Overload-resistant Network Traffic Analysis

Marek Švepeš[1], Tomáš Čejka[2]
[2]CESNET, a.l.e. [1]FIT, CTU in Prague
Zikova 4, 160 00 Prague 6 Thakurova 9, 160 00 Prague 6
Czech Republic Czech Republic

cejkat@cesnet.cz, svepemar@fit.cvut.cz

Abstract. Flow-based monitoring is currently a leading approach of network security analysis. A flow record is an aggregated information about network traffic. Since various network attacks use just a few packets per flow, the advantage of aggregation is seriously limited. As a side effect, monitoring infrastructure and analysis system are affected. This paper proposes an overload-resistant architecture of the detection system that would overcome high load of flow records in time of attack.

Keywords.

1 Introduction

Monitoring computer networks is an important task for every network operator. Monitoring systems can provide valuable information about network traffic for various purposes such as accounting, performance measurement or network security. Since the volume of network traffic is high and it is still growing, it is very complicated to perform packet-based monitoring and analysis in large networks. Therefore, approaches based on analyzing flows, which are aggregated packets, are mostly used.

With focus on network security, the analysis can be done by detection systems which try to find an abnormal behaviour in the network traffic. There are lots of various types of such traffic, however, the most dangerous are DoS or DDoS attacks that can affect operability of a network infrastructure by depleting bandwidth or hardware resources. As a side effect of such attacks, monitoring and detection systems are vulnerable too especially when they are not designed with oversized resources.

Since the DDoS attacks are based on generating lots of connections from many sources, flow-based systems observe increased number of flows. That goes against the idea of flow records aggregating packets of each connection into lower number of records with total statistics about the connection.

A typical infrastructure needed for this analysis consists of several components: i) monitoring probes which export flows, ii) collector for storing or preprocessing received flows, iii) detection system. Fig. 1 shows the described infrastructure where the NEMEA system is used for detection. NEMEA is being developed by CESNET and it is described in more detail in Sec. 2.

Even though this paper shows a work in progress and a design of the future work, our aim is to propose an improved architecture of the NEMEA system. The new version of NEMEA should be overload-resistant, i.e. it should be able to perform additional aggregation of flow records in case of a huge attack. Additionally, to handle higher numbers of flow records, the system should be horizontally scalable. The goals of this work can be summarized as follows:

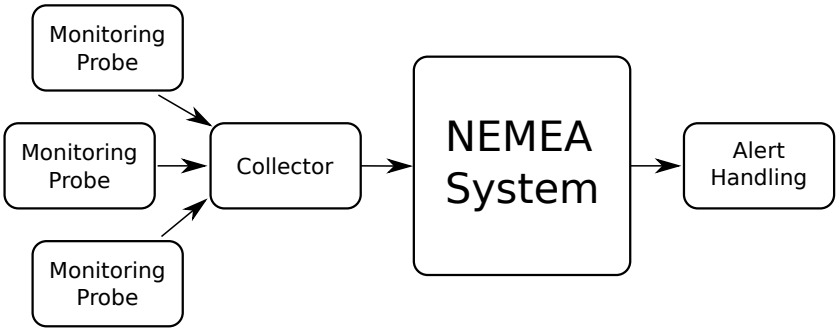

Figure 1: Monitoring infrastructure: i) monitoring probes which export flows, ii) collector for storing or preprocessing received flows, iii) detection system (NEMEA system) for network traffic analysis.

- to detect possible overload and minimize it using additional aggregation or bypassing flow records that belong to the attack,
- to overcome resource limits of a single machine running the detection system using a distributed environment with multiple machines,
- to manage and monitor instances of distributed NEMEA system.

2 NEMEA System

NEMEA system is a stream-wise, flow-based and modular detection system for network traffic analysis. It consists of many independent modules which are interconnected via communication interfaces and each of the modules has its own task. Communication between modules is done by message passing where the messages contain flow records, alerts, some statistics or preprocessed data. Fig. 2 shows a connection of several modules with the following tasks: anomaly detection, alerts aggregation, logging and alerts reporting. There is also one special module called Supervisor. It is a management module for monitoring and configuration of the whole system.

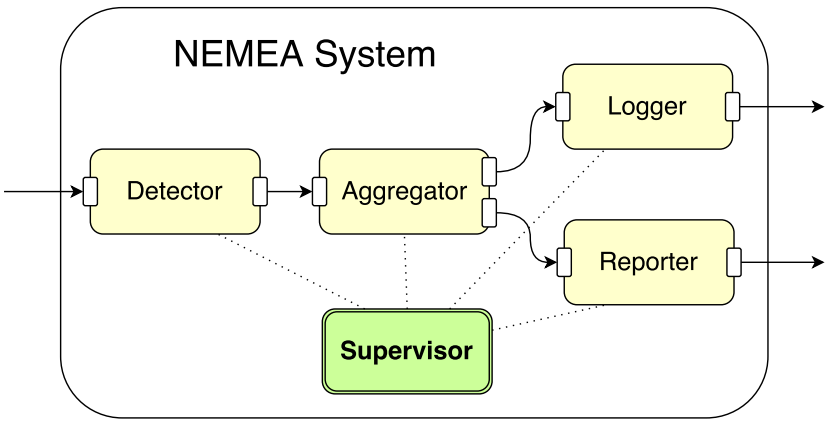

Figure 2: Modular detection system NEMEA for network traffic analysis. Consists of many independent interconnected modules and one special control module called Supervisor.

3 Management system Supervisor

Supervisor loads a configuration file where the modules are specified and it controls modules of the NEMEA system. Supervisor runs as a system daemon and there is a thin client that allows for manipulation with the daemon.

Main functions provided by supervisor:

- *enable* or *disable* a module or groups of modules,
- show *status* of the system (status of every module),
- show *loaded configuration* in detail (all information about modules, e.g. parameters, communication interfaces etc.),
- *reload configuration* which updates running configuration on runtime,
- browse *log files* of supervisor and modules.

The *reload configuration* function is important because of its three basic use-cases: i) adding, ii) removing and iii) modifying a module in the running configuration.

Monitoring of the modules:

- module status (can be *running* or *stopped*),
- *statistics* of the communication interfaces of each module (e.g. number of sent messages),
- CPU and memory usage of each module.

These statistics are used by the Munin [**?**] plugin for long term graphs and also by a simple tool which shows the statistics of modules at real-time. Fig. 3 shows an example of graph created by Munin and Fig. 4 shows real-time statistics of several modules.

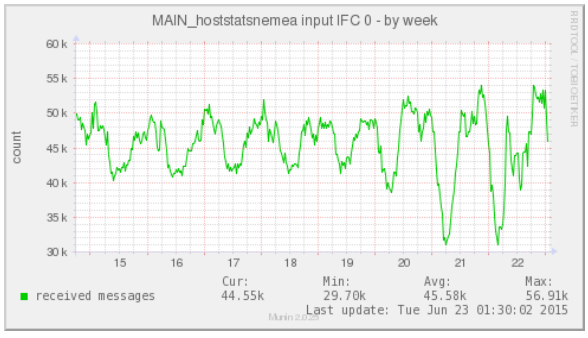

Figure 3: Long-term graph generated using statistics about NEMEA module interfaces. It shows number of received messages by one input interface.

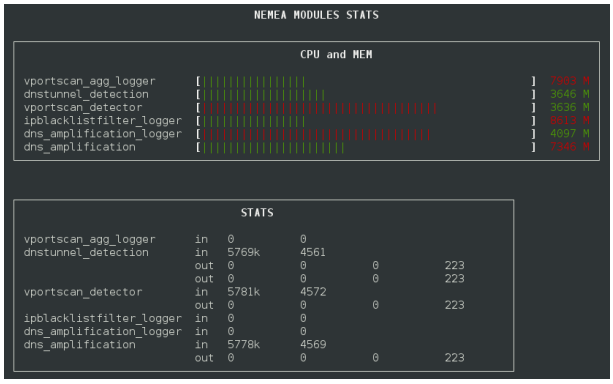

Figure 4: Tool showing statistics about NEMEA modules at real-time. The statistics are fetched from modules by Supervisor.

4 Overload-resistant NEMEA

Network Measurements Analysis (NEMEA) is a modular system and its modules can be easily distributed on separated nodes, i.e. computing machines. Modules running on different nodes can communicate with each other over the computer network using a TCP communication interface supported by NEMEA. However, a non-trivial issue is a division of one stream of flow records from the collector between many nodes without any disruption of the detection results.

For many detection mechanisms, it is crucially important to analyze all related flow records in order to detect an attack. If flow records of an attack were distributed on different computing nodes, the detection would be slower or, in the worst case, the attack would remain undetected at all.

Fig. 5 shows our proposed solution, where a stream of flow records is distributed to groups of nodes with respect to characteristics that are sensitive for the specific detection methods. For instance, many detection methods need to see all flow records of the same IP address. A special scatter function is needed to choose which node is going to process a flow record.

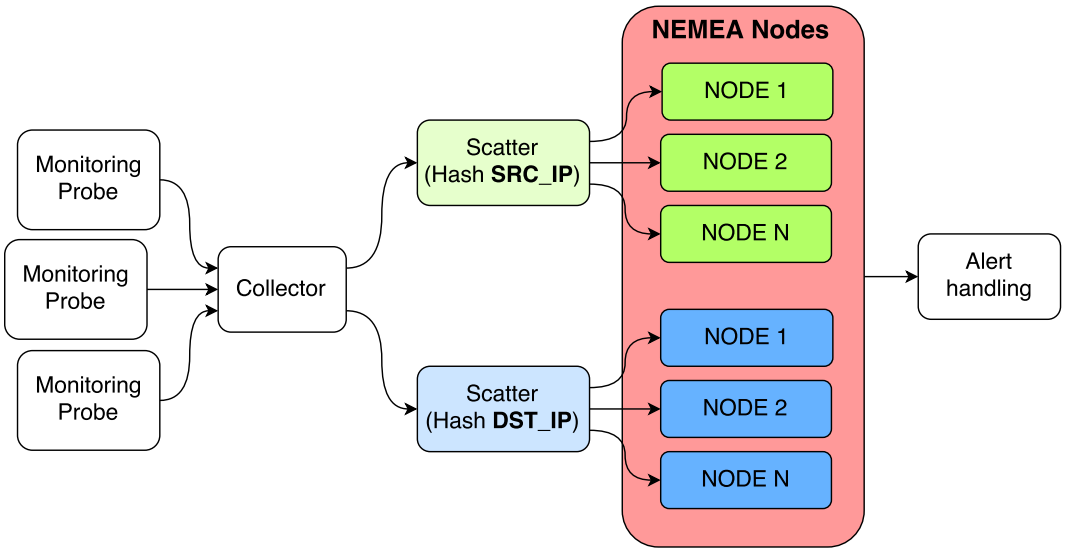

Figure 5: Horizontal scalable NEMEA system - monitoring infrastructure (Fig. 1) extended by many NEMEA nodes. Stream of flows is divided between nodes using flow IP address hashing. To not disrupt the detection methods on each node, stream of flows has to be duplicated and once the source IP is hashed and once the destination IP.

Every node has its own NEMEA system running but receives only part of the flow records (in ideal situation *1/N* where N is number of nodes). To decide which node receives flow record x, there is a scatter hashing IP address from the flow record and mapping it on the nodes. There are some detection methods based on source IP address and some methods based on destination IP address. It means detectors based on source IP address must receive all flow records with the same source IP and vice versa. That's why there are two scatters (each with group of nodes), one for source IP and one for destination IP. The flow records stream has to be duplicated.

Benefits of the horizontal scalability

- The proposed architecture overcomes resource limits of one machine, thus the system in total can process bigger amount of data,

- load of each machine is decreased due to distribution of data, thus the hardware resources of each node can be lower.

Cons of this solution

- The data is duplicated, the same flow records are processed twice — waste of resources can be mitigated if the detectors on green nodes are disjoint with the detectors on the blue nodes (in Fig. 5).

The scalability improves amount of processed data but it cannot protect individual nodes against overloading. For example, all flows from DDoS attack targeted to one destination IP address would be processed by one blue node.

Fig. 6 shows an extension of NEMEA system running on each node. There is one extra module with storage that receives all flow records and resends them to other modules. In addition when high number of flow records belonging to an attack is received, the module can start Filtering, additional Aggregating and Deduplication (the module is called FAD).

Such a huge attack that would cause the system overload should be detected quickly. Once the attack is detected by some detector, an alert is sent to Supervisor and it adds a rule into FAD. Immediately after adding the rule, FAD module can start additional aggregation/filtering of the flows or it can throw away the flows associated with the attack or they can be stored for later analysis. Anyway, this can limit the impact of the attack, because the rest of modules (especially the detectors) on the same node won't have to process all the data.

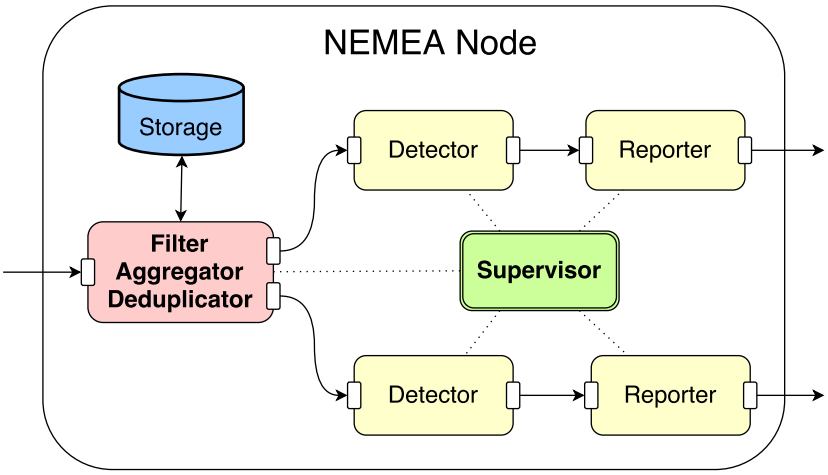

Figure 6: Overload-resistant NEMEA system is extended by one module with extra storage. Its function is filtering, additional aggregating and deduplicating (FAD).

5 Conclusion

Handling huge network traffic is a non-trivial task, meanwhile, the volume of data is still growing. For large networks, it is a common practice to use aggregated information for network traffic analysis and even for malicious traffic detection. However, some attacks, by their nature, generate almost as many flow records as original packets. This can lead to overload of not only the network infrastructure but also the detection system.

In order to deal with the proposed issue, we are proposing a design of improved architecture for the open-source modular system NEMEA. Even though this paper presents just a work in

progress, the resulting architecture and the enhanced system might handle higher load using a horizontal scaling and so called overload-resistant feature.

Acknowledgments

This work was partially supported by the "CESNET E-Infrastructure" (LM2015042) and CTU grant No. SGS16/124/OHK3/1T/18 both funded by the Ministry of Education, Youth and Sports of the Czech Republic.

References