Memory Efficient Regular Expression Pattern Matching Architecture for Network Intrusion Detection Systems

Pawan Kumar

MSc Engg Thesis
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore

Abstract

The rampant growth of the Internet has been coupled with an equivalent growth in cyber crime over the Internet. With our increased reliance on the Internet for commerce, social networking, information acquisition, and information exchange, intruders have found financial, political, and military motives for their actions. Network Intrusion Detection Systems (NIDSs) intercept the traffic at an organization's periphery and try to detect intrusion attempts. Signature-based NIDSs compare the packet to a signature database consisting of known attacks and malicious packet fingerprints. The signatures use regular expressions to model these intrusion activities.

This thesis presents a memory efficient pattern matching system for the class of regular expressions appearing frequently in the NIDS signatures. Proposed Cascaded Automata Architecture is based on two stage automata. The first stage recognizes the sub-strings and character classes present in the regular expression. The second stage consumes symbol generated by the first stage upon receiving input traffic symbols. The basic idea is to utilize the research done on string matching problem for regular expression pattern matching. We formally model the class of regular expressions mostly found in NIDS signatures. The challenges involved in using string matching algorithms for regular expression matching has been presented. We introduce length-bound transitions, counter-based states, and associated counter arrays in the second stage automata to address these challenges. The system uses length information along with counter arrays to keep track of overlapped sub-strings and character class based transition. We present efficient implementation techniques for counter arrays. The evaluation of the architecture on practical expressions from Snort rule set showed compression in number of states between 50% to 85%. Because of its smaller memory footprint, our solution is suitable for both software based implementations on network chips as well as FPGA based designs.