Virtual Private Networks and tinc

by Guus Sliepen

This presentation will show you what Virtual Private Networks (VPNs) are and what is so good about them. Real life applications with VPNs will be described and explained. Finally, we will show you how to build a VPN with the software we wrote ourselves: TINC.

What is a Virtual Private Network?

This needs an introduction first. The term VPN consists of three words. Only if we understand the words correctly will the full meaning of the term VPN become clear.

Virtual

Virtual means that something does not really exist, only in one's "imagination". For computers, this almost always means that it is faking or simulating something that is normally done in hardware.

From the perspective of the user or an application, everything works exactly the same, you just don't need the hardware for it you would normally need.

Private

We all know what private is: behind closed doors, without other people watching or listening, none of somebody else's business. For computers, private almost always means encryption. Since eavesdropping and copying information is so easy for computers (actually this is why they are so useful), only encryption can guarantee that even when data falls into the wrong hands, nobody else can read it.

Network

With the word "network" we can mean many different things, but it all comes down to a group of computers which are connected together somehow. Some may see a network as two or three computers in one room that are interconnected, others think a network is the thing they have at their company, while other still explain the word network as the biggest one we know: the Internet.

So what is a VPN then?

When we combine the three words, we get our own network which is made up of virtual network cards and cables and which nobody can listen or tap in to except for those who are allowed.

A VPN will look like a seperate local network to you, but in fact it can be composed of computers from all around the world, because the computer simulates the virtual network cables by making connections via the real Internet.

When is something *not* a VPN?

Something which claims to be a VPN is not a VPN when:

- It is not virtual: it needs hardware or it requires you to use real Internet addresses
- It is not private: there is no encryption or not everything is encrypted
- It is not a network: it does not allow you to use any networked application, only a few or actually none at all

Applications of VPNs

- For companies: let users connect from home to the company's internal network
- On a campus network: interconnect the networks you and others have behind (masquerading) firewalls
- On a severely restricted network: communicate on all ports with a remote network through firewalls with only a few ports open

How does a VPN work?

Most VPNs work like this:

- Set up a virtual network interface
- Set up a tunnel to another computer
- Intercept all the traffic coming from the virtual network interface
- Encrypt the traffic and send it over the tunnel
- And vice versa of course…

Well-known VPN solutions

- PPtP, L2TP or other PPP tunnels
- IPsec
- vpnd, VTun and TINC

PPtP, L2TP or other PPP tunnels

By using the well-known PPP protocol, encapsulated into normal Internet packets, tunnels can be set up to other hosts. PPtP and L2TP are both standardized ways of doing this (see RFC 2637 and 2661).

PPtP and L2TP, in most implementations, do not encrypt the actual network traffic. Therefore, they are not private. PPtP is well supported on most platforms though. L2TP is not so widely accepted.

IPsec

IPsec is an encryption standard for network packets. Instead of encapsulating packets and sending them over a tunnel, which in turn is encrypted, IPsec directly encrypts the packets. It actually is part of the IPv6 specification, and is not meant for making VPNs, but for making secure networks. But when IPsec is applied on tunneled packets, it does allow for the creation of a VPN.

IPsec is the way to go for secure networks, and it is being adopted by a lot of organizations for use in their operating systems or other software. IPsec based tunneling is also available on most platforms. FreeS/WAN is the best known solution for UNIX.

vpnd, VTun and tinc

Although the formerly mentioned standards are widely accepted, they all suffer from some problems. Some are really not so secure at all, therefore not being private, other standards are especially made for security but not for virtual networks. On top of that, almost all implementations of these standards require some modifications to the operating system itself, a huge barrier for most people.

Since the creation of the ethertap device in Linux, a lot of userspace VPN solutions have been created. Best known are vpnd, VTun and TINC. They all are specifically written to meet all needs of VPNs, and care more about perfectioning than standardizing.

tinc

there is no cabal

What is tinc?

TINC is a Virtual Private Network daemon, a piece of software that will simulate a local network and communicate with other TINC daemons on other computers using encryption.

TINC currently only works on Linux, but it is already being ported to FreeBSD and Solaris.

Who made tinc?

Since TINC is Free (as in Speech) Software, a lot of people have already contributed in some way to it. The main authors however are:

- Ivo Timmermans `<itimmermans@bigfoot.com>`
  Mathematics student in Delft, team leader of the Dutch section of the Translation Project, author of gldap.

- Guus Sliepen `<guus@sliepen.warande.net>`
  Physics student in Utrecht, author of the route-based load balancing patch for the Linux kernel.

Goals of tinc

- Security: it should be as private as possible, and that is not an easy thing to do...
- Reliability: the TINC daemon should run as long as your computer is running without having to restart it
- Efficiency: avoid any unnecessary overhead
- Scalability: hundreds of TINC daemons should be able to connect to eachother without having configuration files hundreds of times as large
- Easy to use: it should be relatively easy to set up and maintain

Security

Encrypting packets is a nice thing, but encryption by iteself does not make a program secure. Buffer overflows, memory leaks and human errors lurk around the corner everywhere. Although we put a lot of effort in avoiding and fixing security holes, we made a *cryptoblunder* anyway.

We are trying to use and follow widely accepted standards now, like OpenSSL and SSH, and we have read up on cryptography. Auditing by third parties will also help making a secure product.

Scalability

Most VPN solutions provide connections between only two sides of a virtual network. This is mainly because of the fact that most of them use PPP for encapsulating packets or two-sided tunnels. By adding more tunnels manually, larger VPNs can be created, but this does not scale very well:

- Large configuration files
- Hard to maintain all this
- Routing is inefficient, or
- Efficient routing makes even larger configuration files

# hosts	# tunnels (minimum)	# tunnels (efficient network)
2	1	1
3	2	3
4	3	6
5	4	10
10	9	45
100	99	4950
1000	999	499500
n	$n-1$	$\frac{1}{2}n(n-1)$
	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$

Table 1: Scalability of Virtual Private Networks

Scalability in tinc

As can be seen in table 1, the number of tunnels needed for an efficient network is quadratic in the number of hosts. There is no way this number can be reduced, it merely reflects the number of connections there possibly are between the hosts.

Although a lot of VPN solutions require you to put $\mathcal{O}(n^2)$ effort in setting up the configuration files, we tried to make TINC need only $\mathcal{O}(n)$ effort while still making up an efficient network.

TINC achieves this by communicating with all the hosts in a smart way and extrapolating the rest of the information needed to build an efficient network.

Efficiency

Efficiency in TINC is done by:

- Encapsulating packets directly into UDP packets instead of PPP or PPP over TCP streams
- Letting packets take the most direct route to their destination (see scalability)

Where to find tinc?

`http://tinc.nl.linux.org/`