Performance Comparison of VPN Solutions

Lukas Osswald, Marco Haeberle, and Michael Menth
University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany
Email: lukas.osswald@student.uni-tuebingen.de, {marco.haeberle,menth}@uni-tuebingen.de,

I. Introduction

Virtual Private Networks (VPN) is the state-of-the-art method to build secure connections between remote hosts over public networks. In times of high-speed connections to the internet, a need for personal information security and business cases, like cloud computing, high data throughput and a stable connection are increasingly important.

OpenVPN [9] is an open source VPN solution which can be deployed on a wide range of platforms, including common operating systems and public cloud platforms. For cryptography, OpenVPN relies on OpenSSL which enables TLS support. IPsec is a protocol suite consisting of two protocols (Encapsulated Security Payload, Authentication Header) and two modes (Tunnel, Transport). Depending on configuration, IPsec offers confidentiality and authenticity [2]. WireGuard [6] describes itself as a modern, fast, and secure VPN tunnel because it uses modern cryptographic algorithms and implements a new cryptographic API called Zinc. The developer puts big emphasis on minimalism which manifests in its small code base, easy configuration, lack of a key distribution mechanism, and a fixed cipher suite. It is frequently covered by media [8] because of its upcoming merge into Linux Kernel 5.6 and its performance claims [7].

Benchmarks of VPN solutions have been discussed in related work, but the data is quite old or uses other setups [1][4][3]. Furthermore, we noticed that the benchmarks from the WireGuard whitepaper seem unrealistic, even if we take protocol overhead into account [7]. In this work, we have decided to conduct VPN benchmarks ourselves. In the following paragraphs we describe our setup and look at three heavily used VPN solutions: OpenVPN, IPsec and WireGuard.

II. Test setup

The testbed is shown in Figure 1. It consists of two virtual hosts which are connected directly via a 10 Gb/s link. They are configured to be in different subnets. We measure network throughput, ping time, and retransmission count of TCP traffic with and without CPU pinning.

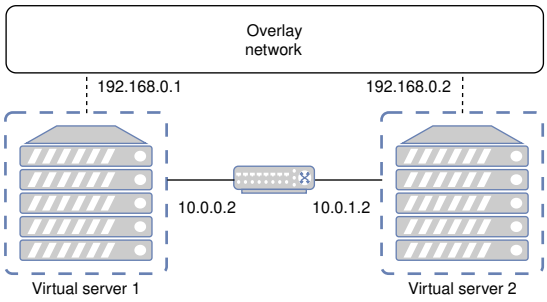

Figure 1. Testbed setup.

Both virtual hosts are running Ubuntu 18.04 with Linux kernel 5.3.5. Unfortunately for us, WireGuard will be upstream only starting with kernel 5.6, thus we use the kernel module with version 0.0.20190913 [5]. For configuring IPsec, we use strongSwan in version 5.6.2. OpenVPN is used in version 2.4.7. The network cards, Intel X710 with 10 Gb/s, are passed through directly to the virtual hosts. The host system is equipped with an Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz, the virtual machines use 4 cores and 16 GB RAM. Our tool set consists of nuttcp 8.1.4 for measuring throughput, ping for measuring ping time and mpstat 11.6.1 for measuring detailed CPU utilization.

III. Results and Discussion

We test the VPN systems in different configurations. For IPsec, we test all combinations of protocols and modes as well as various cryptographic algorithms. We also collect data which shows that route-based IPsec processing is equal to the policy-based processing in terms of throughput. For OpenVPN, we use the same cipher suites except for ChaCha20-Poly1305 because it is not supported in the currently available version of OpenVPN. Furthermore, we test several methods to increase throughput as suggested by OpenVPN [10]. WireGuard only offers one cipher suite for configuration, thus there is only one test case. For AES-based encryption, we also have a look at the performance gain from hardware acceleration through AES-NI. At last, we look at the effect of CPU pinning on network throughput.

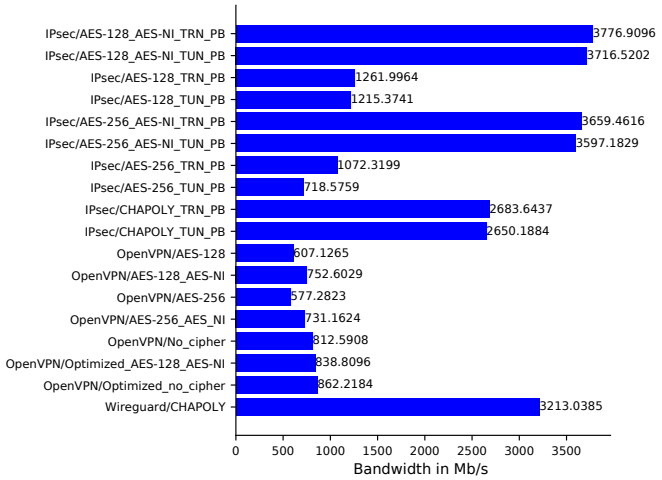

Figure 2. Network throughput without CPU pinning, CPU turbo boost enabled, and 4 CPU cores.

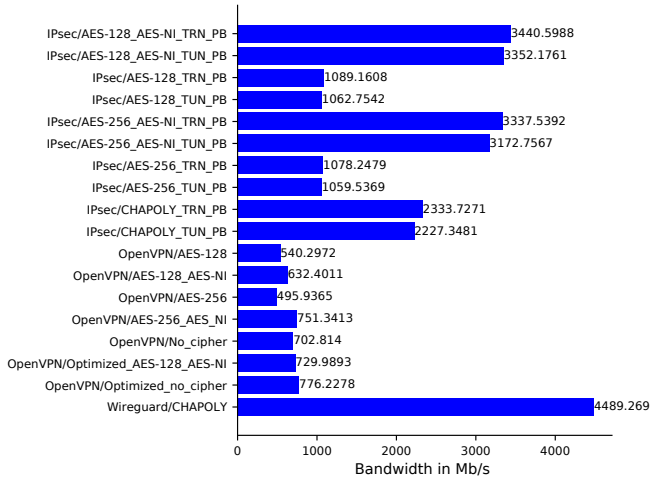

Figure 3. Network throughput with CPU pinning enabled and CPU turbo boost disabled, and 3 CPU cores.

Figure 2 shows the benchmark results of tests without CPU pinning. All IPsec variants with hardware accelerated AES-based encryption surpass WireGuard's throughput. The importance of using AES-NI for AES-based encryption is also shown well by the data, as well as a small superiority of AES-128 compared to AES-256. Also, there is a small, but visible throughput difference in favor of transport mode because of the smaller protocol overhead. OpenVPN does not compete with both systems in any configuration we tested. Even deactivating encryption has no large effect on performance. This shows that encryption of data is not the bottleneck for OpenVPN. The throughput of WireGuard is significantly larger than the ChaCha20-Poly1305 variants of IPsec, presumably because WireGuard uses parallel workers to encrypt and decrypt data [7]. IPsec is implemented in the xfrm part of the Linux kernel and does not take advantage of parallelism. Thus, WireGuard's performance is correlated to CPU performance.

Next we pin the cores of the virtual machines to physical cores. The results are shown in Figure 3. CPU pinning enables WireGuard to increase its throughput by about 40 percent because it can use the CPU more efficiently (less context switches) even though we have to decrease the number of CPU cores by one and disable turbo boost. The disabled turbo boost can be noticed in an overall drop in performance of IPsec and OpenVPN test cases.

IV. Conclusion

We conclude that IPsec with AES-based encryption is a very good choice regarding performance in a virtualised environment without CPU pinning. In non virtualised environments, WireGuard surpasses IPsecs performance by about 30 percent. In general, it should be noted that the choice of the right VPN system should be multifactorial, resulting in use cases for all three systems.

References

[1] C.J.C. Pena and J. Evans. "Performance evaluation of software virtual private networks (VPN)". In: *Proceedings 25th Annual IEEE Conference on Local Computer Networks. LCN 2000* (2000).

[2] S. Kent and K. Seo. *Security Architecture for the Internet Protocol*. RFC4301. 2005.

[3] Shaneel Narayan ; Kris Brooking ; Simon de Vere. "Network Performance Analysis of VPN Protocols: An Empirical Comparison on Different Operating Systems". In: *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing* (2009).

[4] P. Rybár I. Kotuliak and P. Trúchly. "Performance comparison of IPsec and TLS based VPN technologies". In: *2011 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)* (2011).

[5] Jason A. Donenfeld. *Wireguard is now in Linus' tree*. 2020. URL: https://lists.zx2c4.com/pipermail/wireguard/2020-January/004906.html.

[6] Jason A. Donenfeld. *WireGuard website*. 2020. URL: https://www.wireguard.com/.

[7] Jason A. Donenfeld. *WireGuard Whitepaper*. 2020. URL: https://www.wireguard.com/papers/wireguard.pdf.

[8] Sebastian Grüner. *Wireguard in Linux-Kernel eingepflegt*. 2020. URL: https://www.golem.de/news/vpn-technik-wireguard-in-linux-kernel-eingepflegt-1912-145437.html.

[9] OpenVPN Inc. *OpenVPN website*. 2020. URL: https://openvpn.net/.

[10] OpenVPN Inc. *Optimizing performance on gigabit networks*. 2020. URL: https://community.openvpn.net/openvpn/wiki/Gigabit_Networks_Linux.