A Performance Analysis of Gateway-to-Gateway VPN on the Linux Platform

Peter Dulany, Chang Soo Kim, and James T. Yu

PeteDulany@yahoo.com, ChangSooKim@yahoo.com, jyu@cs.depaul.edu

School of Computer Science, Telecommunications, and Information Systems
DePaul University, Chicago, IL, USA

Abstract

This paper presents a gateway-to-gateway configuration of IPSec-based Virtual Private Network (VPN) for the enterprise environment. The growing popularity of the high speed internet connection drives the need for cost-effective, secure intranet connection. Traditionally, such connection is provisioned via expensive leased lines or frame relay trunks. This paper presents a software-based VPN solution that interconnects offices over the public Internet with a secured tunnel. We build an emulated VPN environment on the Linux platform, and employ a high performance traffic generator and analyzer to measure its performance. We conducted many experiments to emulate different traffic and presented the results of ICMP traffic over Gigabit Ethernet. Our study demonstrated the feasibility of the proposed solution and the performance results show significant overhead of software-based VPN solution. Directions of further research to expand the work scope and to improve the performance are also presented in the paper.

Keywords: VPN, Linux, OpenSwan, FreeSwan, intranet, one-way delay, throughput

1 Introduction

An enterprise networking environment requires both the Internet and intranet connections. The need for intranet is to interconnect the headquarters with multiple branch offices. Traditionally, the intranet connection is provisioned via expensive leased lines or frame relay trunks. In some cases, companies may rely on a dial-up connection where the data speed is limited to only 53/48 kbps. The growing popularity of high speed internet access drives the need to interconnect multiple sites over the public Internet via a secured tunnel. The technology of IP-based Virtual Private Network (VPN) provides a solution to this need and offers great benefits to an organization. Secure remote access over public high-speed communication lines can help to reduce the need for costly private point-to-point telecommunication services. This secure “tunnel” communication between different sites of an organization is essential to cost effective operations of any enterprise. Recently, the trade report and study show the growing demand for VPNs as Information Technology (IT) departments are continually charged with the task of creating a secure and robust network infrastructure. A Gartner report predicts [1] that VPNs will be implemented by more than 80 percent of enterprises by the year 2007, as a form of network access control.
This paper explores the feasibility and performance of IP-based VPN configured on the Linux platform. Previous studies [2] have used packet loss, latency, and jitter as metrics for VPN performance. We were interested in total throughput and delay under different traffic loads of Gigabit Ethernet.

2 VPN Overview

There is a wide variety of choices for VPN implementations, from expensive hardware devices to inexpensive open source software that can be implemented with a minimal hardware cost. Implementing the wrong type of VPN has the potential to render the inherent cost savings useless [3]. Careful analysis must be performed to determine the best type of VPN that is appropriate for enterprise implementation. This is a difficult task, since there are few independent, concrete statistics on VPN efficiency. Many vendors are reluctant to publish exact efficiency statistics for their products. This is due in part to the wide variability of VPN encryption methods, hardware implementations, and supporting infrastructures. There are two primary methods of VPN tunneling:

- **Point-to-Point Tunneling Protocol (PPTP, RFC2637)** – this protocol is supported on the Windows Server environment, and it is widely available. However, PPTP is not an IETF standard (with an information only category) and its encryption mechanism is considered weak.

- **Layer-2 Tunnel Protocol (L2TP, RFC2661)**: L2TP is an IETF standard and it provides a layer-2 tunnel over the IP network. L2TP does not have an encryption protocol and it relies on IPSec for an encryption scheme to protect data integrity sent through the tunnel. IPSec/L2TP is considered more secure than PPTP, although its implementation is also more challenging. We chose to implement an IPSec-based VPN for maximum security protection, even though some members of the IT community believe it to be a difficult implementation choice on the Linux platform [4].

IPSec provides two encryption schemes: Authentication Header (AH, RFC2402) and Encapsulating Security Payload (ESP, RFC2406). Most IPSec implementations are based on ESP as it is considered more secure than AH. The experimental data presented in this paper is based on ESP. There are two VPN configurations for the enterprise environment: client-to-gateway VPN and gateway-to-gateway VPN. In a client-to-gateway VPN, mobile workers and telecommuters can access secure information on the company’s intranet over the public Internet as illustrated in Figure 1. The RADIUS server provides the AAA (authentication, authorization, and accounting) function to support remote clients.
Figure 1. Client-to-Gateway VPN Configuration

The gateway-to-gateway VPN configuration is used between headquarters and remote branch offices or between branch offices. This connection is traditionally provisioned via leased lines, frame relay, ATM, ISDN, or dial-up where they are either too expensive or too slow. If the connection is across multiple countries, it will not only be very expensive but also requires a long waiting time. For example, an office in Chicago needing to communicate proprietary information with a branch office in Brazil would not be able to easily communicate via a completely independent and private network. Establishing a tunnel VPN is an ideal solution to this problem as illustrated in Figure 2.

Figure 2. Gateway-to-Gateway VPN Configuration

The VPN tunnel is transparent to clients on private IP subnets, and there is no need for user authentication by the VPN gateways. The VPN gateways share a secret among
themselves as the mechanism for authentication and data encryption. As previously mentioned, this type of VPN is commonly used to send private, encrypted information over a public internet telecommunications line.

3 Previous Work

A 2001 study from the University of Texas measured CPU utilization across a VPN tunnel when running IPSec and PPTP implementations [5]. The study concluded that IPSec should be used when processing power is available, since it has a higher data transfer rate than PPTP. Our objective included determining the efficiency of an IPSec VPN implementation on a Linux platform. In contrast to the University of Texas study, we were interested in total network throughput, instead of focusing on CPU utilization.

A recent study [1] has shown that VPN efficiency degrades substantially when multiple protocols are in effect over the same network. (ICMP, HTTP, and FTP are running on a WLAN.) This study used Cisco Routers instead of Linux routers, and was focused on measuring delay in a videoconference with sound. Our objective also included picking a single type of traffic (ICMP) to accurately measure total network throughput.

A study of the efficiency of VPN traffic over a Wireless LAN [6] concluded that VPN traffic does not significantly impact the performance of a wireless network. This same study provided baseline data for a wired LAN, and our observations were comparable with that study, as well as a study from an annual IEEE conference on VPN efficiency [7].

4 Experimental Design

Several open source VPN solutions exist, but the most popular two are FreeSwan and OpenSwan. We attempted this experiment using the FreeSwan software, but found the implementation to be difficult with our equipment setup. Instead, we decided to implement an open source VPN using the shareware product, OpenSwan. Initial setup can be seen in Figure 3.
In Figure 3, each Linux server is configured as a router with two Gigabit Ethernet (IEEE 802.3z) interfaces – one for the private subnet, and the other for the VPN tunnel. The VPN tunnel passes through a Cisco 2950 switch, which also supports Gigabit Ethernet. Version 2.6 and above of the Linux kernel includes native IPSec support, though recompiling the Linux kernel is often necessary to enable it. In addition to recompiling the kernel, we installed version 0.6 of ipsec-tools, a Linux module that supports IPSec. Each Linux server was configured for IPSec and OpenSwan 2.3.1. Before establishing a VPN tunnel between the two servers, ICMP traffic was sent from aptop#1 to laptop#2. We confirmed that communication existed between laptops on different subnets, as shown in Figure 3.

We also confirmed the VPN configuration through a `tcpdump` command that ICMP traffic was being sent over the link in an unencrypted manner. After the tunnel was enabled, we confirmed that ICMP traffic was encrypted before being sent over the link. A `tcpdump` session showed that ESP (Encapsulating Security Payload) was present as packets were transferred across the VPN interface.

![tcpdump session](image)

**Figure 4.** `tcpdump` of traffic showing ECP packets

The initial setup in Figure 3 was modified to accommodate the IXIA traffic generator, so that we could measure network throughput and one-way delay before and after the VPN was enabled. Our modified setup can be seen in Figure 5.
5 Experimental Results

We follow the IETF standard of Benchmarking Methodology for Interconnect Network Devices (RFC2544) and collect the measurement data for various frame sizes. We first measure the data throughput before the VPN tunnel configuration. With 1Gbps links throughout the experiment, the IXIA generated ICMP traffic at 100% of the line rate (1000 Mbps). We measured an actual throughput rate of 210.4 Mbps from the left subnet to the right subnet at a frame size of 512 bytes. After applying the VPN tunnel under the same conditions, traffic throughput dropped by more than 785% to 26.8 Mbps.

<table>
<thead>
<tr>
<th>Frame Size</th>
<th>Throughput BEFORE VPN tunnel</th>
<th>Throughput AFTER VPN tunnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 bytes</td>
<td>34.4 Mbps</td>
<td>7.1 Mbps</td>
</tr>
<tr>
<td>128 bytes</td>
<td>68.7 Mbps</td>
<td>12.3 Mbps</td>
</tr>
<tr>
<td>256 bytes</td>
<td>137.3 Mbps</td>
<td>19.2 Mbps</td>
</tr>
<tr>
<td>512 bytes</td>
<td>210.4 Mbps</td>
<td>26.8 Mbps</td>
</tr>
<tr>
<td>1024 bytes</td>
<td>190.5 Mbps</td>
<td>33.5 Mbps</td>
</tr>
<tr>
<td>1450 bytes</td>
<td>180.3 Mbps</td>
<td>36.0 Mbps</td>
</tr>
</tbody>
</table>

Figure 6. Data Throughput before and after VPN Configuration
There is a corresponding increase in delay as the VPN is applied. Delay was measured in microseconds as the IXIA generated traffic at 1 packet per second.

<table>
<thead>
<tr>
<th>Frame Size</th>
<th>Delay BEFORE VPN tunnel (μs)</th>
<th>Delay AFTER VPN tunnel (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 bytes</td>
<td>165 μs</td>
<td>315 μs</td>
</tr>
<tr>
<td>128 bytes</td>
<td>169 μs</td>
<td>334 μs</td>
</tr>
<tr>
<td>256 bytes</td>
<td>171 μs</td>
<td>374 μs</td>
</tr>
<tr>
<td>512 bytes</td>
<td>196 μs</td>
<td>445 μs</td>
</tr>
<tr>
<td>1024 bytes</td>
<td>228 μs</td>
<td>612 μs</td>
</tr>
<tr>
<td>1450 bytes</td>
<td>260 μs</td>
<td>736 μs</td>
</tr>
</tbody>
</table>

The delay is most prominent at the 1450 byte frame size; delay is increased by more than 283% when the VPN tunnel is applied.

6 Conclusion and Future Work

Our experiment shows that VPN traffic requires a significant amount of processing overhead. There is a need to study methods of reducing the overhead of an encrypted VPN tunnel. Future work could include comparing these results to a Cisco-based VPN solution (with a VPN accelerated card), to see if performance is improved in a proprietary, hardware-based implementation. A recent study [2] measured packet loss, jitter, and latency of the VPN but did not measure total throughput. In addition, that study utilized several different types of data (ICMP, FTP, HTTP, voice and video traffic) while ours only used one – ICMP traffic.
Further, our experiment did not utilize a VPN feature known as Opportunistic Encryption. We manually configured each side of the VPN tunnel with an encryption key. Ideally, the encryption key needs to be changed often, so that would-be hackers cannot crack the key and eavesdrop the data being sent over the tunnel. IKE (Internet Key Exchange) daemons aid the secure automatic exchange process of new encryption keys. Several daemons exist on the Linux platform to implement IKE – Racoon, Pluto, and Isakmpd, to name a few. Our experiment did not take this additional overhead into account, and future work should include this in an efficiency study.

REFERENCES


