A Granularity-adaptive System for in-Network Attack Detection

Thomas Gamer, Marcus Schöller, and Roland Bless
Institut für Telematik, Universität Karlsruhe (TH)
Germany

***Abstract*— The early detection of uprising DDoS attacks and worm propagations is still a challenge for today's network operators. An effective attack mitigation makes the detection of such network hazards close to its sources necessary. We therefore propose to use an in-network attack detection system which can be installed on routers. In high-speed networks a detailed per-packet analysis on a link's aggregated traffic, however, is infeasible without special-purpose hardware which causes additional costs. Our design addresses this issue by adapting detection granularity and analysis effort to the current stage of the attack detection. In this paper we introduce such a granularity-adaptive attack detection system.**

I. Introduction

Automatic detection of various kinds of hazards which, appear more frequently in today's networks, is still a major challenge for network operators. A major threatening type of such hazards are distributed denial-of-service (DDoS) attacks like the ones against Yahoo, CNN, eBay, the million dollar homepage, and many more which recently have attracted public attention. With DDoS flooding attacks [1] the attacker does not exploit a weakness of the victim's operating system or application but aims to overload resources like link capacity or memory by flooding the system with more traffic than it can process. The attack traffic is generated by many slave systems that the attacker has compromised before.

Another threat to the Internet today are worms [2]. A worm automatically exploits security holes in operating systems or applications to infiltrate a system. After a successful break-in the worm starts to propagate itself to as many other systems as possible. One side effect of this propagation is the increasing bandwidth consumption since more and more worm instances try to propagate themselves to other systems.

The earlier such attacks can be detected the better the network can be protected against them. This requires a fast reacting detection system within the network. The detection system has to apply realtime traffic analysis on the traffic to be able to detect DDoS attacks and worm propagations. In this paper, the notion *packet stream* designates a link's total aggregated traffic whereas a set of packets with same characteristics, e.g., all TCP packets, is referred to as an *aggregate*.

Applying traffic analysis on a packet stream causes several problems in high-speed networks: Due to the high bandwidth of backbone links an inspection of all packets is infeasible without affecting a router's forwarding performance even with today's router hardware. One approach is to use a packet selection mechanism to reduce the number of packets that have to be inspected by the detection system. The IETF working Group PSAMP [4] proposes various packet selectors for the Internet, especially within the background of traffic measurement. In [3] we already investigated the suitability of different packet selectors in regard to an anomaly based detection system. In order to be able to cope with the data rate in high-speed networks anomaly detection systems have to use *sampling* mechanisms. A sampling mechanism effectively reduces the number of packets that are inspected, but it also introduces estimation errors. Thus, the parameters of the applied sampling mechanism have to be chosen in such a way that the error caused by packet selection is restricted to a predefined tolerance level. Therefore, a tradeoff has to be found between traffic analysis scalability and the estimation error caused by sampling.

Another problem with traffic analysis on a packet stream in high-speed networks is that not only packet inspection but *deep* packet inspection is needed to reliably detect DDoS attacks and worm propagations, i.e., information from higher layer packet headers above the network layer are needed, too. This is only possible if the number of to be deeply inspected packets can be reduced to a feasible level that is much lower than for simple packet inspection. In case of a system for anomaly detection, however, the error caused by packet selection has to be restricted to a predefined tolerance level. Thus, the number of packets selected by a packet selector depends on the predefined tolerance level and cannot be additionally adapted to a feasible level for deep packet inspection. Therefore, deep packet inspection without additional hardware also is infeasible on backbone links without affecting a router's forwarding performance due to the high link bandwidth.

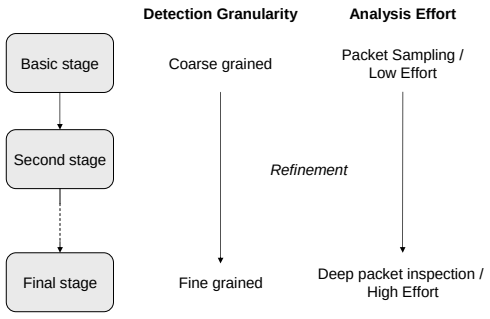

Fig. 1. Architecture of a hierarchical detection system using refinement

Our approach to solve this problem is to build a hierarchical detection system which uses *refinement*, i.e., detection granularity is increased with each subsequently loaded detection stage (see figure 1). Thus, the basic stage of the detection system performs just a coarse grained detection that only scans for indications of an attack by using low analysis effort. Further stages are loaded whenever an attack is assumed in the basic stage. These further stages analyze only a part of the whole packet stream due to the information about the assumed attack gathered by the basic stage. Therefore, the further stages are able to do a more fine grained hazard detection by applying deeper packet inspection on the reduced packet stream. Thus, the detection system gathers more detailed information about the attack in each of the further stages by using a higher analysis effort.

This paper details on such a granularity increasing system for attack detection and is organized as follows: In section II we detail on packet selectors suitable for an hazard detection system and show the dependency of sampling parameters on the observed bandwidth. Section III presents the architecture of the hierarchical detection system and special characteristics of the system. Additionally, a short example of a concrete anomaly-based attack detection system is proviced. Finally, section IV gives a short summary.

A. Related Work

There are some existing approaches that design a DDoS attack detection system. [7] uses network processors to perform a deep packet inspection of all observed packets in a backbone network. Similar, [10] uses special purpose hardware to timestamp packets and do the analysis offline afterwards.

Another approach, the pushback mechanism [6], is activated as soon as congestion occurs on a router and only dropped packets are inspected. Sterne et al. [5] detects stochastic anomalies by using a simple threshold based DDoS detection mechanism on active networking nodes, but no further refinement is done if an attack has been detected. Bro [8] is an open source network intrusion detection system that works with refinement. But – unlike our approach – the refinement has a different scope. Bro is an event-driven approach and consists of three parts: the packet capture, the policy-neutral event engine, and the policy layer. A problem of this approach is that Bro creates lots of state by deep packet inspection and semantic analysis. Finally, the MVP architecture of Cisco Systems [9] also uses refinement for detection of DDoS attacks but this refinement is not very flexible and is only done in two steps, i.e., multiple stages are not possible for refinement.

II. Packet selectors

The IETF PSAMP working group defined two types of packet selectors: filtering and sampling [4]. *Filtering* is used if only a particular subset of packets is of interest. In [3] we already examined which packet selectors are suitable for an attack detection system and decided to use the sampling method called *systematic count based sampling*.

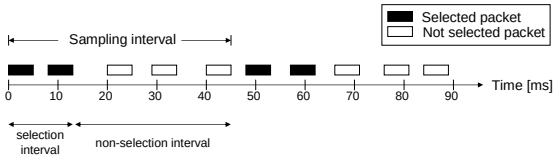

Fig. 2. Example of packet selection with systematic count based sampling

This sampling method is deterministic but independent of packet content and router state. For this method a *sampling interval* is defined consisting of a *selection interval* and a *non-selection interval*. A periodic trigger defines the beginning of a sampling interval. The unit of the intervals is *count based*. An example of this sampling method with a sampling interval of 5 packets, a selection interval of 2 packets, and a non-selection interval of 3 packets is shown in figure 2. Even though systematic count based sampling is not a random sampling method, it is said that the example sampling method has a sampling probability of 40 %.

TABLE I
Relative deviations of systematic count based sampling

	Average rate [packets / interval]	Sampling parameters	Relative Deviations TCP	UDP	ICMP
Interval length: 6 seconds					
A	125 000	4 / 100 (4 %)	0.67 %	6.18 %	14.62 %
B	102 000	6 / 100 (6 %)	0.24 %	2.36 %	13.15 %
Interval length: 0.6 seconds					
A	12 500	3 / 10 (30 %)	0.47 %	4.18 %	14.75 %
B	10 200	7 / 20 (35 %)	0.55 %	4.53 %	13.40 %

In [3] we calculated a relative deviation between a sampling run and the original packet trace to get the estimation accuracy of a packet selector. This relative deviation is also used in the following examination. We applied systematic count based sampling to some network traces [11] and assumed a predefined tolerance level for the smallest aggregate of 15 % relative deviation. In order not to exceed this tolerance level the sampling probability must be increased if the average number of packets per interval – the *interval bandwidth* – of an observed packet stream is reduced. Our examination shows, however, that in this case the absolute number of selected packets per interval gets smaller. Table I lists the relative deviations of the aggregates TCP packets, UDP packets, and ICMP packets for two different packet traces *A* and *B*. The first two data rows used an interval length of 6 seconds which corresponds to an interval bandwidth of about 100 – 125 k packets per interval. We can clearly see that our predefined tolerance level is not exceeded for any of the given aggregates. In case of trace *A* a sampling probability of 4 % is used. The next two data rows used an interval length of 0.6 seconds which corresponds to an interval bandwidth of about 10 – 12.5 k packets per interval. We alternatively could have used a network trace with a lower packet rate of about 10 – 12 k packets in a 6 seconds interval instead of reducing the interval length of the same network trace. In case of trace *A* a sampling

probability of 30 % is used to avoid an exceeding of the tolerance level. We now see that, though the sampling probability did increase depending on the lower interval bandwidth, the absolute number of selected packets per interval gets smaller.

III. Architecture

We developed an anomaly-based detection system for network hazards that is hierarchical and extensible. Extensibility of the system means that new anomalies can be introduced to different stages of the detection system easily. This ensures that the system can be adapted to different network scenarios.

The system is designed hierarchically (see figure 1) to incrementally increase inspection depth. Therefore, the basic stage of our detection system analyzes the packet distribution within specific aggregates and scans for indications of an attack by detecting stochastic anomalies. A stochastic anomaly is a rapid increase of packets observed in a specific aggregate. Then for each predefined aggregate the number of packets that belong to this aggregate is counted in every interval. An indication of an ongoing attack is found if the observed number of packets exceeds a predefined *packet threshold* of the aggregate.

Such a dynamic packet threshold is calculated representing the average packet count in an aggregate for the last couple of intervals to make the system self-adaptable to network load changes. To prevent the system from generating too many false positive indications and starting the next stages for deeper inspections unnecessarily an *interval threshold* is defined. This interval threshold is necessary due to the self-similarity of Internet traffic [12] which can cause normal traffic to exceed the packet threshold even though no attack is currently going on. Therefore, an indication only is generated if the packet threshold is exceeded in more consecutive intervals than the interval threshold defines.

Since the basic stage only has to classify which aggregate a packet belongs to, only little information from the network header is needed. Furthermore, checking stochastic anomalies and adapting aggregate thresholds at the end of each interval requires only some simple calculations. Thus, in conjunction with the usage of a packet selector the basic stage needs only a small amount of resources and does not need deep packet inspection of selected packets. This leads to a coarse grained attack detection which only can generate hints on attacks but nevertheless is able to reduce the packet stream that has to be analyzed in further stages.

After detecting an indication of an attack by a stochastic anomaly a second stage is loaded. This stage – and all subsequent loaded stages, respectively – has two possibilities for refinement of detection granularity:

- Performing a deeper packet inspection of those packets that preceding stages considered suspicious, or
- analyzing data collected during the preceding stage.

Inspecting only those packets deeper that the preceding stages considered suspicious is motivated by the fact that this suspicious traffic is usually only a fraction of the packet stream observed in the preceding stage. To restrict the error caused by packet selection to the same predefined tolerance level as in the preceding stage, the sampling probability of the packet selector has to be increased due to the lower bandwidth of the suspicious traffic. Nevertheless, a smaller total number of packets has to be selected than in the preceding stage (see subsection II) and therefore, deeper packet inspection is feasible without affecting a router's forwarding performance if the packet rate of the suspicious traffic is small. In case that the difference of the selected number of packets in the current and the preceding stage is just marginal, a negative impact on a router's forwarding performance is possible if deeper packet inspection is applied in the current stage.

The other possibility for refinement of detection granularity is to proactively collect data in a preceding stage. This is necessary for example if the current stage needs history data of previous intervals that can not be easily collected by the current stage itself. The analysis of the collected data can then be done in the current stage based on the data from the preceding stage. The big advantage of this proactive approach is the separation of data collection and data analysis, i.e., calculations or scanning for further anomalies based on the already collected data is performed not until a subsequent stage is loaded and therefore, only causes additional *computational overhead* if it is really necessary. A drawback of this approach is that additional memory is needed to store data for a subsequent stage.

A. Attack detection in a small provider network

One example for the usage of the hierarchical attack detection system described above are high-speed networks. Another example are small provider networks that we detail on in this section. In small provider networks the focus of the detection system lies on detection of DDoS attacks and worm propagations of new worms which are not well-known yet – and therefore cannot be detected by a signature-based detection system. Our detection system uses different kinds of anomalies to detect ongoing DDoS attacks and worm propagations, e.g., stochastic anomalies, distribution anomalies or protocol anomalies. All these anomalies give hints to an ongoing attack. Figure 3 shows the architecture of our system that could be deployed in small provider networks and will be explained in the following. The same architecture could be used in high-speed networks, too.

The functionality of the basic stage was already described in the previous section. In summary, the basic stage analyzes the packet distribution of predefined aggregates and detects *stochastic anomalies* by using packet thresholds. If the packet threshold of an aggregate is exceeded in more consecutive intervals than the given interval threshold an indication of an attack is generated and the second stage is loaded.

Our second stage analyzes additional data collected proactively by the basic stage to refine detection granularity. The detection system uses a *distribution anomaly* to distinguish DDoS attacks from worm propagations. This can be achieved by analyzing the distribution of packets into subnet prefixes based on destination addresses. Therefore, the whole address

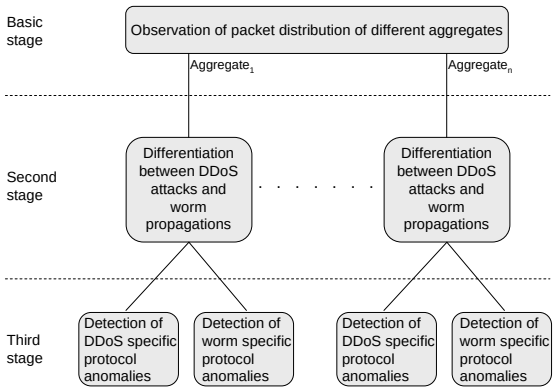

Fig. 3. System architecture for attack detection in small provider networks

space is divided into subnet prefixes based on the routing table of the node executing the detection system. If large parts of the suspicious traffic – the number of packets by which the packet threshold was exceeded – are sent into exactly one subnet a DDoS attack is indicated since only one victim is currently attacked. If the suspicious traffic is equally distributed to all existing subnets a worm propagation is assumed since worms spread more or less evenly distributed all over the Internet. Thus, the second stage of the detection system gains more information about the ongoing attack on basis of the data collected by the preceding stage.

Having done all calculations on the data collected in the basic stage a third stage is loaded. This third stage uses deeper packet inspection for refinement of detection granularity. The suspicious traffic is reduced by the information of the aggregate in which an indication for an attack was found in the basic stage and by the information of the attacked subnet in case of DDoS attacks derived by the second stage. Since the suspicious traffic observed in this third stage is reduced the sampling probability has to be increased but nevertheless the total number of selected packets can be reduced in most cases according to section II. The third stage scans for attack specific *protocol anomalies* to identify either DDoS attacks or worm propagations in more detail.

The detection of DDoS specific protocol anomalies is based on the fact that most of the existing DDoS attacks lead to a breach of symmetry between incoming and outgoing sub-aggregates which belong together by protocol definition. Thus, a SYN flooding attack for example can be detected by an increasing asymmetry of the sub-aggregates *incoming TCP packets with SYN flag set* and *outgoing TCP packets with SYN and ACK flag set* which arises if the victim's TCP instance is already down. Such a deeper packet inspection reveals more information about the ongoing DDoS attack. Such asymmetries as the one just described, however, can be caused by routing asymmetries, too. This must be considered when designing an anomaly-based detection system.

An example for a protocol anomaly that can be used for detection of a worm propagation utilizes the fact that a worm tries to infect other hosts usually rather randomly. A worm propagation, therefore, sends packets to randomly selected hosts, but if the system or network does not exist at all, an ICMP message "host/network unreachable" is generated. Thus the ratio of ICMP packets with this error message will increase during a worm propagation. Here, too, a deeper packet inspection is done to gather more information about the ongoing worm propagation.

In our detection system the third stage also is the final stage but one can think of using more stages to get even more detailed information about an ongoing attack, e.g., detecting application specific anomalies in a fourth stage, and thus, doing a even better refinement of detection granularity.

IV. Conclusion

In this paper we presented a system for in-network attack detection which is hierarchical, anomaly-based, and extensible. Additionally, the hierarchical character of the system enables a granularity-adaptive detection system that uses refinement. We showed that adapting the detection granularity and analysis effort to the analyzed packet stream ensures that the system can be deployed in high-speed networks without affecting a router's forwarding performance and without the need for additional special-purpose hardware.

Future research has to address an adaptive sampling mechanism that adaptively chooses suitable sampling parameters based on the bandwidth of the analyzed packet stream. This is necessary to ensure a limitation of the estimation error caused by packet sampling to a predefined tolerance level.

References

[1] A. Hussain, J. Heidemann, and C. Papadopoulos, *A framework for classifying denial of service attacks-extended*, Technical Report ISI-TR-2003-569b, USC/Information Sciences Institute, June 2003.

[2] C. Shannon and D. Moore, *The spread of the witty worm*, IEEE Security and Privacy, 2(4), 2004.

[3] M. Schöller, T. Gamer, R. Bless, and M. Zitterbart, *An Extension to Packet Filtering of Programmable Networks*, In Proc. of 7th Annual Int. Working Conf. on Active Networking (IWAN), Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Nov 2005.

[4] N. G. Duffield, *A framework for packet selection and reporting*, Internet Draft, draft-ietfpsamp-framework-10.txt, Work in Progress, Internet Engineering Task Force, January 2005.

[5] D. Sterne, K. Djahandari, R. Balupari, W. L. Cholter, B. Babson, B. Wilson, P. Narasimhan, and A. Purtell, *Active network based ddos defense*, Proc. of DANCE, 2002.

[6] J. Ioannidis and S. M. Bellovin, *Implementing pushback: Router-based defense against DDoS attacks*, In Proceedings of Network and Distributed System Security Symposium, Catamaran Resort Hotel San Diego, California 6-8 February 2002, 1775Wiehle Ave., Suite 102, Reston, VA 20190, February 2002. The Internet Society.

[7] L. Ruf, A. Wagner, K. Farkas, and B. Plattner, *A Detection And Filter System for Use Against Large-Scale DDoS Attacks In the Internet-Backbone*, In Proc. of 6th Annual Int. Working Conf. on Active Networking (IWAN), Lawrence Kansas, USA, Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Oct. 2004.

[8] Vern Paxson, *Bro: A System for Detecting Networks Intruders in Real-Time*, Computer Networks, 31 (23-24), 1999.

[9] Cisco Systems, *Defeating DDoS attacks*, White Paper. 2005

[10] D. Sass and S. Junghans, *I^2MP – An architecture for hardware supported high-precision traffic measurement*, Proceedings of the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems. 2006.

[11] NLANR Measurement and Network Analyis Group. http://pma.nlanr.net.

[12] K. Park andW.Willinger, *Self-similar network traffic: An overview*, In Self-Similar Network Traffic and Performance Evaluation. Wiley Interscience, 1999.