Distributed Detection of Large-Scale Attacks in the Internet*

Thomas Gamer
Institute of Telematics, University of Karlsruhe, Germany
gamer@tm.uka.de

ABSTRACT

Despite the many research activities that are performed in the field of attack prevention, detection, and mitigation, large-scale attacks like Distributed Denial-of-Service (DDoS) attacks still pose unpredictable threats to the Internet infrastructure and Internet-based business today. This paper outlines new mechanisms that facilitate a distributed real-time in-network attack detection. In addition, the foundations for a meaningful evaluation of large-scale detection mechanisms by means of simulations are laid.

1. INTRODUCTION

Large-scale attacks like distributed denial-of-service (DDoS) attacks or worm propagations in fact belong to the daily routine of Internet usage. According to [1] the number of attacks that are observed in the Internet today is even increasing. An early detection allows for a fast reaction—and thus, provides a basis for a suitable defense of the victims and the network.

In order to detect large-scale attacks, anomaly-based detection is applied due to the immense traffic flows within the network and the fact that various flooding attacks use protocol-conforming packets. To facilitate an in-network deployment, we build on a previously developed anomaly detection system [5] that works resource-saving by applying refinement of detection granularity. In general, anomaly-based detection systems usually return a set of detected anomalies as result of the detection process. It would, however, be much easier for taking countermeasures or for visualizing the network state if the detection system returns the attack detected instead of a set of anomalies. Most systems, furthermore, base their detection on local knowledge and observations only, i. e. they work as single independent instances. Especially within the network, this often leads to a high false negative error rate, e. g. due to unfavorable aggregation of attack flows or usage of packet sampling. Other approaches that actually use distributed knowledge frequently rely on close trust relationships, require a central control entity for communication, or are not able to perform real-time detection.

Taking the mentioned problems into account, we propose two mechanisms that improve detection quality and *facilitate a distributed real-time attack detection*:

- Collaboration of independent detection instances
- Identification of attacks prior to communication

The heterogeneity of routers deployed in today's Internet and the multitude of available anomaly detection methods lead to the situation that different detection systems scan for different sets of traffic anomalies. Thus, an identification of attacks is necessary to allow for a collaboration of multiple detection instances in heterogeneous environments.

Having developed new mechanisms for the distributed detection of large-scale attacks, the challenge of evaluating these mechanisms has to be faced. Establishment of real testbeds is expensive and maintenance is time-consuming and complex. In addition, Ringberg et al. [6] recently pointed out the need for simulation in attack detection but currently no feasible solution exists. Therefore, in Section 3 we present a methodology of evaluation by means of simulations.

2. DISTRIBUTED ATTACK DETECTION

Some existing identification mechanisms, e. g. parametric classifiers, require a stochastic model of involved parameters, which is not available in case of attack detection. Other mechanisms like rule-based identification and non-parametric classifiers cannot achieve both flexibility and high-quality results if applied on their own. Thus, we propose a two-stage identification system using a combination of the latter mechanisms. The rule-based first stage provides high-quality results in case of perfect information. If this first stage returns no result, a geometric classifier is started to eliminate the problems of the rule-based mechanism. Furthermore, the rule-based mechanism is able to control the processing of the detection system itself. Instead of a fixed execution sequence of anomaly detection methods, a dynamic

*I want to thank all my students that contributed with their excellent work to the research described in this paper.

ACM CoNEXT 2008, December 9-12, 2008, Madrid, SPAIN

iterative identification is performed. This means, the detection method is started next from which the highest information benefit is expected in the current situation. Since each method is executed only once during a particular identification, processing loops cannot occur.

The identification must be very flexible to cope with heterogeneous systems that make use of different sets of anomaly detection methods. To achieve this, we established a general model of those entities that form anomaly-based attack detection: attacks, anomalies, and anomaly detection methods. In a second step, the established model is combined with a description of actual attack characteristics as provided e. g. by [2]. This results in a concrete mapping of detected anomalies—required and optional ones—to described attacks. Necessary meta data for both mechanisms of our two-stage system can easily be derived from this model.

The collaboration for achievement of distributed attack detection is started after local detection and identification of an attack. Our approach is based on the assumptions that neither close trust relationships nor a central communication control entity exist. An instance first has to decide about which other instances to collaborate with. In our solution, only neighbors are considered for collaboration. The notion neighbor in this case is defined by the particular neighbor discovery mechanism used, e. g. path-coupled, ring-based or multicast-based discovery. Each neighbor actually receiving attack information has to take its own independent decision on how to react on this information. Therefore, a metric-based decision algorithm is applied that comprises parameters like local workload, distance, and significance. If this algorithm decides to process the message, a local fine-grained verification of the received information is started since it is not trusted implicitly. This ensures that attacks missed by the coarse-grained detection of the basic stage still can be detected due to collaboration. In future work, protection of the collaboration and further trust schemes have to be examined.

3. METHODOLOGY OF EVALUATION

If simulations are used for evaluation it must be ensured that reasonable simulation environments are generated. Thus, important aspects like topology, traffic, and attack flows have to be reproduced as realistic as possible. Examples for realistic characteristics are the hierarchical structure of topologies—on AS-level as well as on router-level within each AS—or the self-similar behavior of the aggregated traffic in the Internet. We developed *ReaSE* [4], which is able to generate such environments for OMNeT++ [7] with up to about 200 000 nodes. *ReaSE* is easily extensible, e. g. by additional traffic profiles that define specific application traffic. Furthermore, attack traffic simulating DDoS attacks is generated based on the real tool Tribe Flood Network.

Easy integration of existing real attack detection into the simulator OMNeT++ is provided by our framework *Distack* [3], which serves as a basis for the implementation of the proposed distributed attack detection. The `NetworkManager` of this framework also enables transparent deployment in different other runtime environments, e. g. Linux/Windows-based real systems. The `ModuleManager` allows for an integration of various anomaly detection methods, which the identification system presented in Section 2 iteratively starts according to their expected benefit. The collaboration of neighbor instances is facilitated by the modules `RemoteMessaging` and `Communication`, which allow for the integration of the decision algorithm as well as for the actual neighbor discovery. Figure 1 outlines all presented components for the evaluation of the distributed attack detection in large-scale environments.

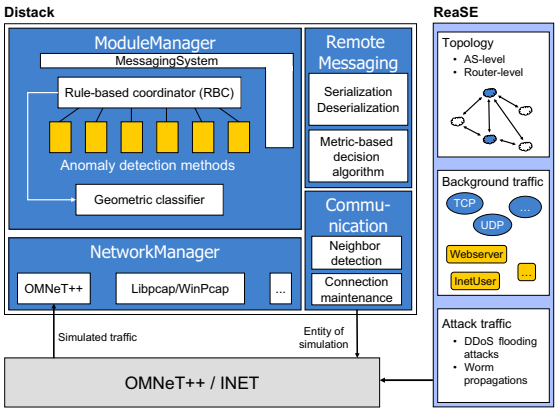

Figure 1: "The Big Picture"

Currently we are working on the implementation of the distributed attack detection into *Distack*. In future work, we have to conduct an evaluation of the proposed distributed attack detection.

4. REFERENCES

[1] Arbor Networks. Worldwide Infrastructure Security Report. http://www.arbornetworks.com/report, Sept. 2007.

[2] C. Douligeris and A. Mitrokotsa. DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art. *Computer Networks*, 44(5):643–666, Apr. 2004.

[3] T. Gamer, C. P. Mayer, and M. Zitterbart. Distack—A Framework for Anomaly-based Large-scale Attack Detection. In *Proc. of 2nd SecurWare*, pages 34–40, Aug. 2008. Available at *https://projekte.tm.uka.de/trac/Distack*.

[4] T. Gamer and M. Scharf. Realistic Simulation Environments for IP-based Networks. In *Proc. of the 1st OMNeT++ Workshop*, Mar. 2008. Available at *https://projekte.tm.uka.de/trac/ReaSE*.

[5] T. Gamer, M. Schöller, and R. Bless. An extensible and flexible System for Network Anomaly Detection. In *Proc. of Autonomic Networking*, pages 97–108, Sept. 2006.

[6] H. Ringberg, M. Roughan, and J. Rexford. The Need for Simulation in Evaluating Anomaly Detectors. *SIGCOMM Computer Communication Review*, 38(1):55–59, Jan. 2008.

[7] A. Varga. The OMNeT++ Discrete Event Simulation System. In *Proc. of 15th ESM*, pages 319–324, June 2001.