Intrusion Detection for Network Infrastructures

Steven Cheung Karl N. Levitt Calvin Ko

Department of Computer Science
University of California
Davis, CA 95616

Our goal is to design intrusion detection technologies for network infrastructures. This work is part of a larger development project "Intrusion Detection System for Large Networks" sponsored by ARPA. In this abstract, we describe our work on domain name systems and routers.

A domain name system (DNS) is a large distributed database indexed by "names"; a set of distributed name servers collaborate to provide the name service. For instance, DNS is responsible for mapping host names to IP addresses and vice versa.

Many distributed applications, such as file transfer, remote login, WWW, and electronic mail, rely on DNS. Thus compromises to DNS can cause denial of service to these applications. Some applications (e.g., *rlogin*) use name-based authentication. Attacking DNS can change the name-to-address mapping, and hence allows crackers to masquerade as a trusted host. From a broader perspective, DNS is interesting because of its distributed nature and the trust relationships among the name servers. Intrusion detection techniques for DNS may be applicable to other distributed services.

We propose a detection-confirmation strategy to protect DNS. Intrusion detection components are used to monitor the traffic going in or out of a server. The traffic is checked against our DNS "specifications" and deviations from these specifications are noted as anomalous. By specifications, we refer to the expected behaviors of DNS that are security-related. An example specification is "DNS packets from a server only contain information on which that server has authority". For each deviation, the intrusion detection component confirms with a server responsible for the corresponding part of the database. If the monitored traffic disagrees with the authoritative answer, this event is flagged as a possible attack. To illustrate the usefulness of our approach, we have written simple specifications that can detect the cache poisoning attack [Schuba].

Routers exchange control packets to reflect changes in a network, such as topology changes. Malicious routers may use bad control packets to cause network problems. We have worked on a problem in which routers may remove transit packets so as to prevent these packets from reaching their destinations. The situation becomes worse if the malicious routers also send out routing updates claiming that they have low-cost connections to (some) other routers. These malicious routers are called *blackholes*. If blackholes exist, routers in their neighborhood will direct (some of) their traffic to these blackholes. Thus the blackhole attack can cause widespread denial of service.

We have designed protocols to locate blackholes under some assumptions, which concern the connectivity of the network, the maximum number of blackholes occurring simultaneously, the routing protocol used, stability of the blackholes (i.e., permanent vs. probabilistic). Our results use a system diagnosis approach [Preparata, Metze, Chien]. First, the source S will send out a probe. Then a designated set of routers will report if they have received the probe. Note that bad routers may lie. Based on the reports, S performs diagnosis and either carries out another test or identifies a blackhole router.