Time-Deterministic Firewalls: Improve Latency and Jitter Behavior of Industrial Firewalls

Markus Schramm
Eberhard Karls Universität Tübingen
markus.schramm@student.uni-tuebingen.de

I. Motivation and Problem

Industrial networks usually involve time-critical applications which transmit and receive messages in a fixed schedule. With an increasing network size, it becomes more and more important to segment the networks using firewalls to improve the security. But to date, there are no firewalls available that enable the transmission of packets with a guaranteed latency and jitter to satisfy the requirements in an industrial network.

The goal of the thesis is to modify an existing software firewall so that it guarantees a maximum packet latency and a low jitter to satisfy the time constraints of industrial applications. To achieve this goal, the thesis will propose and compare several methods, which aim to reduce latency and jitter, by measuring their performance and analyzing their security.

II. Design

The three proposed methods to reduce latency and jitter are described in the following. After that, the features that a firewall must implement to support these methods are described.

A. Methods

a) Timebound method: The timebound method defines a maximum time budget (i.e., amount of time like $100\mu s$) that is invested into processing a packet. The time budget is either the same budget for each packet or individual for each packet flow. If the time budget exceeds, the packet is either forwarded without further checks or dropped. After the packet was forwarded, the checks are continued without any time constraints (but still as fast as possible).

b) Passive method: The passive method forwards packets without any checks. Instead, all checks are performed after the packet was forwarded.

c) Priority method: The priority method divides packets into high-priority packets and low-priority packets (the mechanism to determine the priority of a packet may be the same as the mechanism used to derive the time budget for the timebound method). High-priority packets are processed with a higher priority than low-priority packets. This means, if a high-priority packet arrives while a low-priority packet is currently processed, the processing of the low-priority packet is paused to speed up the processing of the high-priority packet. The processing state of the low-priority packet can be saved so that processing can be continued at the position where it was paused as soon as the high-priority packet processing is finished.

B. Features

From the methods above, the following features can be inferred which allow a firewall to support these methods.

a) Time tracking: The firewall has to measure the time since the packet is processed by the firewall. This means, the arrival time (or the time when processing starts) must be stored and constantly be compared to the current time to recognize when the time budget exceeds. Constantly means that, for example, the processing time is compared after each ACL rule check.

b) Determine time budget: To limit the processing time, the time budget of a packet must be known to the firewall. There are three sources for the time budget: a) Hard-coding the same time budget for all packets. b) Reading the priority from the packet, e.g., the code point field which is part of the VLAN header (IEEE 802.1Q[1]) and translating the priority into the time budget (the priorities in the VLAN header are also used for Time-Sensitive Networking). c) Letting the user configure a time budget/priority rule for each individual flow using additional firewall rules (compared to configuring ACL rules). Option c) might introduce a significant overhead if many priority rules are configured (as processing the priority rules is similar to processing ACL rules), so it should be used sparsely.

c) Early forwarding/dropping: Normally, a firewall requires a full pass through all checks before a packet is forwarded. To support the timebound and passive methods, the firewall must forward or drop packets early while storing the position where the checks were interrupted for later use.

d) Continue paused processing: After an interruption, the firewall must notice that there are packets that need a subsequent check. Here it is beneficial if the packet processing can continue where it was paused to save some processing time.

e) Handle wrong decisions: If a packet was forwarded or dropped wrongly (timebound or passive method), the firewall should be capable of logging the event or taking additional

action. Possible actions in addition to logging will be discussed in the thesis.

f) Pause low-priority packets: To support pausing low-priority packets as defined in the priority method, the firewall must be able to pause packet processing at any time without forwarding or dropping the packet. The pausing should be triggered by the arrival of a high-priority packet. The classification into high-priority or low-priority can be similar to the mechanisms to determine the time budget which were already explained.

g) Late forwarding/dropping: If checks of a low-priority packet are continued after processing was paused, the packet still needs to be forwarded when the checks are finished and the packet is permitted. This is in contrast to the timebound and passive methods where the packets were already forwarded or dropped after the initial check.

h) Stateful packets: Stateful packets may receive a special handling. For example, packets that do not contain sensitive data may be permitted without initial checks, but any packets containing sensitive data should be paused until it is clear that the connection is permitted. Sensitive data may, for example, consist of control commands to a machine while insensitive data may consist of parts of the connection establishment.

C. Firewall Selection

All the features above will be implemented into an existing software firewall (as long as it is technically possible to implement them) which allows measuring the influence of the proposed methods on latency and jitter (one method at a time or a combination of them).

Three software firewall candidates were examined on their performance and ease of implementability of the features: *netfilter* which is part of the Linux kernel in combination with iptables/nftables[2], *bpf-iptables*[3] which enables the use of iptables commands to configure an eBPF firewall and *FD.io*[4], a user space network stack which completely replaces the Linux network stack for improved performance (including an ACL plugin for firewall functionality).

The examination revealed that FD.io offers the best throughput and latency and is also easily expandable due to its plugin architecture (however, at least some modifications of the ACL plugin will be required). In contrast, for example netfilter requires modifications of the Linux kernel while throughput is much lower and latency is much higher than when using FD.io.

Because of the (in comparison) simple modifiability and the high performance (as the latency and jitter should be as low as possible), the implementation in the thesis will rely on FD.io.

III. Evaluation

The evaluation will consist of latency and jitter measurements as well as security considerations.

Latency and jitter of each method (or a combination of them) will be compared with the unmodified version of FD.io. The measurements will be performed on a relatively low-performance industrial firewall whose software is replaced with a clean, Linux-based OS on which the modified version of FD.io is installed. A dedicated traffic generator will put (a yet unspecified) load on the firewall to test its latency and jitter behavior (the measurements may not be limited to latency and jitter, but they are most important).

Besides throughput, latency and jitter, the security is the most important aspect of a firewall. Because of this, each method will be analyzed for its security and possible security tradeoffs.

The goal is to provide the best latency and jitter behavior possible (at least it should be better than the unmodified version, ideally comply to the specifications of Time-Sensitive Networking) while maintaining the security level or providing known tradeoffs.

IV. Related Work

While latency and jitter are important in industrial networks, they are often not considered when measuring firewall performance or when optimizing firewalls as throughput is more important in standard networks.

But there is some research which puts more focus on firewall latency, for example, Cereia et al. [5] measured the latency and jitter of an industrial firewall in three modes, decommissioned mode (only forwarding), normal firewall operation mode and DPI mode (to apply Modbus filtering rules) and compared the results. The measured Modbus connection was the only load on the firewall, no other packets were processed by the firewall during the measurement which is a rather synthetic scenario.

Cheminod et al. [6] assumed a scenario where the firewall is placed between an office network and an industrial control network. They measured how much load (typical office traffic and Modbus traffic) they could put on the firewall while still staying below a defined latency. The latency, in this case, is the duration of a whole Modbus request which incorporates the time of the packet reaching the Modbus device and its answer back to the sender. It must be noted that their results are also affected by the DPI performance of the firewall to validate the Modbus packets while the office traffic is checked by simple layer 3 and 4 rules.

Another paper by the same authors [7] extends the paper mentioned above. They did not use the former network layout anymore, instead they used a simple testbed with a traffic generator, a firewall and a receiver. The authors additionally measured the jitter and included measurements without DPI.

In contrast to the already mentioned works, Zvabva et al. [8] performed measurements of latency, jitter and packet loss with the concept of zones, conduits and security levels according to IEC 62443 in mind, which is a standard for industrial network security.

Instead of just measuring the performance, Pesé et al. [9] developed a proof-of-concept automotive firewall. To achieve a low latency and a low jitter, they combined a hardware and

software solution while the focus in the thesis will be a pure software solution.

V. Result

The result of the thesis will be a proof-of-concept firewall with corresponding measurements that show whether a firewall with latency and jitter guarantees is a realistic idea.

When successful, the proof of concept can be further refined and transformed into a production-ready firewall which improves the security of large industrial networks where strong time constraints must be satisfied.

References

[1] "Ieee standard for local and metropolitan area network–bridges and bridged networks," *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1–1993, 2018.

[2] "netfilter/iptables project homepage - The netfilter.org project," https://www.netfilter.org/, accecssed: 2021-06-10.

[3] S. Miano, M. Bertrone, F. Risso, M. V. Bernal, Y. Lu, and J. Pi, "Securing linux with a faster and scalable iptables," *SIGCOMM Comput. Commun. Rev.*, vol. 49, no. 3, p. 2–17, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3371927.3371929

[4] "FD.io - The Universal Dataplane," https://fd.io/, accecssed: 2021-06-10.

[5] M. Cereia, I. C. Bertolotti, L. Durante, and A. Valenzano, "Latency evaluation of a firewall for industrial networks based on the tofino industrial security solution," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–8.

[6] M. Cheminod, L. Durante, A. Valenzano, and C. Zunino, "Performance impact of commercial industrial firewalls on networked control systems," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–8.

[7] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "Performance evaluation and modeling of an industrial application-layer firewall," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2159–2170, 2018.

[8] D. Zvabva, P. Zavarsky, S. Butakov, and J. Luswata, "Evaluation of industrial firewall performance issues in automation and control networks," in *2018 29th Biennial Symposium on Communications (BSC)*, 2018, pp. 1–5.

[9] M. D. Pesé, K. Schmidt, and H. Zweck, "Hardware/software co-design of an automotive embedded firewall," in *WCX™ 17: SAE World Congress Experience*. SAE International, mar 2017. [Online]. Available: https://doi.org/10.4271/2017-01-1659