networks, by being situated strategically at a single security screening station where the private network or the Intranet connects to the public Internet (ISO, 1988), see Figure 1. It can also be used to isolate sub-networks. A firewall is a computer, router or other communication device that filters access to the protected network (Schreiner, 1998).

Packet filtering refers to the basic operation performed by the firewall to inspect the packet header, verifying any of the fields in the packet header i.e. the IP address or the port etc, then accepting or rejecting the packet with no changes made. Filtering can be applied to incoming or outgoing packets or both. Inspecting the fields is done against a set of rules from a list called the access list. Packet filtering is transparent to the users or independent of the user's knowledge or intervention. Firewalls of this type are cheap, simple, fast, efficient and provide a good level of security (Habtamu, 2000). IP (Internet Protocol) level filtering has proved to be efficient and effective at improving system security (Schuba & Spafford, 1997). But the cost of filtering involved may still be a significant bottleneck (Ballew & Scott, 1997).

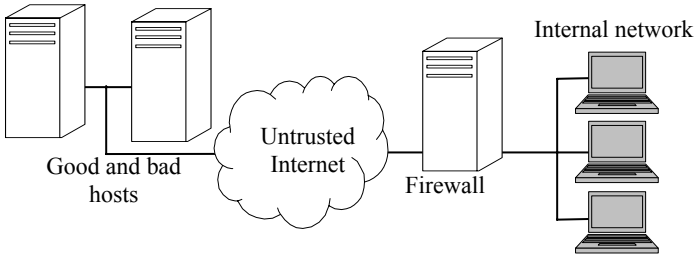

Figure 1. A typical firewall setup protecting an internal network

TCP/IP (Transmission Control Protocol/Internet Protocol) filtering is done by looking at the Network and Transport layer packet headers depending on the protocol suite. Figure 2 shows the header fields and their width in bits for packets at two levels of the protocol stack TCP and IP. Packet filters typically enables to manipulate (that is, accept or reject) packets based on a number of controls the most common are:

- the physical network interface that the packet arrives on
- the packet source and destination IP addresses
- the protocol of the packet, type of transport layer (e.g. UDP, TCP etc.)
- the transport layer source and destination ports (TCP/UDP)
- some flags (e.g. a TCP packet contains connection status flags).

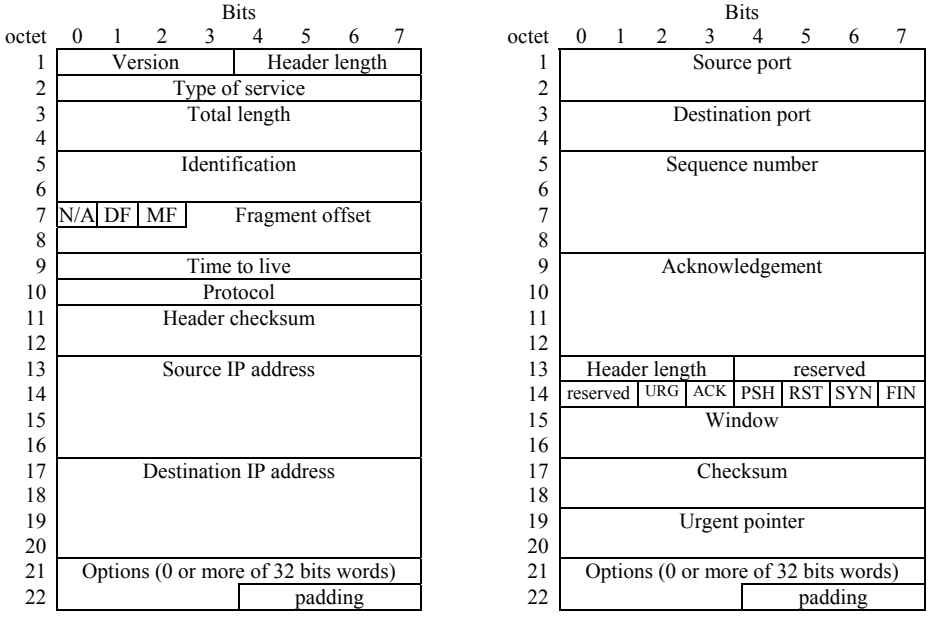

Figure 2. The Internet Protocol (IP) header structure (left) and the Transmission Control Protocol (TCP) header structure (right).

An access list is an ordered list of statements (rules) denying or permitting packets based on matching criteria contained within the packet headers (IP address, port number etc). In addition, there is an implicit statement at the end that says "Deny all" or "Accept all". Typically, the policy is either "deny any service not expressly permitted" or "permitting any service not expressly denied".

Rules can be in the following format: *If (condition) then action* where the action is either a packet "accept" or "reject". An example of a rule for a router (Cheswick & Bellovin, 1994) is as follows:

Access-list 101 permit TCP 20.9.17.8 0.0.0.0 , 121.11.127.20 0.0.0.0 , range 23 27

The rule indicates that any TCP protocol packet with an IP source address 20.9.17.8 and destination of IP address 121.11.127.20 is to be accepted provided the destination port address is in the range 23…27. If the condition does not match the rule then checking continues with the next rule. The rules are checked in a specific order and that order is very critical. Changing the order of the rules could result in a different decision of acceptance or rejection for some packets (Hazelhurst, 2001).

2. LIMITATIONS OF FIREWALLS

IP was not designed with security as a priority and is, as such, inherently insecure. Firewalls are a common method of addressing these insecurities but are often regarded the only line of defence (Haeni, 1997). Firewalls do not eliminate these insecurities entirely, but do make unauthorised access to a system from the Internet far more difficult. Indeed, packet interception and filtering can suffer from some difficulties:

- A firewall is not effective against users with authorised access or what is known as internal attacks.
- Most firewalls are not real defence against attacks using known software bugs or malicious code problems like mail bombs, viruses and Trojan horses etc.
- Reliance on accurate IP source addresses for making filtering decisions. IP addresses can be faked (Bellovin, 1992).
- The difficulty of correctly specifying (Chapman, 1992), changing and ordering access list filtering rules (Hazelhurst, 1999).
- Scalability, with reference to having a single point (firewall) where all traffic flowing through making it a central bandwidth bottleneck. Putting extra strain on keeping up with the growing packets processing computational cost due to increased traffic and increased sophistication of filtering required. Security versus performance dilemma (Friedman, 2001).
- Packet fragmentation complications. IP will fragment large transmission datagrams by breaking them into small data units or fragments. This is done so the data can be transferred over networks that support small maximum packet sizes, as well as perform the re-assembly of those datagrams when the data is received at the destination (Tanenbaum, 1996). This increases the complication for firewalls, in situations like when a fragment of a packet is accepted and another is rejected.

3. PACKET CLASSIFICATIONS

Internet routers that operate as firewalls, or provide a variety of service classes, perform different operations on different flows. A flow is defined to be all the packets sharing common header characteristics; for example a flow may be defined as all the packets between two specific IP addresses. Another flow may contain all packets that have the same source or destination IP addresses. Packets within a flow obey a pre-defined rule and are processed in a similar manner by the router.

In order to classify a packet, a router consults a table (or classifier) using one or more fields from the packet header to search for the corresponding flow. The classifier is a set of rules that identify each flow and the actions to be performed on each. Packet classification is the process of categorising packets into “flows” in an Internet router based on specified collection of rules.

Packet classification is employed by Internet Routers to implement a number of Internet services, such as routing, rate limiting, access-control in firewalls, resource reservation such as virtual bandwidth allocation, policy-based routing, service differentiation, Virtual Private Networks (VPN), IP security gateways and quality of service (QoS), load balancing, traffic shaping, and traffic billing services (Feldmann & Muthukrishnan, 2000). While in firewalls, rules are mainly used to accept or deny a packet, traditionally in a router they are used to ultimately find the IP address of the next hop where the packet needs to be routed to.

4. MODELLING THE ACCESS LIST OPERATION

The way access list rules operate is that a packet is inspected against each rule in the access list until a clear accept or reject rule is met which applies to the packet. Otherwise, all the rules in the list are checked. From a performance point of view, it is desirable that each packet meets that critical (accept or refuse) rule at or near the start of the list. This would reduce the time required for the packet to be inspected and consequently improve performance. If the critical rule for a packet is met at or near the end of the list, or the rule is never met, the processing time for the packet will be very high and performance will degrade.

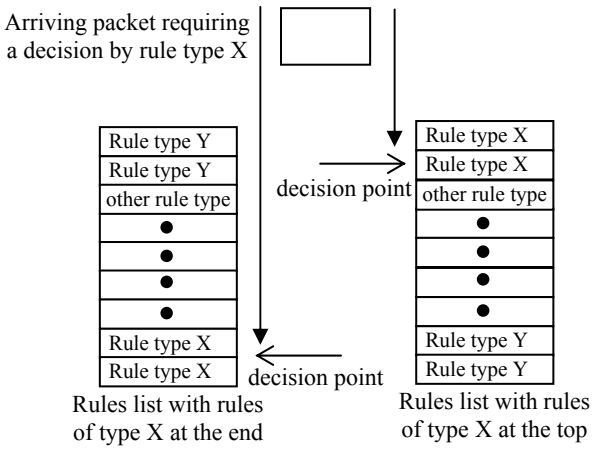

Figure 3. Effects of rules list type ordering on arriving packets.

It is feasible to classify access list rules into different types based on a number of factors. For example, rules can be divided into those that are intended for inspecting and filtering based only on the source address or only on the target address or on both. Also, rules can be intended for checking and filtering based on the port number in the packet or the protocol used. In fact, rules can be based on any combination of the fields in the packet, that is one field or all the fields if relevant. So, it is proposed for the purpose of this work to first classify rules in an access list into different types and secondly, order the rules in the list according to their type. Test runs are then carried out to test the performance of different orders of the same rules with the same set of packets arriving into the system.

Considering this approach then a number of assumptions can be made. Consider that rules are ordered in a particular way such that all rules of a certain type, let us say type "X" are made to be located at the top of the access list. When packets start arriving, and if all arriving packets or most of them are accepted or refused by a rule of type "X", then our ordering of the rules list was a successful choice and performance is expected to improve, see figure 3. The reason is that all or most packets need only a few rules to be checked at the start of the list before a deciding rule for acceptance or refusal is reached. While, if the list was ordered in such a way that "X" type rules are placed at the end of the list. Then for each packet arriving, most of the rules in the list have to be checked before eventually reaching the needed "X" type rule at the end of the list.

Each rule the packet is inspected against has a time cost attached to it, and the more rules the packet has to pass through the more time is wasted. The ultimate aim here is to reduce this time required for processing each packet, in other words, reducing the average processing time needed per packet. A cost time is associated with each rule-type that reflects the computing complexity of the rule. For example the rule that refuses a packet if it belongs to a particular protocol performs much less computation than a rule which will accept a packet that belongs to a particular protocol and comes from a specific address and goes to a particular address on a particular port.

One other assumption was made here that it was known in advance for each arriving packet what type of a filtering rule it required to determine acceptance or refusal of the packet. If every time a packet arrives it is possible to recognize the type of rules it requires, and also, if the list of rules then can be sorted according to the packet requirement then packets will meet their critical rules in the list early at the start of the list. Unfortunately, this approach of reordering the list of rules based on individual packet requirement is neither practical nor possible from a performance point of view.

We are suggesting that if a profile of arriving packets can be developed over a period of time for a network device then it will be possible to determine the pattern of these packets (Gupta, and McKewon, 2001). Classification of packets arriving in the past history of a device can determine types and numbers of packets arriving, which makes a pattern. This pattern can periodically be inspected to determine the most effective order of the rules list. A pattern consists of different packet flows. Patterns differ in the number of packets in each flow. In this simulation different packet patterns were generated by altering the number of packets in the flows that make up the pattern. For example, consider a pattern having two flows one receives four as many packets than the other. This can be simulated by randomly generating 20% of packets needing a

rule type 1 to decide their fate, and 80% of packets needing rule type 2. It was possible using this method to generate many different flows for experimentation.

Another assumption is being made that access lists can relatively easily be reordered to facilitate better performance for a particular pattern of arriving packets. Well, that is not the case, as changing the order of the rules within the list can in fact change the acceptance or refusal of a packet (Hazelhurst, 1999).

Reordering access rules can be a difficult and dangerous operation. The real difficulty is in ensuring that the list is or remains an accurate translation of the security policy in as far as permitting those packets intended to pass and blocking those intended to be blocked. Ordered lists require some form of validation to ensure truthful adherence with the security policy. Part of the problem is the time constraints on actually reordering and verifying. This problem of the effects of ordering the list on the security policy is not the subject for discussion in this paper. Though, one way of eliminating this time constraints problem is to have few versions of the access list pre-ordered and verified (Hazelhurst, 2001). So, in real time no actual ordering takes place, only a selection of the most suitable order of the access list is made.

The following characteristics were concluded after considering results collected from over 793 packet classifiers from 101 ISPs (Internet Service Providers), with a total of 41,505 rules (Stoica, 2001):

- That the mean number of rules = 50 rules. The max=2734 rules, and only 0.7% contained over 1000 rules.
- Many fields are specified by range, e.g., port number equal to and greater than 1024, or between 20 and 40. This is more usual than specifying a single value e.g. port number = 1024.
- 14% of classifiers had a rule with a non-contiguous mask
- Rules in the same classifier tend to share the same fields
- 8% of the rules are redundant, i.e., they can be eliminated without changing classifier's behaviour.

5. IMPLEMENTATION AND EVALUATION

The steps involved can be summarised in the following scenario:

- Classifying of access list rules into different types based on the inspection required in each rule.
- Generation of an access list then producing different copies of the same list in different ordering based on types of rules.
- Classifying arriving packets based on values contained in their headers' fields. Determine a pattern of packets arrival based on this classification.
- Generation of a number of sets of packets based on these patterns and save such sets. This will enable using exact same patterns with different orders of the same access list rules.
- Observing the results of the simulation runs and making the appropriate comparisons and extracting the conclusions.

A number of lists representing packets were generated, each list resembling a different pattern of packets arriving into a network device. The patterns were emphasised by the different percentages of packets in each list depending on the packet classification. In this sense packet classification reflects the type of rule that will decide the fate of the packet (accept or reject). To perform the simulation a number of access lists of varying mixture of rule types were generated. Rules were broken into types: type 1, type 2 etc. based on the different field combinations they are inspecting. For each list a number of copies were regenerated in different orders based on the type of rules. For the same access list, all copies of its individual ordered lists were tested with each of the packet patterns. In other words, for all possible patterns of packets, all different orders of the same access list were simulated. The results for which were observed. The main factor of performance was the length of time taken to process all packets in a pattern, and therefore, the average processing time for each packet.

Table 1. Execution runs of packets 80% of same class of patterns checked by three access lists

Access list rules (2200 rules)	80% source address check packet pattern (class 1 packet)	80% port number check packet pattern (class 6 packet)
	Average processing time per packet in unit time	
Unsorted rules	36,337	33,635
Sorted with source address checks first (rule type 1)	17,863	62,456
Sorted with port number checks first (rule type 6)	54,924	10,099

All other variables and conditions that may influence the results are kept unchanged in all simulations where appropriate. For the same packet pattern the variables are the total number of packets, frequency of packets arriving (time between packets). While for the access lists, the same total number of rules, and the same number of rules for each type, also the processing time for same type of rules.

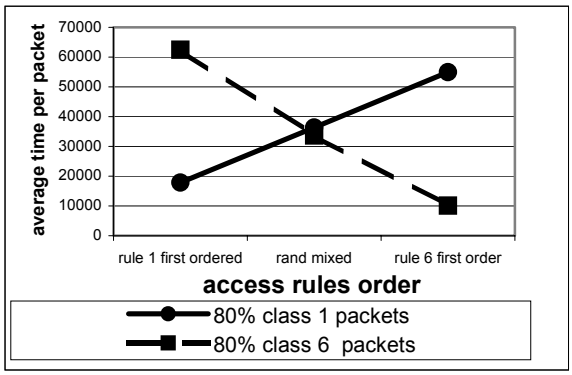

Figure 4. Effects of ordered access rules on different packet patterns

Table 1 shows the results of two simulation executions performed. Two packet patterns were generated; one contained 80% of its packets requiring source address inspection (class 1), while the second contained 80% of its packets requiring port number inspection (class 6). The other 20% of the packets consist of a mix of other classes of packets. There were 2200 rules in the access list. These two lists are considered extreme cases in most networks, but for some network devices these figures can be a good reflection of real life.

Both these two packet patterns were tested by three versions of the same access list rules. One unsorted, one sorted with all the rules for testing the source address (rule type 1) are at the top of the list, and the third being sorted with all the rules for testing the port number requests (rule type 6) placed at the top of the list. Six simulation runs were executed and the results are shown in Table 1. Similar results in line with these results in Table 1 were observed on all executions under varying other values such as the number of packets, other classifications of packets on one hand, and the varying numbers and types of access list rules on the other.

In Table 1, shaded area marks the two best performances represented by the lowest average processing time per packet in each pattern. The unsorted access list performance falls between the other two best and worst order scenarios. The unsorted list performance is near the average performance of the other two sorts. In fact, in some of the cases it is actually slightly worse than the average of all the different orders used.

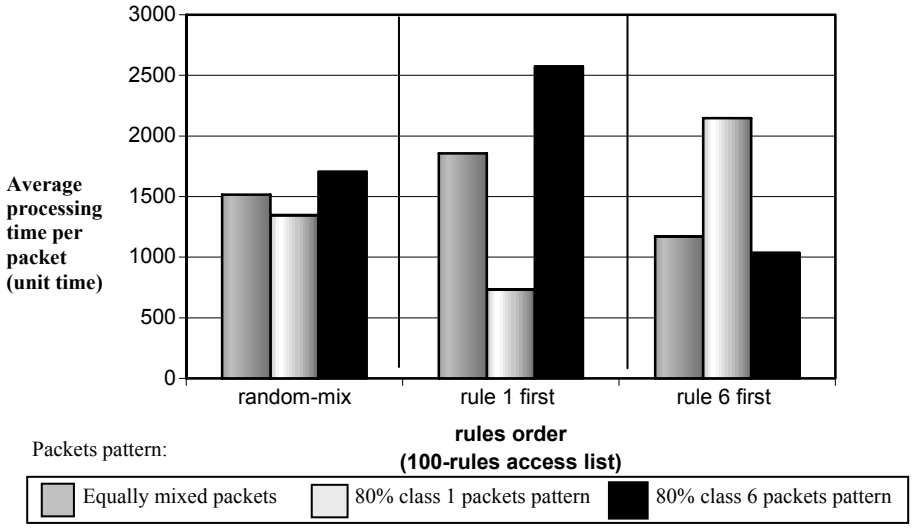

Figure 5. Execution runs of three packet patterns checked by one access list in three different orders

For the 80% source address check packet pattern (the middle column in Table 1), the unsorted rules performance (average processing time per packet) was: 36337 unit time, while the average of the other two ordered lists is (54924+36393)/2=36393.5 . While, for the 80% port number check packet pattern (the third column in Table 1), the unsorted rules performance was: 33635, while the average of the other two ordered lists is (62456+10099)/2=36277.5 .

Figure 4 shows that the lowest values of average packet service time through a list depend on both the order of the list and the packet pattern. Even if the pattern of arriving packets in real life in a network device fluctuates from the expected, better performance will still be obtained from an ordered list even if it is not the most desired order for the pattern. Provided the pattern of arriving packets does not go in a complete reverse for the entire period of operation. Such changes of patterns should be the cue for the change of the order of the access list used for the more suitable order. A pattern analyzer program can easily make the selection of the best order for the new pattern.

Figure 5 is a chart showing the results of nine simulation runs where three packet patterns were used, one consisted of equal mix of packet classes, the other two consisted of mostly (80%) of one particular class of packets we called them class 1 and 6. Three orders of the same access list were used, one random list, one with rule type 1 on top and the third with rule types 6 on top of the list. There were 100 rules in this access list used in all these simulations. The values of the average packet service time obtained were considerably smaller than those indicated in Table 1 because of the smaller size access list used this time.

6. CONCLUSION

In this paper we simulated ordering the rules in an access list and analysed its effect on performance of packet filtering. The results of the simulation showed clearly that ordering the access list can have an effect when considered with the pattern of the arriving packets to the device. The results indicate that different patterns of packets arriving into a network device will require specific access list order to yield the best performance. Performance as measured by the average processing time per packet to filter through the list.

Future work involves two specific points that came up as a result of this work. The first is the development of an automatic analyser of packet patterns arriving to a network device and making the appropriate selection of the most suitable ordered version of the access list available. Such an application can be executed at different time intervals depending on the nature of changing patterns of packets arriving at a particular network device. Such application will require the log of packets arriving to a communication device be kept for analysis. Real time analysis is a possibility to be considered, but the time constraints on such devices may make it impossible or places the device dangerously close to the performance limitations.

The second area to be considered for future work is the operations of reordering of an access list and verification of adherence to the security policy. Such applications would prove to be invaluable tools for network managers.

REFERENCES

Alexander D.S. et al, 2001. The process of safety in an active network. *IEEE/KICS Journal of Communications and Networks (JCN)*, March 2001.

Ballew, Scott M. 1997. *Managing IP Networks with Cisco routers*. O'Reilly, 1st edition. October 1997.

Bellovin, 1992. *There Be Dragons*, Proceedings of the Third USENIX UNIX Security Symposium, Baltimore, MD, September 1992

Chapman, D., 1992. *Network (In)Security Through IP Packet Filtering* Proceedings of third USENIX UNIX Security. Symposium, Baltimore, MD September 1992

Cheswick, W.R. and Bellovin, S.M. 1994. *Firewalls and Internet Security, Repelling the Wily Hacker*, Addison-Wesley.

Feldmann, A. and Muthukrishnan, S., 2000. *Tadeoffs for Packet Classification* Proceedings of IEEE Infocom 2000, Vol (March)

Friedman David, and Nagle David, 2001. *Building Firewalls with Intelligent Network Interface Card*s. Technical Report CMU–CS–00–173. CMU, May 2001.

Gupta, P., and McKewon, N., 2001. *Algorithms for Packet Classification* IEEE Network Special Issue, March/April 2001, vol. 15, no. 2, pp 24-32

Habtamu, Abie, 2000. *An Overview of Firewall Technologies* Norwegian Computing Centre January 2000.

Haeni, T., 1997. *Firewall Penetration Testing* The George Washington University January 1997

Hazelhurst, S., 1999. *Algorithms for Analysing Firewall and Router Access Lists*", Techincal Report TR-Wits-Cs-1999-5, Dept. of Computer Science, University of Witwatersand, South Africa July 1999.

Hazelhurst, S., 2001. A proposal for Dynamic Access Lists for TCP/IP Packet Filtering. *SAICSIT 2001*. Pretoria, South Africa.

ISO, 1988. *Information Processing Systems OSI RM, Part 2: Security Architecture*, ISO/TC 97 7498-2.

Muftic, S. et al, 1994. *Security Architecture for Open Distributed Systems*. Wiley, Dublin, October 1994.

Schreiner, R., 1998, *CORBA Firewall White Papers*, TechnoSec Ltd.

Schuba, C.L. and Spafford, E.H., 1997, A Reference Model for Firewall Technology. *In Proceedings of the Thirteenth Annual Computer Security Applications Conference*, December 1997.

Stoica, Ion, 2001. Rout lookup and packet classification CS 268, February 2001. Available from: http://www-inst.eecs.berkeley.edu/~cs268/sp03/ [Accessed 06 May 2003]

Tanenbaum, A., 1996 *Computer Networks*, Prentice Hall, Upper Saddle River, New Jersey, USA.