The journal of Apple technology.

Volume Number: 18 (2002)
Issue Number: 07
Column Tag: Viewpoint

by Rich Morin

Privacy by Default

Making Rendezvous safe "for the rest of us"

Although Apple does not use these words to describe it, their new Rendezvous system is designed to enable "opportunistic, promiscuous" IP discovery. That is, it will take advantage of any opportunity to collect IP information (hence, opportunistic) and it will happily interact with any cooperative system (hence, promiscuous). These characteristics make Rendezvous extremely convenient for the user.

Unfortunately, the same characteristics could also make Rendezvous extremely convenient for anyone who wants to "listen in" on network traffic. If a user walks into a conference facility or coffee bar with a PowerBook, how many uninvited recipients will see his network traffic?

I'd like to see Apple provide "convenient privacy", as part of its convenient networking. To be specific, I'd like them to implement "opportunistic, promiscuous" packet-level encryption, basedon IPSec and related standards. This isn't a new idea, by any means. The "Linux FreeS/WAN" project (http://www.freeswan.org) has been working on it for several years now; their early releases are currently being tried out in the field.

The high-level view of FreeS/WAN is quite simple. If my system has a packet to send to your system, it will first attempt to set up a VPN (Virtual Private Network), using IPSec, etc. If your system doesn't honor the request, my system will simply send the packet "in the clear".

If your system does understand the request, however, both systems will send off for each others' public keys (e.g., from each others' DNS servers). The two systems will then perform a key exchange, with the result that they both end up with the needed "session keys". Et voila, we have a VPN!

The low-level description is a bit more complicated, but some folks may find it interesting. See http://www.freeswan.org/freeswan_trees/freeswan-1.95/doc/intro.html if you're into that sort of thing...

Meanwhile, let's look at some of the implications of the technology. In general, privacy and security tend to be at odds with convenience. And, as we all know, when "I really should" gets in a fight with "I don't want to", it usually loses. Consequently, although PGP and other strong privacy tools have been around for several years, they aren't actually used very much. Even SSH has had an uphill fight against its (demonstrably insecure) predecessors.

By making network privacy the default, however, Apple could remove the "convenience factor" from the equation. Joe and Sally Sikspak don't have to install privacy software, set up keywords, or any of that hassle. Better yet, they don't have to decide which programs (or files, or ...) deserve encryption. No decisions, so no mistakes! In fact, they don't even have to know that encryption is going on; their packets are simply a bit safer from snooping.

Apple likes to be seen as a standard-setter. By allying themselves with the Linux FreeS/WAN project (and, ideally,

providing an Open Source BSD implementation), Apple could help to make opportunistic privacy an established standard. There are no guarantees, of course, but privacy and authentication are very salable attributes these days...

Even without total buy-in by the computer industry, the effects of opportunistic encryption could be quite dramatic. For instance, it would be quite possible to set up a FreeS/WAN “gateway server” at an ISP or on the border of a LAN, providing encryption capability for any external traffic. Like the Sikspaks, the machines being protected would never need to know that their packets were being encrypted.

In addition, large-scale use of packet-level encryption would make it much harder to single out encrypted data streams for attack. If even 5% of the Internet’s traffic is encrypted, the mere fact of encryption is no longer an “interesting” characteristic for snoopers.

Rich Morin has been using computers since 1970, Unix since 1983, and Mac-based Unix since 1986 (when he helped Apple create A/UX 1.0). When he isn’t writing this column, Rich runs Prime Time Freeware (www.ptf.com), a publisher of books and CD-ROMs for the Free and Open Source software community. Feel free to write to Rich at rdm@ptf.com.