In Poster Session of 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), Nice, France, May 2005.

Towards Distributed Network Intrusion Prevention with Respect to QoS Requirements

Andreas Hess, Mathias Bohge, Günter Schäfer
Telecommunication Networks Group
Technische Universität Berlin
Einsteinufer 25, 10587 Berlin, Germany
Email: $[$*hess, bohge, schaefer*$]$*@tkn.tu-berlin.de*

Abstract

An Intrusion Prevention System (IPS) analyzes each packet for malicious content before forwarding it and drops packets that originate by an intruder. To do so, the IPS has to be physically integrated into the network and needs to process the actual packets that run through it, instead of processing copies of the packets at some place outside the network. Therefore, independent of the way they are built, all IPS share the same problem — a decrease in performance of the network they try to protect. Therefore, the main objective in improving IPS performance is to develop an architecture that minimizes the overall delay and maximizes the network's throughput while ensuring a sufficient level of security.

Keywords

network security, intrusion prevention, programmable routers, load distribution

1. Distributed and Demand-driven Intrusion Prevention

In order to relieve the strain on end-users and administrators of continuously having to deal with today's massive amount of security challenges, we propose to install modular intrusion prevention systems (IPSs) on top of programmable routers as an additional line of defense. Our long-term goal is the efficient protection of the end-systems that are part of an administrative domain (AD) as for example shown in Figure 2. Each router in the AD can either be of passive or programmable nature. The only constraint to bear in mind when setting up the AD is that there has to be at least one programmable router on the path between the Internet and each subnet.

Assuming, for example, that subnet N_7 of figure 2 consists of three hosts whereas each one requests the installation of the same five protection services ($s_1, s_2, ..., s_5$), then $4^{5*3}(= 1.073.741.824)$ possibilities exist to fulfill the requirement of filtering all traffic between the Internet and all subnets. On a per router basis, we differentiate between three types of traffic: traffic that is forwarded by the router without being analyzed, traffic that must be analyzed by at least one security service that is running on the router and traffic that is filtered / blocked by the router without being analyzed.

Consequently, the question arises how to configure each router of the AD in order to satisfy the security requirements of all end-systems while simultaneously not decreasing the network performance.

0-7803-9088-1/05/$20.00 (C) 2005 IEEE

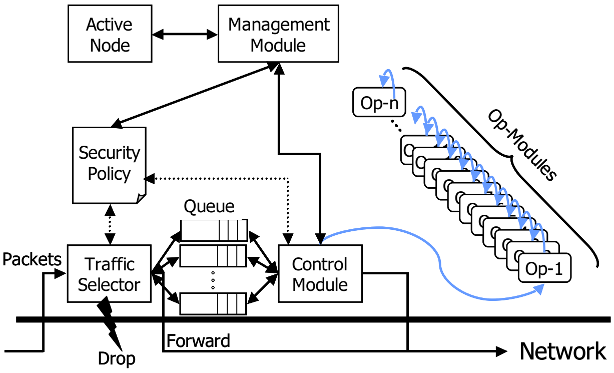


Figure 1: The FIDRAN-architecture

2. The FIDRAN Architecture

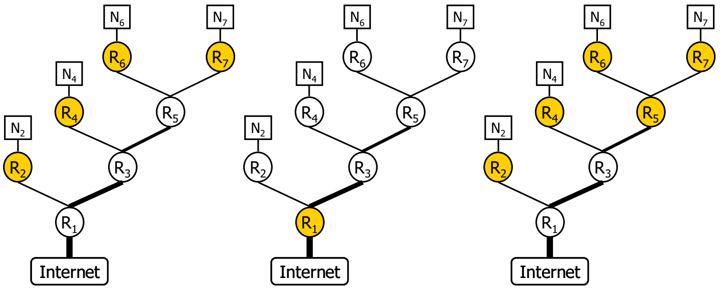
This section shortly describes the flexible intrusion detection and response framework for active networks (FIDRAN); for a detailed discussion we refer to [2] and [1]. The framework consists of core components that are required to run and of add-on components — the security services — which are dynamically integrated into the system when needed (cf. Figure 1). The core functionality comprises the traffic selector, the security policy, the control/management module and the default queuing discipline. Security services are implemented as **op**erational-modules featuring IPS specific networking services. The system is designed in a manner such that a dynamic reconfiguration at runtime — insertion and deletion of security services — is possible.

The complete network traffic is redirected to the traffic selector, which — according to the rules specified in the security policy — assigns the traffic to one of the categories: *forward*, *process* or *drop*. Traffic that is assigned to the category *forward* is directly forwarded and not analyzed by any installed op-module (see figure 1). Another task of the security policy is to inform the traffic selector about how to queue which network traffic of category *process*. Moreover, it specifies which traffic must be analyzed by which security services and how to react in case of a detected attack.

We evaluated the performance of the prototype by conducting several sets of experiments by varying the load (constant bit rate, UDP) and the number of integrated services on the FIDRAN host. In order to facilitate the task of comparing the results, all services were of the same type — delaying a packet either $10\mu s$ or $100\mu s$. Consecutively, we determined an approximation function (see equation 1 whereas λ represents the load in MBit/s and n the number of services) for the measured values for the service that delays a packet for $100\mu s$ which we used for a simulative study of three different security services deployment strategies.

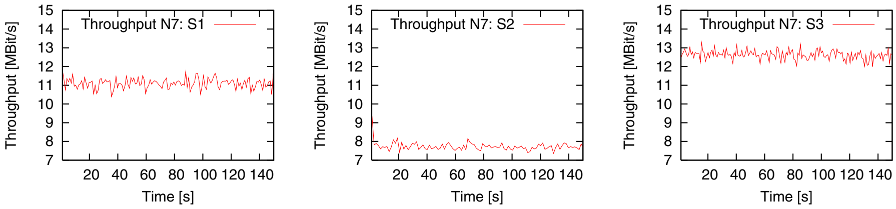
$$f(\lambda, n) = \begin{cases} -0.095680671 + 0,10422221 * n + 0.033666753 * \lambda; & 0 \leq \lambda \leq 5 \\ -0.64071440 + 0.15794931 * n + 0.095781562 * \lambda; & \lambda > 5 \end{cases} \quad (1)$$

0-7803-9088-1/05/$20.00 (C) 2005 IEEE



(a) S1: late (b) S2: early (c) S3: adaptive

Figure 2: Example deployment strategies



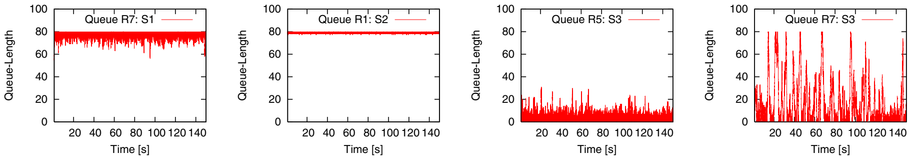
(a) S1: late (b) S2: early (c) S3: adaptive

Figure 3: Subnet N_7: Comparison of throughput (late, early, adaptive)

3. Security Service Deployment Strategies and Simulation

The Figures 2(a) and 2(b) show two oppositional deployment strategies. The *late*-deployment strategy places the requested security services as close as possible to the requesting subnet/end-system. In contrast thereto, the *early*-deployment strategy uses the first available router on the path from the Internet to the requesting subnet/end-system. Finally, Figure 2(c) depicts a strategy that is adapted to the situation, i.e. services for subnet N_7 are split among the programmable routers R_5 and R_7.

We simulated the three strategies using the *omnet++* simulation environment. As traffic source, we implemented two traffic generators, a constant bit-rate generator and a Poisson generator that were connected in parallel to the gateway router $R1$. We implemented the routers' core functionalities (receiving a packet, table lookup, forwarding, dropping) and added some programmable-node features, namely classifying incoming packets and processing the relevant ones. Instead of actually processing a packet, the programmable router in our simulation, just delays the packet before forwarding it. It computes the matching delay using the linear approximated delay function. A non-programmable router delays a packet $11\mu s$.



(a) S1: R7 (b) S2: R1 (c) S3: R5 (d) S2: R7

Figure 4: Router waiting queues (late, early, adaptive)

For the simulation, we made the following assumptions: about 80% of the complete Internet-traffic ($15MBit/s$) is destined to subnet N_7. The remaining traffic was uniformly distributed among networks N_2, N_4 and N_6 and each router is capable to buffer 80 packets. Additionally, all subnets request the same set of security services (three services whereas each one delays a packet $100\mu s$). The simulation results — focusing on subnet N_7 — in terms of throughput and router status are given in figures 3 and 4. The first row of figure 3 consists of the three curves, one for each strategy, depicting the throughput for subnet N_7. Accordingly, figure 4 depicts for each strategy the buffer-states of the programmable routers.

As a result, it can be seen that already for small ADs the decision of the IPS deployment can have a remarkable influence on the network performance. Consequently, it does make sense to split IPS among several routers in order to minimize the impact on the performance.

4. Future Work

In future work, we plan to study the influence of more complex deployment strategies on the network performance for bigger administrative domains. Moreover, we aim at developing a service deployment algorithm that allows to automatically distribute the requested services in an intelligent manner, as our long term goal is to extend our work towards a self-organizing network, that is able to autonomously recognize and satisfy the security requirements that are posed by the end-system of an administrative domain, while at the same time aiming to satisfy given QoS objectives.

References

[1] A. Hess, M. Jung, and G. Schaefer. Fidran: A flexible intrusion detection and response framework for active networks. In *8th IEEE Symposium on Computers and Communications (ISCC'2003)*, Kemer,Antalya,Turkey, July 2003.

[2] A. Hess and G. Schäfer. ISP-Operated Protection of Home Networks with FIDRAN. In *First IEEE Consumer Communications and Networking Conference (CCNC'2004)*, January 2004.