Secure and Scalable Remote Access Tunnels for the IIoT: An Assessment of openVPN and IPsec Performance

Frederic Pohl, Hans Schotten

Secure and Scalable Remote Access Tunnels for the IIoT: An Assessment of openVPN and IPsec Performance

Frederic POHL and Hans Dieter SCHOTTEN

German Research Center for Artificial Intelligence,
Intelligent Networks Research Group,
Trippstadter Str. 122, D-67663 Kaiserslautern, Germany
frederic.pohl@dfki.de

Abstract. Nowadays, industrial production already benefits from an increased level of interconnection involving various heterogeneous production assets. Future development in the area is likely to lead to a scenario often referred to as the Industrial Internet of Things (IIoT), a promising factor in achieving unseen productivity goals. One of the key IIoT use cases is remote access, which can drastically reduce the requirement for on-site presence of technicians and thus eliminate a large cost factor. In this paper, we present a detailed examination of two widespread Virtual Private Network (VPN) remote access frameworks and analyse their suitability for IIoT remote access facilities. We introduce a cloud architecture that seamlessly integrates with existing highly segmented and firewalled industrial networks, yet providing secure connectivity through the use of openVPN and IPsec technology. With scalability being a key factor for a cloud architecture, we give an analysis of our favoured protocols in order to derive potential performance bottlenecks. We finally verify our assumptions by providing empirical performance measurements.

Keywords: Industrial Internet of Things, Network Security, Remote Access, Virtual Private Networks, IPsec, openVPN

1 Introduction and Motivation

Complex industrial production processes, as of today, are highly computerized and involve a large number of interconnected devices. Yet, interconnection of production environments as a driver for highly optimized production processes is predicted to continue in the future, thus allowing for novel business models often summarized by the visionary term of a “fourth industrial revolution” [11].

Within this vision of heavily interconnected “smart factories” [19], a key element is remote access to the interconnected components involved in production processes. A robust remote access framework not only allows to reduce costs by reducing on-site maintenance and incident durations but also is an enabler for various machine-to-machine interaction scenarios. Malicious use of remote access frameworks, however, must be prevented by enforcing secure authentication and encryption facilities, which should be flanked by an anomaly detection framework. IPsec [13] and openVPN [1] are well-established solutions to achieve the first goal on the network layer; the second goal, despite being out of the scope of this work, can be achieved on the same cloud infrastructure by inspecting traffic that is forwarded by a centralized VPN endpoint between the involved entities.

This paper evaluates the suitability of the aforementioned VPN technologies for such a massive IIoT remote administration architecture and is organized as follows: Section 2 gives an overview of the related work. Section 3 describes our evaluation platform and compares involved IPsec and openVPN protocol properties. In Section 4 we present an empirical performance evaluation of the core cloud component for both protocols. Section 5 discusses the results we obtained and concludes this work.

2 State-of-the-Art and Related Work

The wide availability of Internet Protocol (IP) based packet switched networks, in conjunction with IP-based VPN protocols allowing to tunnel traffic to and from different private domains[1], allows for flexible remote access setups. Nowadays, there exists a variety of VPN protocols to tunnel network or data link layer traffic, yet many of them provide little to no security [14]. With an increasing awareness of security requirements in the internet domain, the most widely used VPN technologies therefore either comply with the IPsec standard or use a Transport Layer Security (TLS) [9] framework, as openVPN does.

In the context of IIoT scenarios involving thousands of connected devices, the performance of VPN technology is very important. A comparison of maximally achievable bandwidths and response times using IPsec and openVPN was performed by Kotuliak, Rybár, and Truchly [15] with IPsec outperforming openVPN. Migault et al. analysed processor overheads of different IPsec and cipher suite operation modes and observed significant performance improvement upon activation of hardware acceleration for encryption [16].

Most related work however focuses on evaluating performance in bidirectional VPN setups and thus only partly applies for the remote access platform we will present in section 3. Our contribution consists in a performance evaluation of a remote access platform taking the role of a trusted intermediary in secure tunnelling scenarios for the IIoT.

3 Platform Architecture

Fig. 1 depicts our evaluation architecture for secure, session-based end-to-end tunnelling between entities located in disjoint private network zones A, B, each isolated by at least one firewall and/or Network Address Translation (NAT) [20]

[1] employing private IPv4 address ranges according to [18]

layer. An entity in this context represents any IP addressable device. The architecture's core component is a cloud platform trusted by the operators of both private networks, which consists of:

- a session database that contains all scheduled tunnelling events,
- a VPN endpoint that provides encryption and authentication facilities,
- a routing engine that forwards incoming packets to the respective recipient.

The cloud platform is located in zone C and must be reachable from the private zones. Tunnels are established by the entities in the private subnets, traffic within the remote access tunnel is therefore always directed to and originated from the platform's VPN endpoint, minimizing firewall configuration effort for operators of the respective private zones.

The platform is not limited to traffic forwarding tasks. An important architectural property lies within traffic being available in decrypted plain-text inside the platform, which we deem beneficially in the context of data aggregation and anomaly detection scenarios as described in [10]. Other processing scenarios such as accounting and monitoring are conceivable. Note that the architecture does not break with application layer security entities may employ to prevent deep packet inspection within the platform.

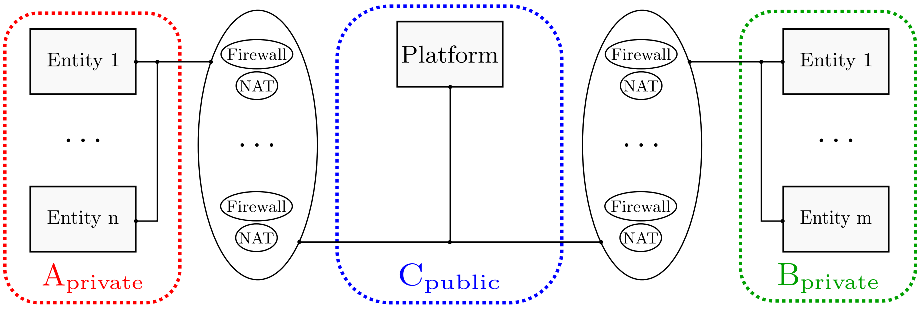


Fig. 1: Platform and Test-bed Architecture

In the context of these remote access scenarios, we deem high relevance to the performance of the platform's VPN endpoint in high traffic load conditions and a large number of connection attempts. From a cryptographic point-of-view, there exist various optimizations [7] which allow for fast cryptographic processing of VPN traffic. Nonetheless, different implementation approaches of IPsec and openVPN introduce overheads: openVPN encrypts and decrypts VPN traffic in user space and uses TUN/TAP interfaces to interact with system space routines responsible for actual traffic dispatching via physical network interfaces; session keys are exchanged using a TLS handshake [17]. IPsec traffic, in contrast, is processed by system space routines based on traffic selection and session key container structures called Security Associations (SA). SAs can be setup in the system space using the Internet Key Exchange (IKE) [12] protocol that allows for session key exchange with a reduced number of messages in comparison with the TLS handshake.

Given these considerations and due to the fact that switching from user to system space and vice versa introduces a context switching overhead, we expect IPsec to perform more efficiently under heavy traffic load conditions as it should not be subject to context switching overhead. Yet, both IPsec/IKE and openVPN should provide similar performance when confronted with a large number of key exchange requests.

4 Experimental Performance Evaluation

In order to verify our assumptions, we provide two separate evaluations of the performance of the central platform depicted by Fig. 1. The first measurement targets at the maximum achievable platform throughput that can be realized with openVPN and IPsec and compares the resulting CPU utilization. The second measurement evaluates the platform's CPU utilization for both VPN endpoints upon being confronted with a large number of key exchanges. While maximum throughput provides a good performance measure in a highly active network, key exchange performance is relevant in the context of massively interconnected IoT devices where connections are established and closed frequently.

We use an evaluation test-bed consisting of both virtual and physical entities in the private zones and a virtualized central platform. NAT/Firewall layers are also virtualized with the help of isolated kernel network namespaces. The virtual entities use the QEMU [3] virtualization engine with each entity allocated a dedicated CPU core (Intel Core i7-6700K) and a Virtual/IO-Network device that provides link speeds in the range of the underlying system's PCI Bus, in our case 25 GBit/s. It should be noted however that, due to the architectural approach of routing all traffic within the platform, the maximum theoretically achievable end to end bandwidth is only half the link bandwidth, thus 12,5 GBit/s. Nonetheless, this setup allows us to efficiently stress the central platform without needing to deploy hundreds of IIoT devices.

openVPN as well as the strongSwan [4] IPsec suite were evaluated using the AES [5] symmetric cipher in Cipher Block Chaining (CBC) mode with 128-Bit key size in conjunction with HMAC-SHA256 [6] as PRF and for integrity checking. The AES algortihm was selected with respect to AES NI hardware acceleration available in the testbed. Nonetheless, with a measured maximum AES en-/decryption rate of 1.5 GB/s, we ensured that the CPU, not the link, formed the platform's bottleneck. All CPU and network metrics were recorded on the central platform and evaluated using the Performance Co-Pilot open source software suite [2].

4.1 Maximum Throughput

Fig. 2 shows the maximum platform throughput achieved for openVPN and IPsec and highlights that IPsec clearly outperforms openVPN in this respect. The main reason can be recognized from Fig. 3a, which shows the CPU partly running in `user`, `kernel` and `irq`. `irq` mode handles interrupt routines required when switching from `user` to `kernel` mode and vice-versa, but in Linux systems also performs IPsec packet processing. This is visualized in Fig. 3b, which highlights that IPsec processing does not trigger expensive context switches and confirms our previous implications.

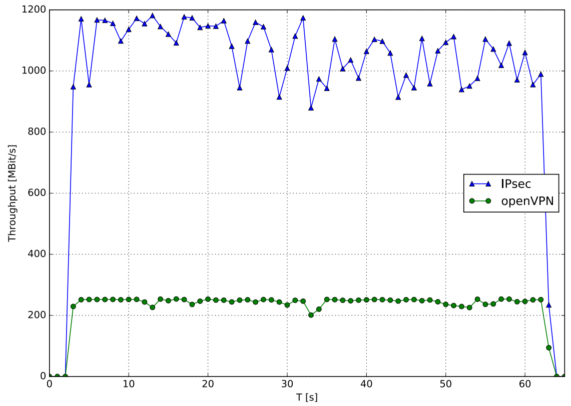
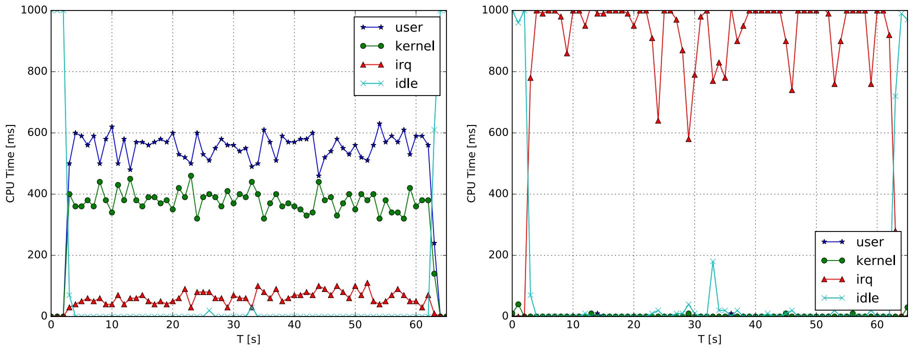


Fig. 2: Maximum Platform Throughput achieved by IPsec and openVPN

4.2 Key Exchange

In order to compare the platform's key negotiation performance for openVPN and strongSwan, we repeatedly initiated tunnel initiation floods originating from a total of four entities towards the openVPN platform endpoint. After successful key exchange, tunnels were closed immediately. We determined a maximum frequency $f_{\max} = 0.04s$ where all key exchanges were still successful. Fig. 4a shows a 60 seconds key exchange flood towards the platform's openVPN endpoint. Processing mostly occurs in user space, which is what we expected. However, one easily observes the remarkable portion of overall `idle` CPU time frames, which we can only suspect to be caused by openVPN implementing an internal key exchange rate limiter not known to us.

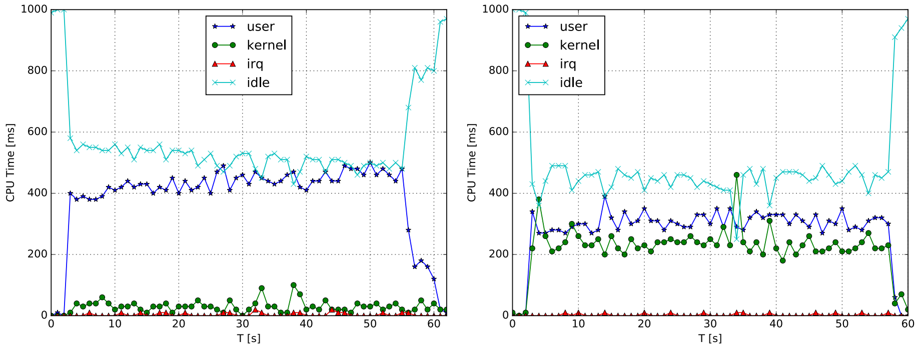
To provide better comparability, we flooded strongSwan using the same parameters. Fig. 4b shows that strongSwan deals more efficiently with the key exchange, despite often switching between `user` and `kernel` mode which most likely results from installing negotiated IKE and IPsec SAs in the respective kernel structures.



(a) openVPN

(b) strongSwan

Fig. 3: Platform CPU Utilization during Throughput Measurement



(a) openVPN

(b) strongSwan

Fig. 4: Platform CPU Utilization during Key Exchange flood at $f_{\max}$ from four entities

5 Conclusion and Outlook

In this paper, we presented a scalable architecture that is able to flexibly interconnect heterogeneous IIoT entities located within segmented and highly firewalled environments. We therefore focused on the widespread and well-known openVPN and IPsec tunnel protocols which not only provide good security mechanisms but also are able to carry legacy protocols, which is extremely important in industrial contexts. Our work gives abstract estimates on their packet processing and key exchange performance, which are widely confirmed by our empirical measurements. Both theoretical and empirical results strongly suggest that in case of a critical performance, either imposed by throughput or by key exchange rate requirements, IPsec is favourable over openVPN. A promising approach of an IKEv2 mediation server that mediates direct IPsec host-to-host connections has been proposed by [8] and would even increase IPsec performance in similar architectures.

These performance parameters however, do not denote all aspects of both protocols. Although openVPN suffers from weak performance, its very simple configuration by far outperforms IPsec complexity and possible resulting security issues on the other hand. The simple portability of openVPN additionally makes it more attractive in certain situations.

While in the future, remote assistance protocols might arise that integrate more specifically with the IoT and IIoT specifically, we have shown that state of the art VPN solutions can provide a scalable bridging technology that enables end-to-end tunnelling for legacy as well as novel devices.

Acknowledgement This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen KIS4ITS0001, IUNO). The authors alone are responsible for the content of the paper.

References

[1] openvpn, `https://openvpn.net/`
[2] Performance co-pilot, `http://pcp.io`
[3] Qemu, the fast! processor emulator, `http://www.qemu.org`
[4] strongswan, opensource ipsec-based vpn solution, `https://strongswan.org/`
[5] Fips pub 197, advanced encryption standard (aes) (2001), u.S.Department of Commerce/National Institute of Standards and Technology
[6] Fips pub 180-2, secure hash standard (2002), u.S.Department of Commerce/National Institute of Standards and Technology
[7] Bogdanov, A., Lauridsen, M.M., Tischhauser, E.: Aes-based authenticated encryption modes in parallel high-performance software. IACR Cryptology ePrint Archive 2014, 186 (2014)
[8] Brunner, T.: Ikev2 mediation extension. Internet-Draft draft-brunner-ikev2-mediation-00, IETF Secretariat (April 2008), `http://www.ietf.org/internet-drafts/draft-brunner-ikev2-mediation-00.txt`

[9] Dierks, T., Rescorla, E.: The transport layer security (tls) protocol version 1.2. RFC 5246, RFC Editor (August 2008), http://www.rfc-editor.org/rfc/rfc5246.txt

[10] Duque Antón, S., Fraunholz, D., Zemitis, J., Pohl, F., Schotten, H.D.: Highly scalable and flexible model for effective aggregation of context-based data in generic iiot scenarios. In: Kopp, O., Lenhard, J., Pautasso, C. (eds.) 9th Central European Workshop on Services and their Composition. Central European Workshop on Services and their Composition (ZEUS-2017), February 13-14, Lugano, Switzerland. pp. 51–58. CEUR Workshop Proceedings (4 2017)

[11] Kagermann, H., Wahlster, W., Helbig, J.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry. Forschungsunion (2013)

[12] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., Kivinen, T.: Internet key exchange protocol version 2 (ikev2). RFC 7296, RFC Editor (October 2014), https://www.rfc-editor.org/rfc/rfc7296.txt

[13] Kent, S., Seo, K.: Security architecture for the internet protocol. RFC 4301, RFC Editor (December 2005), https://www.rfc-editor.org/rfc/rfc4301.txt

[14] Khanvilkar, S., Khokhar, A.: Virtual private networks: an overview with performance evaluation. IEEE Communications Magazine 42(10), 146–154 (2004)

[15] Kotuliak, I., Rybár, P., Truchly, P.: Performance comparison of ipsec and tls based vpn technologies. In: Emerging eLearning Technologies and Applications (ICETA), 2011 9th International Conference on. pp. 217–221. IEEE (2011)

[16] Migault, D., Palomares, D., Guggemos, T., Wally, A., Laurent, M., Wary, J.P.: Recommendations for ipsec configuration on homenet and m2m devices. In: Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks. pp. 9–17. Q2SWinet '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2815317.2815323

[17] Novickis, T.: Protocol state fuzzing of an openvpn (2016)

[18] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., Lear, E.: Address allocation for private internets. RFC 1918, RFC Editor (January 1996), https://www.rfc-editor.org/rfc/rfc1918.txt

[19] Sadeghi, A.R., Wachsmann, C., Waidner, M.: Security and privacy challenges in industrial internet of things. In: Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE. pp. 1–6. IEEE (2015)

[20] Srisuresh, P., Egevang, K.: Traditional ip network address translator (traditional nat). RFC 3022, RFC Editor (January 2001), https://www.rfc-editor.org/rfc/rfc3022.txt