Performance Evaluation of Software Routers with VPN Features

Hasan Redžović, *Graduate Student Member, IEEE,* Aleksandra Smiljanić, *Member, IEEE,* and Bogdan Savić

Abstract **- This paper presents implementation and analysis of the VPN software router which is based on Quagga and strongSwan open-source software tools. We validated functionalities of strongSwan and Quagga in realistic environment which include scenarios with link failures. Also, we measured and analyzed performance of encryption and hash algorithms supported by strongSwan software, in order to advise optimal VPN configuration which provides the best performance.**

Keywords **- IPsec protocol, Quagga, Software Routers, Software VPN Solutions, strongSwan.**

I. Introduction

The performance of commodity PC hardware is constantly improving. New affordable hardware components such as faster CPUs with larger number of cores and higher speed network cards are being released each year. These constant improvements of commodity hardware provide new opportunities for utilization of software routers. Flexibility of development, implementation and configuration are the main advantages of software defined networks (SDN) [1] and software routers [2].

One of the most CPU consuming networking features are related to security, in particular to encryption and hash algorithms. In this paper, we examine data plane capabilities of software router with VPN features which is based on the Linux kernel network stack. The data plane which implements IP packet routing and VPN functionality in the Linux kernel space can be used as a foundation for creating SDN with VPN features.

Using Quagga [3] and strongSwan [4] software tools, we have created software router with VPN features and analyzed its performance in real conditions. Quagga is a popular open-source control plane which utilize Linux network stack as a router data plane. VPNs can be implemented using various security protocols such as IPsec, TLS/SSL and SSH that operate at different layers of the TCP/IP stack.

SSH protocol protects data at the application layer and it can be used in port-forwarding mode to create secure tunnels for data exchange between applications. TLS/SSL provides security at the transport layer. IPsec protocol provides protection at the network layer and enables creation of secure tunnels for exchange of all IP packet traffic between distant private networks or users [5]. IPsec protects not only data exchanged between the users, but also their identities.

Software-based IPsec can use Linux kernel IPsec features or it can be implemented in the user space. We have shown earlier that IPsec implemented in user space has poorer performance than the IPsec implemented in Linux kernel [6].

StrongSwan implements IPsec protocol in Linux kernel space. Using IKE daemon strongSwan configures ESP and AH protocols implemented in Linux kernel. When strongSwan is integrated with Quagga, flexible VPNs can be implemented that follow changes of network topology. StrongSwan is a flexible software VPN solution that supports a large number of encryption and hash algorithms. We have analyzed performance of encryption and hash algorithms provided by strongSwan, in order to provide insight into capabilities of strongSwan and determine optimal VPN configuration.

In the rest of this paper, Section II describes briefly Quagga software and its architecture. Section III presents the basics of IPsec protocol. The strongSwan software with list of supported security algorithms is described in Section IV. In Section V, we analyze functionalities of Quagga and strongSwan in real networking environment. Section VI describes evaluation of encryption and hash algorithms in strongSwan. Finally, Section VII discusses results and our plans for future work.

II. Quagga

Quagga is open-source software tool for implementation of routing protocols such as RIP, OSPF, BGP and IS-IS that comprise routing control plane. Linux kernel network stack implements forwarding features of data plane. Using independent process (daemon) called Zebra, Quagga connects control plane in user space to the router data plane in Linux kernel, and integrates software router on commodity hardware in this way. Fig. 1 illustrate Quagga architecture.

Each routing protocol in control plane works as daemon. The main daemon Zebra controls all these daemons, and based on the information obtained from

Hasan Redžović is with the Innovation center of School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (phone: 381-64-4641615; e-mail: hasanetf@live.com).

Aleksandra Smiljanić is with the School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: aleksandra@etf.rs).

Bogdan Savić is with the School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: bogdan.savic1503@gmail.com).

978-1-5090-4086-5/16/$31.00 ©2016 IEEE

them updates the routing table in kernel when the network topology changes.

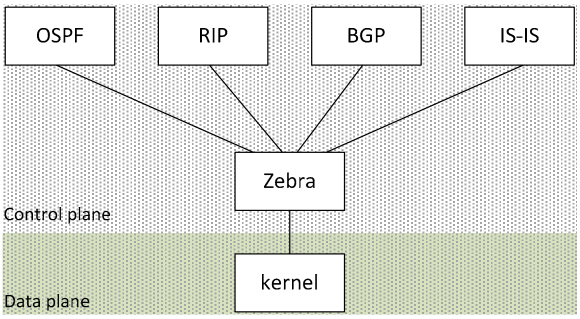


Fig. 1. Quagga architecture.

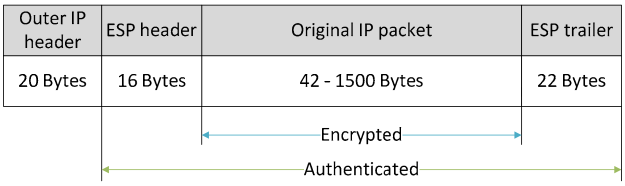
III. IPsec protocol

VPN can be implemented using different security protocols that operate at different layers of TCP/IP stack [7]. IPsec protocol provides data protection at the network layer which means that it will protect all IP packets between distant private networks or users. IPsec protocol is composed of three protocols with different security roles and features:

- Internet Key Exchange (IKE) - handles creation of IPsec tunnels and authentication of distant peers in VPN. IKE protocol also provides periodical symmetrical key exchange;
- Authentication Header (AH) - provides IP packet integrity protection;
- Encapsulating Security Payload (ESP) - provides IP packet confidentiality and integrity protection.

Fig. 2 shows IP packet encapsulation with ESP protocol in tunnel mod. Original IP packet is encrypted with various encryption algorithms such as AES and 3DES which guarantee data confidentiality. ESP header and ESP trailer added to IP packet contain information about IP packet and established IPsec tunnel.

Integrity protection of ESP packet is handled by a hash algorithm, such as MD5, SHA1 and SH2, which creates a fixed size hash unique for every packet. Finally, ESP packet gets additional IP header which is used for sending ESP packet between to VPN gateways or peers through unsecured part of the network.



Outer IP header	ESP header	Original IP packet	ESP trailer
20 Bytes	16 Bytes	42 - 1500 Bytes	22 Bytes

Fig. 2. IP packet encapsulation with ESP protocol in tunnel mod.

IPsec protocol supports many different new and experimental encryption and hash algorithms with different levels of security. Level of security is measured as a number of compute operations necessary to decrypt protected data with brute force.

IV. StrongSwan

StrongSwan is open-source software that implement IPsec based VPN solutions using Linux, FreeBSD, OS X or Android operating system. Encryption and hash algorithms require intensive computing and, for this reason, ESP and AH protocols are implemented in kernel. On the other side, IKE protocol is implemented in user space, and it utilizes ESP and AH protocol kernel support. IKE protocol is implemented as independent daemon named Charon which communicates with different peers in network using IKE protocol messages. After connection establishment, authentication and agreement on IPsec tunnel encryption and hash algorithm, daemon Charon configures ESP and AH protocol in kernel space. Packet processing path is always in kernel space. StrongSwan support following list of encryption and hash algorithms:

- Encryption algorithms: 3DES (rfc1851), AES (rfc3962), CAST-128 (rfc2144), Blowfish, Camellia (rfc3713), ChaCha20 (rfc7539);
- Hash algorithms: MD5 (rfc1321), SHA-1 (rfc3174), SHA-2 (rfc6668), AES XCBC (rfc3566), AES CMAC (rfc4493) and AES GMAC (rfc4543).

V. Evaluation of VPN configuration

Common VPN IPsec network configuration is based on VPN gateways which separate safe private network domain from unsafe public network. The main function of VPN IPsec gateway is to connect remote private networks and/or users by creating secure IPsec tunnels. Fig. 3(a) illustrate basic method of connecting VPN gateway with public network. Fig. 3(b) and 3(c) show cases with redundant links which improve VPN resilience to link failures. Redundant links can be installed between VPN gateway and public network access router, Fig. 3(b), and between VPN gateway and other Internet Service Provider (ISP) as shown in Fig. 3(c). The configurations with redundant VPN links is not fully supported by strongSwan and other software VPN solutions.

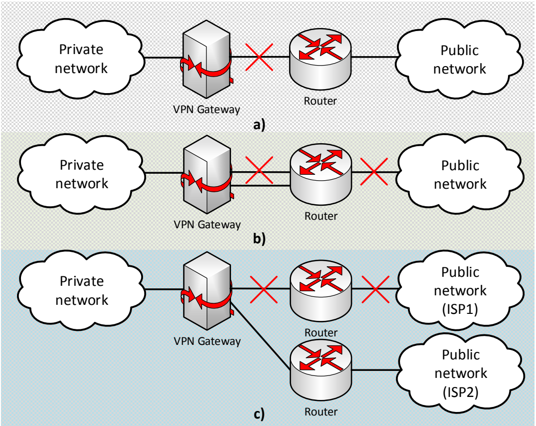


Fig. 3. a) Common VPN configuration; b) and c) VPN configuration with redundant links.

By combining strongSwan VPN gateway and Quagga software router, we manage to configure VPNs which can adapt to the network topology changes when the link failures happen. We will demonstrate behavior of the implemented software router with VPNs in the case of failures.

The testing environment shown in Fig. 4 comprises five software routers connected with 1 Gbit/s links. All software routers in Fig. 4 are using the OSPF routing protocol in Quagga. StrongSwan was installed on the routers R1 and R4. IPsec tunnel has been configured between two private networks: *net_1* and *net_2* using ESP protocol in tunnel mod. The path of the created IPsec tunnel is marked with green color in Fig. 4 and includes routers R1, R2, R3 and R4. The Fig. 4 also shows the path of ESP packets through the router R1 data plane. The numerated arrows in data plane of the router R1 represent the processing sequence for ESP packets received from router R4 and network *net_2*. This sequence has the following order:

- Router R1 is receiving ESP packets through the eth3 port and the IP lookup is performed using IP address of outer IP header shown in Fig. 2;
- It is determined that ESP packets have IP address of the R1 eth3 port;
- ESP packets are passed to strongSwan for decapsulation. Using ESP protocol in kernel space, strongSwan decapsulate IP packets sent from net_2;
- New IP lookup is performed using IP address of original IP packet sent from network net_2;
- It is determined that IP packet have IP address of network net_1. Then, IP packet is passed to the eth1 port of R1 for further delivery through network net_1.

Using OSPF protocol messages, routers communicates with each other in order to learn network topology and determine the shortest path to each node in network. Then, routers update their routing tables. If the network topology changes, for example new link is added or link fails, each router in network is notified and the routing table is updated based on the shortest path Dijkstra algorithm. In the test shown in Fig. 5, link between routers R1 and R2 is disabled. Using OSFP protocol, all routers have updated their information on network topology, calculated the shortest path to each node in network and, finally, updated their routing tables.

Before disabling the link, Router R1 was using eth3 port to communicate with router R2. Also, strongSwan configuration files for routers R1 and R4, use the IP address of the eth3 port as the IPsec tunnel endpoint. When IKE daemon Charon initiate establishment of an IPsec tunnel, IP addresses of the IPsec tunnel endpoints are necessary for exchanging IKE protocol messages. After disabling the link between routers R1 and R2, the OSPF protocol on router R1 is still broadcasting IP address of the eth3 port to the rest of the network. Other routers (R2, R3, R4 and R5) can still communicate to the R1 eth3 port using new path through the R1 eth2 port.

Unhindered communication between R1 eth3 port and rest of the network allows automatic creation of the new IPsec tunnel between routers R1 and R4 along different path shown in Fig. 5. In this case, ESP packets are exchanged between routers R1, R5 and R4, marked in Fig. 5 with blue color. The processing sequence at router R1 for ESP packets received from router R4 is also shown in Fig. 5 and it has the following order:

- Router R1 is receiving ESP packets through the eth2 port and the IP lookup is performed using the IP address of outer IP header;
- It is determined that ESP packets have IP address of R1 eth3 port;
- ESP packets are passed to strongSwan for decapsulation;
- The rest of the packet processing path is the same as in the first case with the link between routers R1 and R2.

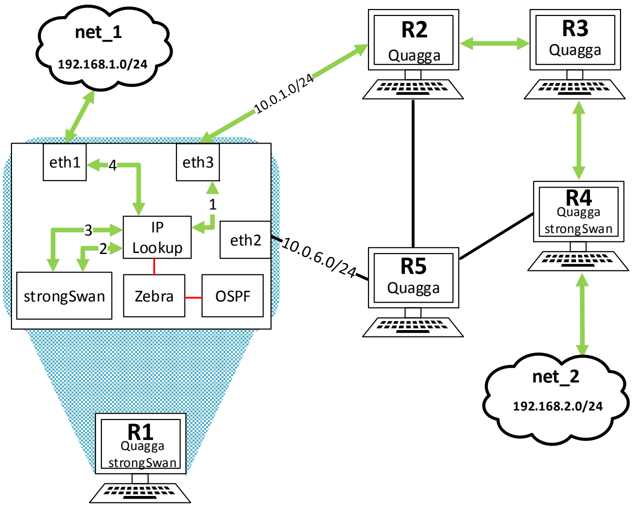


Fig. 4. IPsec tunnel path through kernel and public network.

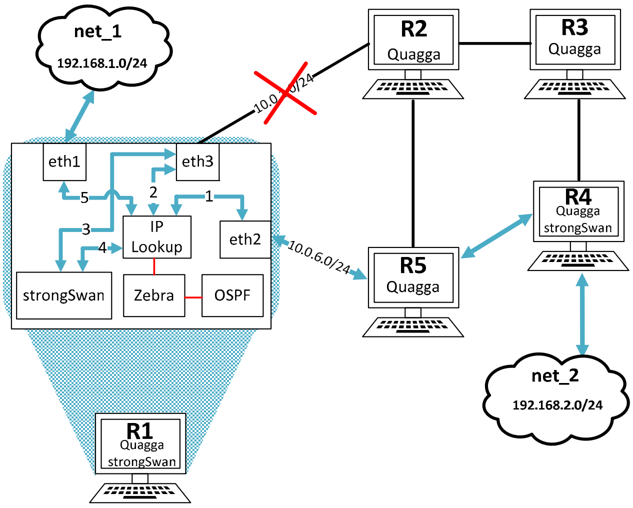


Fig. 5. IPsec tunnel through network and kernel after the link failure.

Unlike in the cases shown in Fig. 3(b) and Fig. 3(c), which cannot be implemented using strongSwan, VPN gateway on router R1 can automatically switch IPsec tunnel to different paths thanks to the Quagga control plane.

VI. Performance of the IPsec security algorithms

Flexibility of software implementations enables fast and relatively easy development of new security algorithms. Encryption algorithms are CPU intensive operations and it is necessary to analyze all strongSwan supported options to determine which encryption algorithm provides the fastest packet processing.

Testing environment is composed of two VPN gateways, connected with 1 Gbit/s link. The IPsec tunnel was established between VPN gateways with ESP protocol and tunnel mode, by means of strongSwan software. Various encryption and hash algorithms were tested, and their speeds were measured.

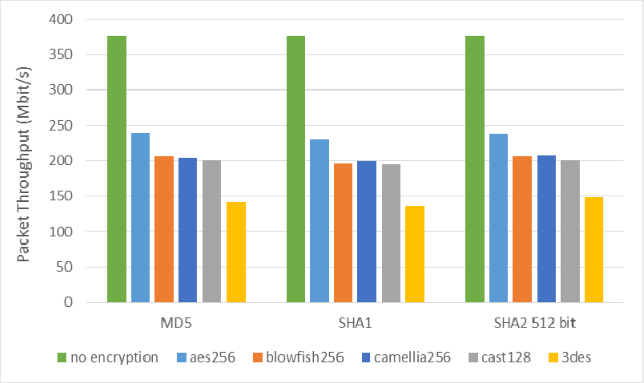


Fig. 7. Processing speed of encryption algorithms in combination with separate hash algorithms

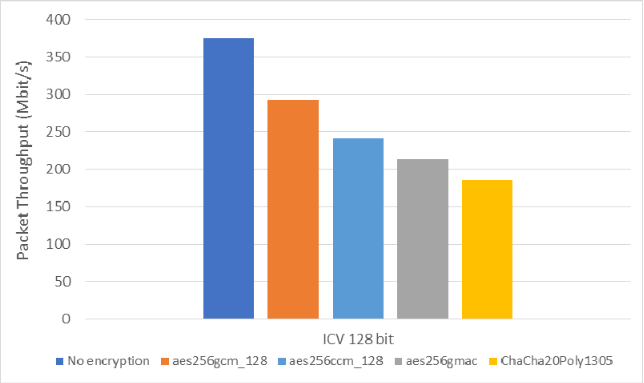


Fig. 8. Processing speed of encryption algorithms that include integrity protection

Fig. 7 shows the performance results for different encryption algorithms (AES, Blowfish, Camellia, CAST-128 and 3DES) combined with three hash algorithms (MD5, SHA1 and SHA2). The hash algorithms which provide integrity protection did not noticeably affect the speed of packet processing. AES algorithms provided the fastest packet processing in comparison to the other encryption algorithms. This performance advantage is due to the fact that strongSwan, Linux kernel and hardware support the Intel AES-NI [8] technology. Fig. 8 shows performance results for algorithms which integrate encryption and integrity protection, StrongSwan supports AES CMM (CBC MAC Mode), AES GCM (Galois/Counter Mode), AES GMAC (Galois Message Authentication Code) and ChaCha20 Poly1305. The CMM mod is combination of algorithms based on CBC (Cipher Block Chaining) and MAC (Message Authentication Codes) architecture.

Security algorithm AES GCM provided the fastest packet processing speed with only 22% speed downgrade in comparison with the routing speed on VPN gateway with no encryption.

VII. Conclusion

In this paper we have analyzed functionalities and performance of software router with VPN features which utilized the Linux kernel. Integration of Quagga and strongSwan software platforms convert commodity hardware into the software VPN router. This configuration provide IPsec tunnels with higher resiliency to link failures. We compared performance of all encryption and hash algorithms supported by strongSwan and concluded that AES GCM provide the best performance.

Acknowledgment

This work was supported by the Serbian Ministry of Science and Education (project TR-32022), and companies Telekom Srbija, and Informatika.

References

[1] A. Sadasivarao, S. Syed and P. Pan, "Open Transport Switch - A Software Defined Networking," in SIGCOMM, Hong Kong, 2013.

[2] R. Bolla, and R. Bruschi, " Linux Software Router: Data Plane Optimization and Performance Evaluation," Journal of Networks, vol. 2, no. 3, 2007.

[3] P. Jakma and D. Lamparter, "Introduction to the Quagga Routing Suite," 2012. [Online]. Available: https://goo.gl/NXbOfL.

[4] "StrongSwan," secunet, 2016. [Online]. Available: https://www.strongswan.org/. [Accessed 1 10 2016].

[5] S. Kent, "Security Architecture for the Internet Protocol," 2005. [Online]. Available: https://tools.ietf.org/html/rfc4301. [Accessed 1 10 2016].

[6] H. Redžović, A. Smiljanić and S. Gajin, "Performance Evaluation of Open-Source VPN Software Implementations," in 3rd International Conference on Electrical, Electronic and Computing Engineering , Zlatibor, 2016.

[7] S. Padhiar and P. Verma, "A Survey on Performance Evaluation of VPN," International Journal of Engineering Development and Research, vol. 3, no. 4, pp. 516-519, 2015.

[8] Intel, "Intel Advanced Encryption Standard (AES) New Instructions Set," 2016. [Online]. Available: https://goo.gl/UGyyfc. [Accessed 1 10 2016].