This column focuses on secure computing, providing tools and tips for those in the information security trenches. Each issue, we'll evaluate new technologies (primarily in the open source space) and discuss ways to integrate them into your organization.

We want to hear from you. Got a great utility or "magic" script that's saved you hours of tedious keyboard pounding? Something new we haven't heard about? Let us know at ciocorner@sandstorm.net.

Virtual Private Networks

Plan before you open the tunnel

Virtual Private Networks (VPNs) have been painted as the ideal solution for remote connectivity. The technology behind implementation has matured, and deployments abound. If you are new to VPNs, however, you will find unexpected twists at every turn.

A VPN provides a secure connection over unsecured channels (such as the Internet). This definition is rather broad, so a lot of technology may fit. We'll try to break down some of the major strategies to creating a VPN, and describe a quick way to get a VPN running on a test network.

A VPN allows remote users to become part of your corporate network with the same addressing scheme as local network users. This allows users to access resources utilizing the authentication mechanisms that are already in place. Additionally, the VPN provides authentication, access control, confidentially (through the use of encryption over a non-trusted connection), and data integrity. Sounds good so far.

Unfortunately, implementing a VPN can be tricky. By encrypting connections, you incur a high performance hit, which may be too high over low-bandwidth connections. Also, inter-vendor compatibility can be an issue with VPN devices. It can be very difficult to connect some devices, even if the device claims to be "compatible" with others.

There is also a philosophical hurdle to overcome when implementing a VPN. By adding this technology to your infrastructure, you are extending the LAN beyond physical boundaries. You may no longer have direct control over the computing resources that are used to connect to your network.

You might unknowingly be giving LAN access to clients at home who don't run firewalls, don't patch their systems, and don't update their anti-virus software. It is critical to insure that remote VPN users do not compromise all the hard work that's been put into securing the network.

Well-defined policies will become extremely important when dealing with these problems. Remote VPN access might need to be limited to a pool of corporate laptops that have gone through rigorous security audits and get regular updates.

TUNNELING

The encapsulation and transmission of packets over a connection is called "tunneling." There are many protocols that may be used over this tunnel. Here are a few of the major ones:

Point-to-Point Tunneling Protocol (PPTP) is used to create on-demand VPNs. PPTP servers handle encryption and decryption of IP datagrams, and routing of the decrypted traffic to the private network.

Layer 2 Tunneling Protocol is a combination of PPTP technology with Layer 2 Forwarding (L2F). This protocol can be used for both site-to-site as well as remote access VPNs.

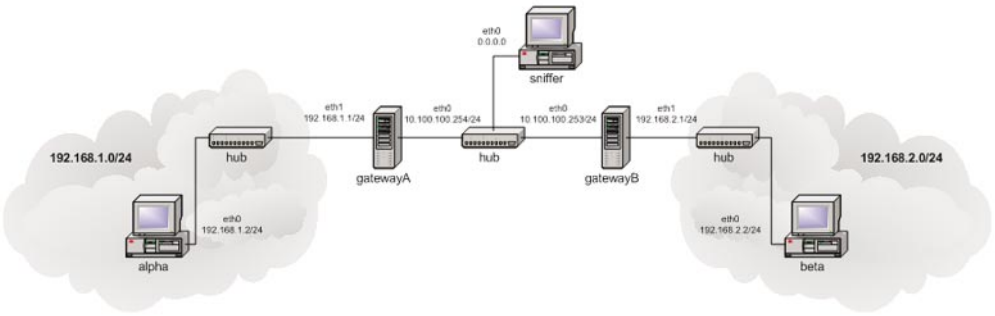
IPSec (IP Security) is a developing standard for encryption at Layer 3 (the packet processing layer of network communications). It provides two choices of security service: Authentication Header (AH) and Encapsulating Security Payload (ESP).

The AH essentially allows authentication of the sender of data, and provides protection against replay

attacks (an attack in which valid data is replayed at a later time). ESP supports both authentication of the sender and encryption of data as well.

There are several free implementations of IPSec, such as FreeS/WAN (www.freeswan.org) and Openswan (www.openswan.org).

Sometimes the best way to get a handle on technology is to try it out. Since Knoppix-std (www.knoppix-std.org) comes preloaded with the FreeS/WAN, we decided to set up a test network. We configured our network topology like so:



We installed Knoppix-std to the hard drives of each machine. The configuration of FreeS/WAN is a little tricky – the major stumbling block was determining the meaning of "left" and "right" in the IPSec config file.

Essentially, you create a public/private key-pair on both gateways, and then set up the config file so that each gateway knows about the other (for our config files, see www.sandstorm.net/ciocorner). Then, start up the IPSec daemon on both machines, and initiate the tunnel.

Once the tunnel is running, both hosts (alpha and beta) will be able to communicate, just as if they were on the same LAN. The "sniffer" machine will only be able to see the encrypted traffic.

Putting in the upfront work to determine which VPN protocols are right for your environment, and setting up a small test network, will allow you to have a much smoother deployment. Working through in advance the new security issues that extended network perimeters create can save several hours of battling unexpected attacks.

Have some VPN war stories to share? Let us know, and good luck engineering a safe tunnel for your remote users.

READER RESPONSE

Robert Z wrote in regard to our column last month ("Incident Response Toolkits," July 2004). He said "I've just read your column and appreciate the incident response steps, but what would you recommend as a good toolkit to have on hand to accomplish the investigation?"

Great question Robert, and we're glad that you enjoyed the article! We first started making customized Knoppix CDs, as kind of a "home grown" toolkit to keep with us wherever we went. However, after discovering how time consuming this was, we looked for other alternatives and came across a project called FIRE (http://biatchux.dmzs.com). We've been using it ever since, and it's currently our toolkit of choice.

Most of our work in this area involves Unix-like systems, so we don't have any good suggestions if you're looking for a Windows based toolkit (apart from the big guns: Encase and the like). But, FIRE does contain many useful programs for scanning Windows machines. If anyone has a good open source Windows toolkit, please let us know so at ciocorner@sandstorm.net. CDM

Walker Whitehouse *is CIO of Sandstorm Enterprises and* **Mike Yamamoto** *is a Network Systems Engineer at Sandstorm. Sandstorm Enterprises develops aggressive software products for network monitoring, network forensics analysis, and security auditing including telephone scanning, penetration testing, and vulnerability assessment.*