Anomaly-based Identification of Large-Scale Attacks

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

***Abstract*—Large-scale attacks like Distributed Denial-of-Service (DDoS) attacks still pose unpredictable threats to the Internet infrastructure and Internet-based business. Thus, many attack detection systems using various anomaly detection methods were developed in the past. These detection systems result in a set of anomalies detected by analysis of the traffic behavior. A realtime identification of the attack type that is represented by those anomalies simplifies important tasks like taking countermeasures and visualizing the network state. In addition, an identification facilitates a collaboration of distributed heterogeneous detection systems. In this paper, we first lay the foundations for a generalized identification system by establishing a model of those entities that form anomaly-based attack detection: large-scale attacks, anomalies, and anomaly detection methods. Based on this flexible model, an adaptable and resource-aware system for the identification of large-scale attacks is developed that additionally offers an autonomous processing control.**

I. Introduction

Internet users all over the world are still exposed to attacks and challenges on the Internet's availability today. Attacks like distributed denial-of-service (DDoS) attacks or worm propagations in fact belong to the daily routine of Internet usage. Such attacks in the following are called *large-scale attacks* since they typically rely on a large number of attackers that are widely distributed across the Internet. During the last few years the motivation for large-scale attacks has changed from gaining fame to gaining money, e. g. by blackmailing the victims with the threat of a DDoS attack or by selling huge botnets established by worm propagations. Those attacks and challenges mostly lead to high financial losses for the victims even if they last only for a short time—Amazon e. g. had an average sales volume of about 36 500 $ per minute [1] in 2008. In addition, according to the Worldwide Infrastructure Security Report [2] such attacks not only threaten victim host systems but also the infrastructure of Internet service providers.

Another reason for the increasing number [2] of large-scale attacks is that tools for launching them—e. g. the Tribe Flood Network [3]—are easily available in the Internet and easy to use with only little technical background. In addition, huge botnets lead to availability of very low-priced DDoS services: About $ 20 have to be paid for a DDoS attack that runs for an hour, about $ 100 for a whole day of DDoS [4].

Due to these permanent challenges many research activities have been performed in the past as well as presently in the field of attack detection. Known worms and viruses are mostly identified using signature-based detection systems like Snort [5]. Since DDoS attacks in the majority of cases use packets that are conform to protocol specifications, they cannot be detected by signatures. Detection of DDoS flooding attacks therefore requires anomaly-based systems like [6]–[9]. In this context, discrepancies from expected behavior, e. g. non protocol-conform behavior or abnormal increases of the traffic volume, are called *anomalies*.

An early detection in the core network instead of at the edges of the Internet is desirable in order to protect the providers' infrastructure, too. This, however, requires a resource-saving detection system, e. g. as proposed in [10], or special hardware. The latter is not considered here because of its higher costs and less flexibility. Thus, existing resource-consuming detection methods cannot be executed continuously. In addition, permanent parallel execution of multiple methods is no longer possible. Therefore, this work contributes an *adaptive and autonomous processing control* for in-network anomaly detection.

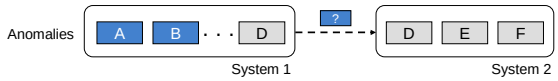

Fig. 1. Heterogeneous detection systems

Another challenge in today's Internet is the increasing *heterogeneity* of participating entities. This may prevent running certain anomaly detection methods on some detection systems while usable on other systems, e. g. due to resource constraints or differences in hardware. Furthermore, different domains may rely on different anomaly detection methods, e. g. due to policy decisions or their particular location. Figure 1 shows a scenario with two heterogeneous detection systems. The question is how to enable System 1 to communicate its knowledge and collaborate with System 2 in case it detected the anomalies A and B? In our opinion, the best solution is to first perform an identification of the according attack type and its characteristics. We therefore contribute a *generalized mechanism for the anomaly-based identification of attacks* in this paper. The identification also may simplify taking countermeasures, quickly interpreting the detection result, and visualization of the network state. Basis of such an identification is the set of locally detected anomalies. As shown in Figure 2, System 1 then is able to communicate that it detected attack 1. System 2 in turn can interpret the received information based on its own locally known set of anomalies.

In order to achieve both the identification as well as the autonomous processing control we, in addition, established a

generalized model of those entities that form anomaly-based attack detection—large-scale attacks, anomalies, and anomaly detection methods. This is necessary since lots of existing anomaly detection methods regard attack identification only vaguely [11] or not at all [12]. Other approaches assume that, in case a certain anomaly is detected, this anomaly directly correlates with a specific attack [6], [7]. This means that an implicit identification is performed. Lastly, approaches like [8], [9] use a more precise attack model but are limited to their respective anomalies. All these approaches are, in summary, mostly limited and inflexible. Furthermore, coherences between the various identified attacks are not regarded.

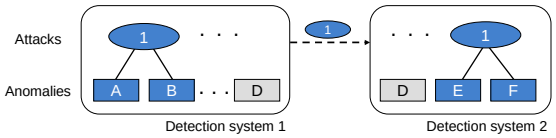

Fig. 2. Heterogeneous collaborative detection systems

This paper is structured as follows: In Section II the generalized model is introduced. Existing identification mechanisms are compared in Section III before our solution's architecture and the new processing control are presented. Section IV outlines the implementation and describes simulation results. The paper concludes with a summary and outlook in Section V.

A. Requirements

Especially in the area of Internet security, preconditions and active threats constantly change. This implicates that a mechanism for the identification of attacks has to provide **flexibility** and **extensibility** to be able to keep up with new developments in attacks as well as detection methods. Thus, easy integration of new anomalies, attacks, and detection methods into the modeling and identification has to be ensured. Since the identification system should be applicable within the network as well as at the edge of the Internet, **adaptability** is another important requirement. This also holds considering the heterogeneity of routers and detection systems previously described. Thus, an identification system has to be able to adapt to its particular environment and furthermore, to new requirements if necessary. A last requirement, we regard as important, is that **no manual interaction** is necessary as soon as the system has been set up and is running. This means that the processing control has to autonomously take decisions and be able to adapt to new situations.

II. Modeling of Large-Scale Attacks and Detection Entities

In this section, we establish a model of those entities that are important in the field of anomaly-based attack detection. To the authors' knowledge currently no such generalized model exists. Our model includes the entities **anomaly**, **large-scale attack**, and **anomaly detection method**. Then, we combine actual characteristics of the modeled attacks with our generalized model in order to create an *attack hierarchy*. This extensible and flexible approach provides the basis for the identification and processing control described in Section III.

At first we have a look at the **anomalies**. Our anomaly model is shown in Figure 3(a). The total set of anomalies can be split into *known anomalies* and *unknown anomalies*. Unknown anomalies are anomalies that have not been observed in the Internet yet. The *detectable anomalies* are a subset of the known ones. They can be classified into different categories, e. g. stochastic, distribution, or protocol anomalies. An example for a stochastic anomaly is a sudden increase in the number of observed packets. Distribution anomalies can be detected if address prefix or port number distributions are analyzed. If a protocol does not behave as expected, e. g. if the number of TCP SYN and SYN-ACK packets shows an imbalance, this is called a protocol anomaly. A subset of the detectable anomalies are the *detected anomalies* that can be used by an identification system, i. e. the anomaly detection system actually detected these anomalies.

The second model (see Figure 3(b)) refers to **large-scale attacks** in the Internet. The set of attacks can—similarly to the anomaly model—be split into *known attacks* and *unknown attacks*. The set of attacks that are *detectable by anomalies* contains known as well as unknown attacks since the latter may cause detectable anomalies, too. The *recordable attacks* form a subset of the intersection of known and detectable attacks and describe those attacks that actually can be identified by a detection system. Recordable attacks can be classified into different categories, e. g. DDoS attacks, worm propagations, or portscans. Although not being attacks at all, flash crowds and configuration errors are also included into the classification since they may also cause anomalies. Thus, these categories should be identifiable, too.

Our third model (see Figure 3(c)) focuses on **anomaly detection methods**. A detection method in its generalized form is described by the following four attributes:

- *Preconditions*—Meta information about the detection method, e. g. resource constraints, benefit or dependency of other detection methods.
- *Input data*—Information necessary for processing, which may e. g. originate of preceding methods or stored values.
- *Configuration data*—Externally specified parameters necessary for processing.
- *Output data*—Results of the detection method. This data is actually used as input for the identification mechanism.

Dependent on the fact if they have preconditions and need input data, detection methods can be further differentiated into *initial methods* and *conditional methods*. Conditional methods require certain input data whereas initial methods don't and thus, can be executed at any time.

Having modeled all involved entities in a generalized manner, in a second step we team these models with a description of actual attack characteristics as provided e. g. in [13]–[15]. This means, we assign a certain set of detected anomalies to a recordable attack. In doing so, we additionally distinguish between *required* and *optional* anomalies. A TCP flooding attack on a single victim, for example, is characterized by a stochastic anomaly of observed TCP packets and a distribution anomaly in destination address prefixes—significantly more

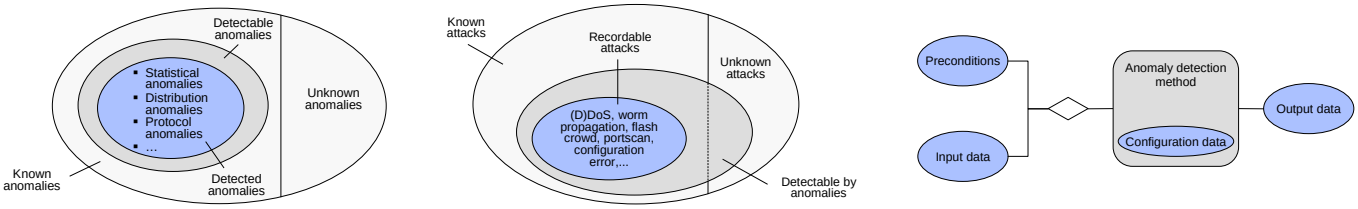

Fig. 3. Modeling of (a) anomalies, (b) large-scale attacks, and (c) anomaly detection methods

packets are sent to the victim's network than usual. These are required anomalies. An increase in ICMP destination unreachable messages, which often occurs in case of a successful DDoS attack, could be an optional anomaly.

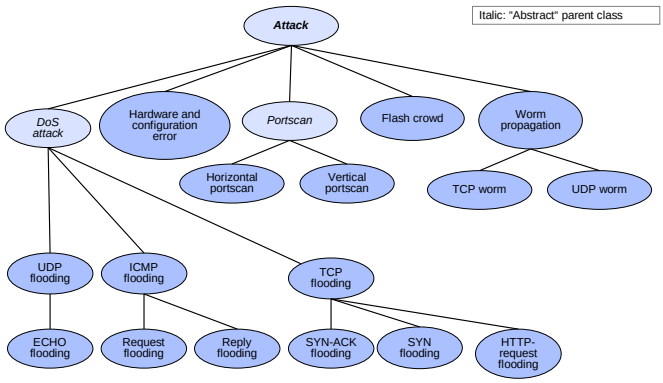

Fig. 4. Attack hierarchy

Based on all detectable anomalies and actually known attack characteristics, we defined various attack classes, which unite attacks closely related to each other. Each recordable attack is assigned to a certain attack class. The attacks of each attack class finally are included into an **attack hierarchy**. Figure 4 shows such a hierarchy, which is not complete but easily extensible and very flexible. In summary, the established attack hierarchy is more general and flexible than implicit identification and contains coherences between the different attack classes.

III. Anomaly-based Identification

Existing identification approaches are pointed out in Section III-A and their pros and cons regarding attack detection are discussed. Then, we introduce a generalized identification system with special respect to flexibility and accurate results in Section III-B before describing the autonomous detection processing control in Section III-C.

A. Related Work

Despite the many research activities that have been performed in the field of anomaly and attack detection, currently no generalized identification system exists, which is easily extensible and adaptable to heterogeneous environments. As discussed in Section I existing approaches for anomaly detection and identification are, to the best of our knowledge, limited or inflexible.

Existing approaches for identification of attacks are based on different mechanisms: [8] uses a rule-based mechanism, [7] hierarchical clustering, and [9] a classifier. Classifiers can be further differentiated into parametric and non-parametric classifiers. Parametric classifiers, e. g. a Bayes classifier, depend on a stochastic model that provides an a-priori probability distribution for the classification. Non-parametric classifiers like the Nearest Neighbor Classifier, on the other hand, must be configured with a classification model, from which the necessary probabilities are estimated. In case of clustering, the hierarchical clustering of [7] or K-Means-Clustering are just two exemplary mechanisms.

TABLE I
Comparison of different identification mechanisms

	Rule-based	Parametric classifier	Non-parametric classifier	Clustering
Result accuracy	good	acceptable	acceptable	poor
Flexibility	poor	good	acceptable	good
Meta data necessary	yes	no	yes	yes
Stochastic model necessary	no	yes	no	no

Table I shows a comparison of all the afore-mentioned identification mechanisms. Each mechanism is rated in different categories with respect to the suitability for a flexible and adaptable identification of large-scale attacks. A rule-based mechanism produces accurate results without need for learning data or an a-priori stochastic model. It, however, needs externally specified meta data—the rule set it works with. Establishing this meta data initially is a time-consuming task. Furthermore, this mechanism is very inflexible, i. e. poor results must be expected in case of imperfect anomaly detection input. Parametric classifiers require a predefined stochastic model of all concerned entities like regular Internet traffic, recordable attacks, detectable anomalies, and their coherences. Such a stochastic model, however, is not constructible due to the complexity of attack detection in the Internet. Clustering methods need a-priori learning data and a suitable clustering metric. Identification of the resulting clusters, however, still has to be performed manually. The last mechanism, a non-parametric classifier, does not require learning data or a stochastic model but some sort of meta data specifying already known categories. It is flexible and thus, is able to work with imperfect data since classification is performed by a *distance metric* and not by accurate comparison with existing categories. The results, however, are only estimated and thus, the result accuracy may vary.

In summary, parametric classifiers and clustering are not usable for a generalized attack identification due to the necessary

stochastic model. The other mechanisms, however, also have drawbacks: a non-parametric classifier does not necessarily produce accurate results and rule-based mechanisms are rather inflexible. Therefore, we decided to combine the advantages of the latter approaches while minimizing their disadvantages.

B. Architecture of our identification system

Figure 5 outlines the architecture of our identification system. A rule-based mechanism—that produces accurate results in case of perfect information—makes the first stage of the identification system. Due to its lack of flexibility the identification, however, is likely to fail in case of inaccurate or imperfect information. In this case, a non-parametric classifier is applied subsequently. Since the classifier is flexible enough to work with imperfect detection information, the probability of a successful identification is very high. Thus, the classifier reduces the drawbacks of the first stage but may return inaccurate results. Resource awareness of the whole identification is ensured by executing the classifier only if really required, i. e. in case the rule-based mechanism fails.

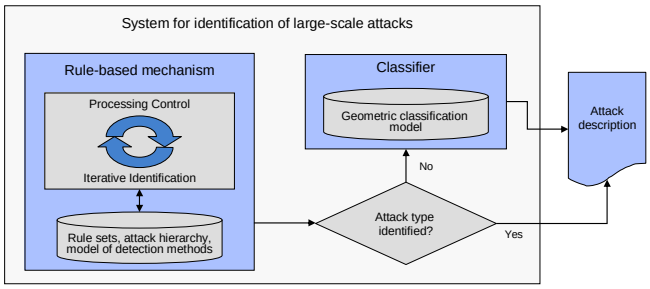

Fig. 5. Architecture of the identification system

Another advantage of our two-stage identification—besides the demand-driven execution of the classifier—is that the advantages of both mechanisms can be combined while configuration overhead remains manageable. The necessary meta data for both mechanisms is obtained from our generalized model and attack hierarchy established in Section II. The rule-based mechanism derives its meta data from the attack hierarchy: the rule sets. Each rule set specifies the required and optional anomalies of a particular attack. The location within the attack hierarchy and the relation to other attacks also is included into the specification. The classifier actually used is a nearest neighbor classifier in combination with a geometric classification model based on cuboids. The cuboids representing recordable attacks are also derived from the generalized model. They primarily are based on the output data of detection methods and anomalies associated to an attack.

In summary, the identification system requires a-priori meta data for both mechanisms. This data can be derived from our generalized model. Thus, the identification is easily extensible, very flexible and fits well into heterogeneous environments.

C. Autonomous processing control

The identification system has to be resource-saving and adaptable to its particular environment, especially if applied within the network. Therefore, we propose a processing control mechanism for the whole attack detection and identification system. Such a control mechanism only is suitable for attack detection if no manual interaction is required, i. e. it must be autonomous and adaptable to changing requirements.

Our autonomous processing control is integrated into the rule-based identification step. Depending on the modeled information about anomaly detection methods—preconditions, input, configuration, and output data—the processing control determines all initial methods. These can be executed all the time. If there are not enough resources to execute all initial methods, the processing control has to decide which methods actually to use. This decision can e. g. be based on preconditions like resource consumption, detection granularity, or benefit. An *identification process* is started as soon as an initial anomaly is detected. In this case, the conditional methods are checked for runnable methods with newly fulfilled preconditions or newly available input data. If multiple conditional methods are runnable, resource availability has to be considered before starting them and methods e. g. have to be executed consecutively. More sophisticated algorithms for the execution in case of low resources are imaginable but out of scope in this paper. This processing is performed iteratively until no additional methods become runnable.

The processing control ensures adaptability to available resources and dependencies between anomaly detection methods. In addition, it is possible e. g. to execute only anomaly detection methods that promise additional benefit. Specific methods that help in detecting certain worm characteristics, for example, can be disregarded if the detected anomalies indicate a DDoS attack. The necessary coherences for such a decision can be derived from the rule sets, which in turn are derived from the attack hierarchy of our generalized model. In summary, no manual interaction is required during detection and identification once the system has been set up. Establishing the generalized model and deriving the necessary meta data, however, still requires manual work.

IV. Implementation

This section presents the implementation as well as a simulative evaluation of the identification system. The simulations are based on the discrete event simulator OMNeT++ [16] and its Internet extension—the INET Framework [17].

Based on the generalized model and the attack hierarchy of Section II we first derived the rule sets for the rule-based mechanism and specified them using XML. Currently, our identification provides exemplary rule sets for most attacks of the subtree *DoS attack* and for the complete subtree *Worm propagation* of Figure 4. Furthermore, the four attributes modeling anomaly detection methods were specified based on XML for the currently used methods: a stochastic anomaly, an address distribution anomaly, and transport layer protocol anomalies [10]. This data is required by the processing control.

In order to actually implement the processing control and identification, we build on the attack detection framework *Distack* [18], which is easily extensible due to its modularity.

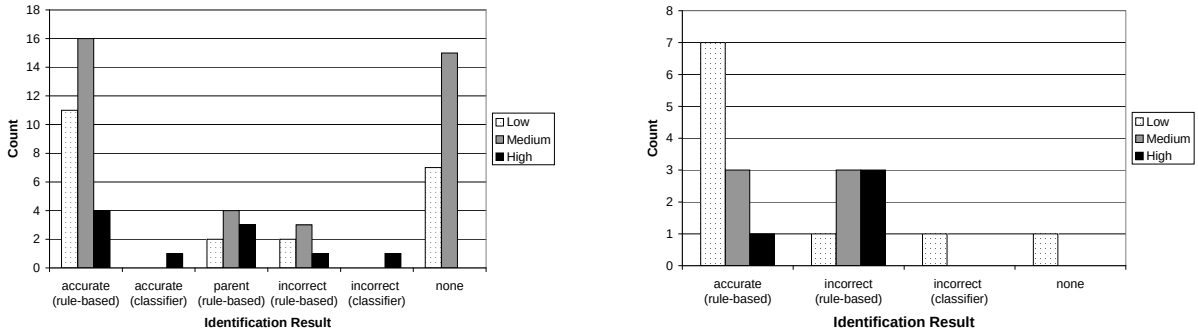

Fig. 6. Identification results of (a) DDoS and (b) worm propagation simulations

The afore-mentioned anomaly detection methods are already implemented for Distack. The processing control of Distack, however, is static and manually configured so far.

First, we implemented a new module called *coordinator*. This module reads the XML specifications, starts the initial methods, and implements the autonomous processing control. The second module we had to implement was the *geometric classifier*. In addition, some new messages for the internal communication between modules and new classes for management of the specification data and rule sets were added.

In case a detection process has ended, a rule-based identification is started. If this identification fails the classifier is activated. Therefore, the output data of all detected anomalies first have to be normalized to a fixed value interval—0 to 100 in our case. This is necessary to support heterogeneity of detection systems and anomaly detection methods. Then, this output data—representing a new cuboid in the n-dimensional classification space—is used as input for the classification process. Based on a specified nearest neighbor distance, e. g. the Euclidean distance, the nearest reference cuboid is calculated and the corresponding attack is returned as classification result. Unknown attacks can be identified by a calculated distance that exceeds a predefined threshold. In this case a manual classification and identification still is necessary. Finally, an attack description is generated (see Figure 5) that can be used e. g. for collaboration of distributed detection systems or initiating countermeasures.

A. Simulation

We decided to evaluate our identification system by means of simulations since testbeds—especially in case of large-scale attacks—are costly, mostly limited in size, and time-consuming to maintain. Furthermore, Ringberg et al. [19] recently pointed out today's lack and the need for simulation in case of anomaly-based attack detection. In addition, simulations are more suitable than real environments in some aspects due to their reproducibility and lack of side effects biasing the results.

We attached great importance on using simulation environments as realistic as possible by relying on *ReaSE* [20]. *ReaSE* is able to construct hierarchical Internet-like topologies and to automatically generate background traffic that shows self-similar behavior. In addition, generation of attack traffic relies on the mechanisms of real attack tools—the Tribe Flood Network in case of DDoS attacks and Code Red I in case of worm propagations.

For the simulative evaluation we generated a topology consisting of about 50 000 nodes in total and added a single anomaly detection and identification system. We conducted 35 simulations of DDoS attacks on a single victim host using 376 malicious zombie system while placing the detection system at the victim network's edge. 35 additional simulations of DDoS attacks used 79 zombies while placing the detection system at a router in the core network. A TCP SYN flooding attack was launched against port 80 of a webserver in 20 of these simulations. Since the webserver was able to accept only a limited number of connection attempts per port, its service finally was disrupted. These simulations thereby varied in attack volume, start time of the attack, and victim webserver. The further 50 simulations additionally varied the type of attack: flooding based on TCP RST, TCP SYN/ACK, UDP, or ICMP packets instead of TCP SYN packets. Finally, 20 simulations of a UDP worm propagation with varying probing volumes and locations of initially infected systems were conducted.

At simulation startup, the identification system starts the initial method of the detection system, which scans for a stochastic anomaly, i. e. a sudden increase in the observed traffic. As soon as a such an anomaly is detected the processing control iteratively checks for runnable conditional methods and starts them. Currently available conditional methods scan for IP address distribution and transport layer protocol anomalies. Having finished the iterative anomaly detection process, the rule-based identification is applied based on the detection result and the specified rule sets. The geometric classifier is only used on demand, i. e. if the rule-based identification fails.

Figure 6 organizes the simulation results into different categories: *accurate* means that exactly that attack was identified that actually was simulated. In case of *parent* a parent class of the accurate attack was identified according to our attack hierarchy (see Figure 4). If an identification resulted in a wrong attack type this is marked with *incorrect*. Finally, *none* means that no identification took place at all.

From the simulation results we learned that correctness and identification accuracy heavily depends on the actual anomaly detection result. In case of high-volume DDoS attacks (black) with attackers sending more than 75 packets/s each, the rule-

based stage of the identification returned an accurate or at least a correct result. The inaccurate results occured in case only a subset of the anomalies were detected and therefore, the rule-based system was only able to identify a parent class of the actual attack. The large number of accurately identified medium- (grey, 20–50 packets/s) and low-volume attacks (white, ≤ 15 packets/s) is caused mainly by ICMP and UDP attacks since background traffic of these aggregates is much lower than in case of TCP. Thus, correct detection is easier for such attacks even in case of medium or low volume. In case of TCP attacks, however, often no identification took place since no anomalies were detected by the detection system at all. Thus, no identification process was started. Lastly, the incorrectly identified DDoS attacks were caused by the fact that the anomaly detection resulted in false positive errors. In such situations the identification returned a correct attack type regarding the anomalies detected—regarding the network situation, however, the identified attack was wrong due to the false positives. In case of worm propagations frequent false positives occurred for medium- and high-volume attacks since probing traffic seemed to be directed to a single address prefix—which is typical for DDoS attacks—due to the simple probing mechanism used.

In only 3 simulations the rule-based mechanism failed to identify an attack. In these cases, the classifier was activated subsequently and identified at least one ongoing DDoS attack accurately. The low number of situations the classifier is started is caused by the fact that the detection system we built on provides only a small number of dimensions, i. e. a small number of anomaly detection methods and thus, a small number of output parameters. If the detection system, however, will be extended by additional detection methods in the future and the rule sets consider additional attack types, we are confident that the classifier will be necessary and compensate the problems of the inflexible rule-based mechanism.

V. Conclusion and Outlook

This paper presented an *identification* system for large-scale attacks like DDoS attacks or worm propagations. Basis of the identification are the locally detected anomalies. Such an identification of attack type and characteristics facilitates collaboration of distributed detection systems in heterogeneous environments and may simplify other tasks like taking countermeasures. In addition, the proposed *autonomous processing control* of detection and identification ensures adaptability to available resources and anomaly detection methods as well as to changing attack situations. In order to ensure flexibility and extensibility to future requirements and detection methods, both processing control and identification build on a *generalized modeling* of the entities participating in attack detection. The proposed anomaly-based attack identification consists of a rule-based mechanism and a subsequent demand-driven geometric classifier. The simulative evaluation showed that the identification is able to return accurate results but to a certain degree depends on the correctness and accuracy of the preceding anomaly detection.

Future work should examine if it is possible to extend the classification by self-learning methods that autonomously improve the reference cuboids of the classifier and adapt the classification to its actual environment. This could be a promising approach primarily to handle unknown attacks. Furthermore, an evaluation and comparison of the identification in terms of efficiency and accuracy should be performed. Finally, integration of further data, e. g. provided by defense-in-depth solutions, into the identification process should be considered in order to improve identification accuracy.

Acknowledgment

I would like to thank especially Jannis Breitwieser for his excellent work during his diploma thesis. His work provided an important and valuable contribution to this paper.

References

[1] Amazon.com, "Amazon.com announces fourth quarter sales 2008," http://phx.corporate-ir.net/phoenix.zhtml?c=97664&p=irol-newsArticle&ID=1250070&highlight=, Jan. 2009.

[2] Arbor Networks, "Worldwide Infrastructure Security Report," http://www.arbornetworks.com/report, Oct. 2008.

[3] D. Dittrich, "The Tribe Flood Network DDoS attack tool," http://staff.washington.edu/dittrich/misc/tfn.analysis, Oct. 1999.

[4] G Data AG, "Das grosse Geschaeft mit dem E-Muell," Oct. 2007, available in German only at http://www.gdata.de/unternehmen/DE/articleview/3920/1/160/.

[5] M. Roesch, "Snort," http://www.snort.org, 2001.

[6] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," in *Proc. of ACM SIGCOMM*, Aug. 2004, pp. 219–230.

[7] J. Wang, D. J. Miller, and G. Kesidis, "Efficient mining of the multidimensional traffic cluster hierarchy for digesting, visualization, and anomaly identification," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1929–1941, Oct. 2006.

[8] L. T. Quyen, M. Zhanikeev, and Y. Tanaka, "Anomaly identification based on flow analysis," in *Proc. of TENCON*, Nov. 2006, pp. 1–4.

[9] W. Wang and R. Battiti, "Identifying intrusions in computer networks with principal component analysis," in *Proc. of 1st ARES*, 2006, pp. 270–279.

[10] T. Gamer, M. Schöller, and R. Bless, "An extensible and flexible System for Network Anomaly Detection," in *Proc. of Autonomic Networking*, Sep. 2006, pp. 97–108.

[11] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-Service Attack-Detection Techniques," *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, Jan. 2006.

[12] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: a statistical anomaly approach," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 76–82, Oct. 2002.

[13] C. Douligeris and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art," *Computer Networks*, vol. 44, no. 5, pp. 643–666, Apr. 2004.

[14] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," *Proc. of ACM SIGCOMM*, pp. 99–110, Aug. 2003.

[15] H. Wang, D. Zhang, and K. Shin, "Detecting syn flooding attacks," in *Proc. of IEEE Infocom*, Jun. 2002, pp. 1530–1539.

[16] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *Proc. of the European Simulation Multiconference*, Jun. 2001, pp. 319–324.

[17] A. Varga, "INET Framework," http://www.omnetpp.org/pmwiki/index.php?n=Main.INETFramework, Sep. 2007.

[18] T. Gamer, C. P. Mayer, and M. Zitterbart, "Distack—A Framework for Anomaly-based Large-scale Attack Detection," in *Proc. of 2nd SECURWARE*, Aug. 2008, pp. 34–40.

[19] H. Ringberg, M. Roughan, and J. Rexford, "The Need for Simulation in Evaluating Anomaly Detectors," *SIGCOMM Computer Communication Review*, vol. 38, no. 1, pp. 55–59, Jan. 2008.

[20] T. Gamer and M. Scharf, "Realistic Simulation Environments for IP-based Networks," in *Dig. Proc. of the OMNeT++ Workshop*, Mar. 2008.