Acl Compressor

S.Kavitha,
Associate professor, Faculty of computer Applications,
Aarupadai Veedu Institute of Technology,VMU,Paiyanoor,Chennai.

Abstract *: An access control list (ACL) provides security for a private network by controlling the flow of incoming and outgoing packets. Specifically, a network policy is created in the form of a sequence of (possibly conflicting) rules. Each packet is compared against this ACL, and the first rule that the packet matches defines the decision for that packet. The size of ACLs has been increasing rapidly due to the explosive growth of Internet-based applications and malicious attacks. This increase in size degrades network performance and increases management complexity. In this paper, we propose ACL Compressor, a framework that can significantly reduce the number of rules in an access control list while maintaining the same semantics. We make three major contributions. First, we propose an optimal solution using dynamic programming techniques for compressing one- dimensional range based access control lists. Second, we present a systematic approach for compressing multi-dimensional access control lists. Last, we conducted extensive experiments to evaluate ACL Compressor. In terms of effectiveness, ACL Compressor achieves an average compression ratio of 50.22% on real-life rule sets. In terms of efficiency, ACL runs in seconds, even for large ACLs with thousands of rules.*

Index Terms: *Access Control List, Packet Classification, Fire- wall, Algorithm.*

I. INTRODUCTION

A. Background

Access control lists (ACLs) represent a critical component of network security. They are deployed at all points of entry between a private network and the outside Internet to monitor all incoming and outgoing packets. A packet can be viewed as a tuple with a finite number of fields such as source/destination IP addresses, source/destination port numbers, and the protocol type. The function of an ACL is to examine every packet's field values and decide how to enforce the network policy. This policy is specified as a sequence of (possibly conflicting) rules. Each rule in an ACL has a predicate over some packet header fields and a decision to be performed upon the packets that match the predicate. A rule that examines d-dimensional fields can be viewed as a d-dimensional object. Real-life ACLs are typically 4-dimensional (over 4 packet fields: source IP address, destination IP address, destination port number, and protocol type) or 5-dimensional (over 5 packet fields: source IP address, destination IP address, source port number, destination port number, and protocol type).When a packet comes to an ACL, the network device searches for the first (i.e., highest priority) rule that the packet Matches, and executes the decision of that rule. Two ACLs are equivalent if and only if they have the same decision for every possible packet. Table I shows an example ACL where the format of the four rules is based upon that used in ACLs on Cisco routers.

Rule	SIP	DIP	SPort	DPort	Proto	Act
1	192.168.*.*	*	[4000,		TCP	discard
2	192.168.*.*	*	[0, 3999]		TCP	accept
3	192.168.*.*	*	[5001,		TCP	accept
4	*	*	*	*	*	discard

TABLE I
AN EXAMPLE ACL

In this paper, we study a general ACL compression problem: given an ACL f , generate another ACL f i that is semantically equivalent to f but has the minimum possible number of rules. We call this process "ACL compression". We focus on five versions of ACL compression that differ only in the format of field constraints of the output ACL: (1) range ACL compression where field constraints are specified by a range of integers (e.g., source port e [5000, 6000]), (2) prefix ACL compression where field constraints are specified by a prefix string (e.g., source IP = 192.168. * .*), (3) ternary ACL compression, where field constraints are specified by a ternary (including prefix) string (e.g., source IP = 192. * .0.*), (4)range-prefix ACL compression where some field constraints are specified by ranges and the remaining field constraints are specified by prefix strings, and (5)

range-ternary ACL compression where some field constraints are specified by ranges and the remaining field constraints are specified by ternary strings. In most ACLs, the source port number and destination port number fields use a range field constraint whereas the source IP address, destination IP address, and protocol type fields use a prefix or ternary field constraint.We give an example that illustrates the possibilities of ACL compression. The input ACL with five rules is depicted in Figure 1(A). For simplicity, we assume this ACL only examines one packet field F , the domain of F is [1, 100], and F uses a range field constraint. The geometric representation of this five rule ACL is given in Figure 1(a) where the predicate of each rule is a line segment, the decision of each rule is the color of its line segment, a packet corresponds to a point on the line, and the decision for a packet is the color of the first line segment that contains the point. To generate another sequence of rules that is equivalent to the ACL in Figure 1(A) but with the minimum number of rules, we first decompose the five rules into non-overlapping rules as shown in Figure 1(B). The geometric representation of these five non-overlapping

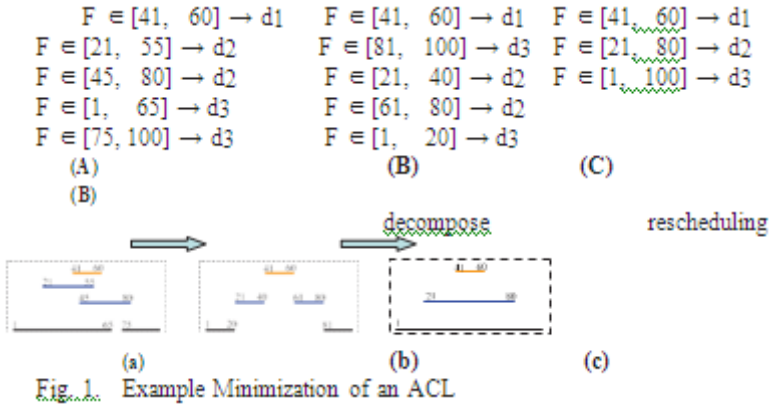

Fig. 1. Example Minimization of an ACL

rules is in Figure 1(b). We now reschedule the intervals to generate a shorter semantically equivalent ACL as follows. We first schedule the interval [41, 60]. This allows us to schedule the two intervals [21, 40] and [61, 80] together using one interval [21, 80] based on first-match semantics. Finally, we can schedule intervals [1, 20] and [81, 100] together using one interval [1, 100] again based on first-match semantics. The three ACLs in Figures 1(A), 1(B) and 1(C) are equivalent, but the rightmost ACL has fewer rules.

Our work on ACL compression has two important motiva-tions. First, ACL compression is useful for network system management and optimization because minimizing large ACL rule sets greatly reduces the complexity of managing and optimizing network configurations. As a result, ACL com- pression tools in general and our ACL compression tool in particular have been used or proposed for use in several prominent network management and optimization projects, such as Yu et al.'s DIFANE work [18] and Sung et al.'s work on systematic design of enterprise networks [16], [17]. Second, some network products have hard constraints on the number of rules that they support. For example, NetScreen-100 only allows ACLs with at most 733 rules. ACL compression may allow users with larger ACLs to still use such devices. This may become an increasingly important issue for many users as ACL size has grown dramatically due to an increase in Internet applications and services as well as an increase in known vulnerabilities, threats, and attacks [2]. For example, our older ACLs have at most 660 rules whereas the ACLs we have more recently acquired have as many as 7652 rules.

B. Summary and Limitations of Prior Art

The main limitation of prior work is, to the best of our knowledge, the lack of work on two key ACL compression problems. First, no prior work has considered range ACL

Fig. 2. Converting a schedule to a canonical schedule

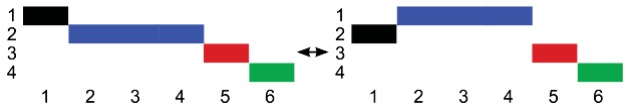

Fig. 3. Swapping two adjacent intervals

For one-dimensional range and prefix ACLs, we achieve optimal compression. Finally we combine the many one-dimensional ACL compression solutions into one multi- dimensional solution to the original multi-dimensional ACL minimization problem. Our approach has two key features. First, the hierarchical representation of ACLs is canonical. That is, two semantically equivalent ACLs will have the same hierarchical representation no matter how they are specified. Thus, our approach eliminates variance due to human factors in the design of given ACLs. Second, our approach allows range, prefix, and ternary fields to be optimized independently using customized algorithms because it deals with one field at a time. We name our approach "*ACL Compressor*".

D. Key Contributions

In this paper, we make three key contributions:1)propose an optimal algorithm for the one-dimensional range ACL compression problem. This algorithm uses dynamic programming techniques. (2) We present a systematic and efficient framework for generating good solutions to the NP- hard multi-dimensional range, range-prefix, and range-ternary ACL compression problems. Our framework combines the locally optimized one-dimensional solutions into a good but not necessarily optimal multi-dimensional solution. (3) We conducted extensive experiments on both real-life and syn- thetic ACLs. The results show that ACL Compressor achieves an average compression ratio of 50.22% on real-life range- prefix ACLs.ACL Compressor is designed to run off-line so that networkmanagers do not need to read or manage the compressed ACL.

II. CONCLUSION

An optimal access control list is an access list that satisfies security requirements with the least amount of processing overhead. In this paper, we have presented several techniques and algorithms for access control list optimization. Some of these algorithms look for rules that can be safely removed, such as shadowed and covered rules, and rules that can be combined in order to reduce the size of ACLs and, subsequently, reduce expected packet latency. Other algorithms reorder the rules in an ACL based on three factors: actual hit counts, hit counts prediction factor, and rule latencies. It was found empirically that Hits Optimizer and Rules Combining procedures yield the greatest bulk of optimization since they are harder to handle manually by average network administrators. The algorithms can be easily customized, where time is reduced at the expense of efficiency, and can be implemented partially or fully,both online and offline.

REFERENCES

[1] A. Velte and T. Velte. "Cisco: A Beginner's Guide", McGraw-Hill Inc. 3rd edition (2004).
[2] Access Control Lists, Cisco Systems, USA, (http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/scacls.htm).
[3] V. Grout, J. McGinn, and J.Davies. "Real-Time Optimisation of Access Control Lists for Efficient Internet Packet Filtering", Journal of Heuristics, Vol. 12, 2006.
[4] E. Al-Shaer and H. Hamed. "Firewall Policy Advisor for Anomaly Detection and Rule Editing." IEEE/IFIP Integrated Management Conference (IM'2003), March 2003.
[5] Bukhatwa, F., (2004) High Cost Elimination Method for Best Class Permutation in Access Lists, ICWI 2004, pp287-294